การศึกษาวิธีการแบ่งกลุ่มลูกค้าด้วยเทคนิคอาร์เอฟเอ็มร่วมกับการวิเคราะห์ข้อความ

THE STUDY OF CUSTOMER SEGMENTATION BY USING RFM MODEL AND TEXT ANALYTICS

เอกปรียา ใบสนิ

บัณฑิตวิทยาลัย มหาวิทยาลัยศรีนครินทรวิโรฒ

2563

การศึกษาวิธีการแบ่งกลุ่มลูกค้าด้วยเทคนิคอาร์เอฟเอ็มร่วมกับการวิเคราะห์ข้อความ

เอกปรียา ใบสนิ

สารนิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
วิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการข้อมูล
คณะวิทยาศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ
ปีการศึกษา 2563

THE STUDY OF CUSTOMER SEGMENTATION BY USING RFM MODEL AND TEXT ANALYTICS

AEKPREYA BAISANI

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

for the Degree of MASTER OF SCIENCE

(Data Science)

Faculty of Science, Srinakharinwirot University

2020

สารนิพนธ์

เรื่อง

การศึกษาวิธีการแบ่งกลุ่มลูกค้าด้วยเทคนิคอาร์เอฟเอ็มร่วมกับการวิเคราะห์ข้อความ

ของ

เอกปรียา ใบสนิ

ได้รับอนุมัติจากบัณฑิตวิทยาลัยให้นับเป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

ปริญญาวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการข้อมูล

ของมหาวิทยาลัยศรีนครินทรวิโรฒ

(รองศาสตราจารย์ นายแพทย์ฉัตรชัย เอกปัญญาสกุล)

คณบดีบัณฑิตวิทยาลัย

คณะกรรมการสอบปากเปล่าสารนิพนธ์

..... ที่ปรึกษาหลัก

(อาจารย์ ดร.ศิริสรรพ เหล่าหะเกียรติ)

..... ประธาน

(อาจารย์ ดร.สุทธิพงศ์ ธัชยพงษ์)

..... กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.ศุภชัย ไทยเจริญ)

ชื่อเรื่อง	การศึกษาวิธีการแบ่งกลุ่มลูกค้าด้วยเทคนิคอาร์เอฟเอ็มร่วมกับการวิเคราะห์ข้อความ
ผู้วิจัย	เอกปรียา ใบสนิ
ปริญญา	วิทยาศาสตรมหาบัณฑิต
ปีการศึกษา	2563
อาจารย์ที่ปรึกษา	อาจารย์ ดร. ศิริสรรพ เหล่าหะเกียรติ

งานวิจัยนี้มีวัตถุประสงค์เพื่อศึกษาวิธีการแบ่งกลุ่มลูกค้า โดยใช้เทคนิคการเรียนรู้ของเครื่อง เพื่อจัดกลุ่มของข้อมูลร่วมกับการวิเคราะห์ข้อความ โดยประกอบไปด้วย 2 วิธีหลักดังนี้ 1. การแบ่งกลุ่มลูกค้าด้วยเทคนิคอาร์เอฟเอ็ม 2. การแบ่งกลุ่มด้วยเทคนิคอาร์เอฟเอ็มและการวิเคราะห์ข้อความด้วยเครื่องมือประมวลภาษาธรรมชาติ โมดูลเอ็นแอลทีเค (Natural Language Toolkit) โดยใช้ชุดข้อมูลการซื้อขายสินค้าของร้านค้าปลีกในประเทศอังกฤษที่เกิดขึ้นในช่วงวันที่ 1 ธ.ค. 2553 ถึง 9 ธ.ค. 2554 จำนวนข้อมูลทั้งหมด 541,909 แถว จากฐานข้อมูล UCI Machine Learning Repository ผู้วิจัยสนใจที่จะเปรียบเทียบผลของการจัดกลุ่มโดยใช้ฟีเจอร์ที่ได้จากเทคนิคอาร์เอฟเอ็มและเทคนิควิเคราะห์ข้อความมารวมกัน เปรียบเทียบกับการจัดกลุ่มโดยใช้ฟีเจอร์ที่ได้จากเทคนิคอาร์เอฟเอ็ม ในการเปรียบเทียบค่า Adjusted Rand Index และค่า Normalized Mutual Information ระหว่างวิธีที่ 1 และวิธีที่ 2 ซึ่งให้ค่า ARI เท่ากับ 0.5116 และ NMI เท่ากับ 0.3646 ค่า Confusion Matrix ที่ได้จากการเปรียบเทียบทั้ง 2 วิธี ผลที่ได้จากการแบ่งกลุ่มไม่สอดคล้องกันมากนัก จากการเปรียบเทียบการแบ่งกลุ่มด้วยเทคนิคอาร์เอฟเอ็มร่วมกับการวิเคราะห์ข้อความ มีประสิทธิภาพดีกว่าวิธีที่ใช้ฟีเจอร์ที่ได้จากเทคนิคอาร์เอฟเอ็มเพียงเทคนิคเดียว ในแง่ที่ผลการจัดกลุ่มที่ได้ จะให้ข้อมูลเชิงลึกของการสั่งซื้อร่วมกับข้อมูลในส่วนของศักยภาพของลูกค้าจากข้อมูลอาร์เอฟเอ็ม

คำสำคัญ : การแบ่งกลุ่มลูกค้า, การวิเคราะห์ข้อความ, การเรียนรู้ของเครื่อง, การจัดกลุ่มของข้อมูล, เทคนิคอาร์เอฟเอ็ม

Title	THE STUDY OF CUSTOMER SEGMENTATION BY USING RFM MODEL AND TEXT ANALYTICS
Author	AEKPREYA BAISANI
Degree	MASTER OF SCIENCE
Academic Year	2020
Thesis Advisor	Sirisup Laohakiat , PhD

This study aims to perform customer segmentation by utilizing clustering in machine learning based on RFM model and by using Text Analytics that consists of two main methods, as follows: (1) customer segmentation based on the RFM analysis Model Two. The process of customer segmentation was based on the RFM analysis model and Text Analytics operations with the NLTK (Natural Language Toolkit). This research used transactional dataset which contained all of the transactions occurring between 01/12/2010 and 09/12/2011, for a UK-based and registered non-store online retail with sample of 541,909 rows from UCI Machine Learning Repository. The results of the two methods were compared using clustering performance indices (adjusted rand index = 0.5116 and normalized mutual information = 0.3646) and the confusion matrix showed that the two methods were inconsistent. The results showed that the clusters obtained from customer segmentation were based on the RFM analysis model and Text Analytics method, which revealed insight on both the potential of customers based on RFM features, as well as the characteristics of the order in the clusters which was not present when using only RFM features alone.

Keyword : Customer Segmentation Text Analytics Machine Learning Clustering RFM Technique

กิตติกรรมประกาศ

สารนิพนธ์นี้สำเร็จลุล่วงได้ด้วยความอนุเคราะห์จาก ดร.ศิริสรรพ เหล่าหะเกียรติ อาจารย์ที่ปรึกษา ที่ให้คำปรึกษา คำแนะนำในการทำสารนิพนธ์ ตลอดจนสนับสนุนข้อมูลทางวิชาการ

ขอกราบขอบพระคุณคณะกรรมการสอบสารนิพนธ์ ที่ได้ให้คำแนะนำและข้อเสนอแนะสำหรับการปรับปรุงสารนิพนธ์ และการใช้เทคนิคการเรียนรู้ของเครื่องเป็นเครื่องมือในการวิเคราะห์และแก้ไขปัญหาทางข้อมูลต่อไป

ขอกราบขอบพระคุณ บัณฑิตวิทยาลัย มหาวิทยาลัยศรีนครินทรวิโรฒ สำหรับทุนสนับสนุนการนำเสนอผลงานวิจัยของบัณฑิตศึกษา ทำให้ได้ประสบการณ์นำเสนอสารนิพนธ์ และได้แลกเปลี่ยนความรู้ทางด้านเทคโนโลยี คณิตศาสตร์ คอมพิวเตอร์ และการใช้เทคนิคการเรียนรู้ของเครื่อง

เอกปรียา ใบสนิ

สารบัญ

หน้า

สารบัญตาราง

หน้า

สารบัญรูปภาพ

หน้า

บทที่ 1
บทนำ

1.1 ภูมิหลัง

ปัจจุบันการตลาดสมัยใหม่หมุนเวียนเปลี่ยนไปอย่างรวดเร็วและแตกต่างไปจากการทำตลาดในอดีตอย่างสิ้นเชิง การแบ่งกลุ่มลูกค้า เพื่อให้สามารถสื่อสารกับลูกค้าแต่ละกลุ่มจึงมีรูปแบบการสื่อสารที่มีเป้าหมายแตกต่างกันออกไป และมีความสำคัญมากในการพัฒนาโมเดลธุรกิจที่ต้องทำให้ชัดเจน, จำเพาะเจาะจง และเป็นเหตุเป็นผลกัน เพื่อเพิ่มประสิทธิภาพในการดำเนินกิจกรรมทางการตลาด

การจัดกลุ่มลูกค้า (Customer Segmentation) เป็นวิธีการจัดกลุ่มโดยที่ลูกค้าที่มีพฤติกรรมใกล้เคียงกันหรือเหมือนกันจะถูกจัดไว้ในกลุ่มเดียวกัน ส่วนลูกค้าที่มีพฤติกรรมแตกต่างหรือไม่เหมือนกันก็จะถูกจัดไว้คนละกลุ่ม การจัดกลุ่มลูกค้าทำให้ธุรกิจสามารถทราบถึงลักษณะหรือพฤติกรรมของลูกค้าของตนเองในแต่ละกลุ่มได้ เมื่อธุรกิจเข้าใจพฤติกรรมหรือลักษณะของลูกค้าในแต่ละกลุ่มก็จะสามารถเสนอสินค้าหรือบริการที่ตรงต่อพฤติกรรมของลูกค้าได้ หากนำเทคนิคคลัสเตอริงหรือการเรียนรู้ของเครื่องประเภทการจัดกลุ่มของข้อมูล ที่ไม่มี Target หรือไม่มีต้นแบบของผลลัพธ์ ซึ่งเป็นโมเดลที่เอาไว้ใช้การจัดกลุ่มของข้อมูล จะทำให้สามารถแบ่งกลุ่มของผลิตภัณฑ์ตามความประสงค์ของลูกค้าได้ และยังสามารถนำมาประยุกต์ใช้กับการทำงานในหน่วยงานต่างๆได้ ก็จะยิ่งส่งเสริมให้งานนั้นๆเกิดขึ้นได้อย่างมีประสิทธิภาพ ดังนั้นงานวิจัยนี้จึงศึกษาวิธีการแบ่งกลุ่มลูกค้าจากข้อมูลการซื้อขายสินค้าโดยใช้วิธีการวิเคราะห์ข้อความและเทคนิคการเรียนรู้ของเครื่อง ประเภทการจัดกลุ่มของข้อมูล เพื่อนำผลลัพธ์ที่ได้มาวิเคราะห์พฤติกรรมลูกค้าและกำหนดเป็นกลยุทธ์ทางการตลาดต่อไป

แม้ว่าจะมีงานวิจัยจำนวนไม่น้อย ที่ศึกษาเกี่ยวกับการจัดกลุ่มลูกค้า การศึกษาเหล่านั้นอาจแบ่งเป็น 2 แนวทางได้แก่ แนวทางที่ 1 ผู้ศึกษาจะอาศัยข้อมูลจากใบสั่งซื้อ เพื่อจัดกลุ่มข้อมูลโดยอาศัยความคล้ายคลึงกันของสินค้าที่สั่งซื้อของลูกค้าแต่ละราย ส่วนในแนวทางที่ 2 ผู้ศึกษาจะจัดกลุ่มโดยอาศัยแบบจำลองอาร์เอฟเอ็ม ซึ่งเป็นการจัดกลุ่มลูกค้า โดยจำแนกตามศักยภาพของลูกค้าผ่านแบบจำลองอาร์เอฟเอ็ม โดยวิเคราะห์จากใบสั่งซื้อ การแบ่งกลุ่มเช่นนี้ อาจทำให้เราไม่สามารถมองเห็นภาพรวมของลูกค้าได้ชัดเจนในทุกๆแง่มุมได้ดีพอ ในการศึกษาที่ผ่านมา ผู้วิจัยจึงทำการจัดกลุ่มลูกค้า โดยอาศัยข้อมูลจากทั้งสองส่วน อันได้แก่ ข้อมูลการสั่งซื้อสินค้าในใบสั่งซื้อ ร่วมกับการใช้แบบจำลองอาร์เอฟเอ็ม เพื่อวัดศักยภาพของลูกค้า มีประกอบร่วมกัน เพื่อหาแนวทางในการสร้างแบบจำลองการจัดกลุ่มลูกค้า เพื่อให้มีประสิทธิภาพในการจัดกลุ่มสูงสุด

นอกจากนี้ ในการวิเคราะห์ข้อมูลการสั่งซื้อสินค้าในใบสั่งซื้อ ผู้วิจัยยังใช้เทคนิคการวิเคราะห์ข้อความร่วมด้วย เพื่อช่วยในการจัดกลุ่มสินค้า เพื่อให้แบบจำลองสามารถจัดกลุ่มได้อย่างมีความแม่นยำมากขึ้น

1.2 ความมุ่งหมายของงานวิจัย

ในการวิจัยครั้งนี้ผู้วิจัยได้ตั้งความมุ่งหมายไว้ดังนี้

1.2.1 เพื่อแบ่งกลุ่มลูกค้าที่มีพฤติกรรมใกล้เคียงกันหรือเหมือนกันออกเป็นกลุ่มย่อย โดยใช้เทคนิค Machine learning แบบ Unsupervised Learning ประเภท Clustering

1.2.2 เพื่อศึกษาว่าตัวแปรใดในข้อมูลการซื้อขายสินค้าที่มีความสำคัญในการแบ่งกลุ่มลูกค้า

1.2.3 เพื่อมีความรู้ความเข้าใจเกี่ยวกับการแบ่งกลุ่มลูกค้าเป็นกลุ่มย่อย (Customer Segmentation)

1.2.4 ทราบถึงหลักการของอัลกอริธึม K-means ที่สามารถใช้ทำ Clustering Model ได้และสามารถนำไปใช้ประโยชน์ในการเพิ่มประสิทธิภาพในการดำเนินกิจกรรมทางการตลาด

1.3 ความสำคัญของการวิจัย

งานวิจัยนี้ศึกษาการแบ่งกลุ่มลูกค้าที่มีพฤติกรรมใกล้เคียงกันหรือเหมือนกันออกเป็นกลุ่มย่อย โดยใช้ Machine learning เป็นเครื่องมือสำหรับสร้างแบบจำลองในการแบ่งกลุ่มลูกค้า แบบ Unsupervised Learning ประเภท Clustering ทดลองกับข้อมูลการซื้อขายสินค้าของร้านค้าปลีกในประเทศอังกฤษที่เกิดขึ้นในช่วงวันที่ 1 ธ.ค. 2553 ถึง 9 ธ.ค. 2554 ประกอบด้วย 8 แอททริบิวต์ และมีจำนวนข้อมูลทั้งหมด 541,909 แถว

1.4 ขอบเขตของการวิจัย

1.4.1 ประชากรที่ใช้ในการวิจัย

ข้อมูลการซื้อขายสินค้าของร้านค้าปลีกในประเทศอังกฤษที่เกิดขึ้นในช่วงวันที่ 1 ธ.ค. 2553 ถึง 9 ธ.ค. 2554 ประกอบด้วย 8 แอททริบิวต์ มีจำนวนข้อมูลทั้งหมด 541,909 แถว

1.4.2 กลุ่มตัวอย่างที่ใช้ในการวิจัย

ข้อมูลการซื้อขายสินค้าของร้านค้าปลีกในประเทศอังกฤษที่เกิดขึ้นในช่วงวันที่ 9 ธ.ค. 2553 ถึง 9 ธ.ค. 2554

1.4.3 ตัวแปรที่ศึกษา

1.4.3.1 ตัวแปรอิสระ มีรายละเอียดดังนี้

1.4.3.1.1 ข้อมูลลูกค้า

- รหัสลูกค้า
- ประเทศที่ลูกค้าอาศัย

1.4.3.1.2 ข้อมูลสินค้า

- รหัสสินค้า
- ชื่อสินค้า
- ราคาสินค้าต่อหน่วย

1.4.3.1.3 ข้อมูลใบสั่งซื้อ

- เลขที่ใบสั่งซื้อ
- วัน/เดือน/ปี และเวลาในใบสั่งซื้อ
- จำนวนสินค้าที่ซื้อต่อหนึ่งใบสั่งซื้อ

1.5 กรอบแนวคิดในงานวิจัย

งานวิจัยนี้ศึกษาการแบ่งกลุ่มลูกค้าที่มีพฤติกรรมใกล้เคียงกันหรือเหมือนกันออกเป็นกลุ่มย่อย โดยใช้ Machine learning เป็นเครื่องมือสำหรับสร้างแบบจำลองในการแบ่งกลุ่มลูกค้า โดยข้อมูลการซื้อขายสินค้าของร้านค้าปลีกในประเทศอังกฤษที่เกิดขึ้นในช่วงวันที่ 1 ธ.ค. 2553 ถึง 9 ธ.ค. 2554 ประกอบด้วย 8 แอททริบิวต์ มีจำนวนข้อมูลทั้งหมด 541,909 แถว โดยข้อมูลจะถูกเก็บในรูปแบบตาราง จากนั้นจึงใช้ Machine learning มาเป็นเครื่องมือช่วย สร้างแบบจำลองเพื่อแบ่งกลุ่มลูกค้า ซึ่งเขียนด้วยภาษาโปรแกรม Python โดยใช้ เทคนิคการจัดการฟีเจอร์ 2 แบบ คือ การสร้างฟีเจอร์ใหม่ด้วยเทคนิคอาร์เอฟเอ็มและตัดฟีเจอร์ที่ไม่เกี่ยวข้อง (Feature Engineering) การวิเคราะห์ข้อความ (Text Analytics) และใช้เทคนิค K-means ในการทำ Clustering เพื่อหาแบบจำลองที่มีประสิทธิภาพในการแบ่งกลุ่มดี โดยประกอบไปด้วย 2 วิธีหลักดังนี้ 1) การแบ่งกลุ่มลูกค้าด้วยเทคนิคอาร์เอฟเอ็ม 2) การแบ่งกลุ่มด้วยเทคนิคอาร์เอฟเอ็มและการวิเคราะห์ข้อความ ผู้วิจัยสนใจที่จะเปรียบเทียบผลของการจัดกลุ่มโดยใช้ฟีเจอร์ที่ได้จากเทคนิคอาร์เอฟเอ็มและเทคนิควิเคราะห์ข้อความมารวมกัน เปรียบเทียบกับการจัดกลุ่มโดยใช้ฟีเจอร์ที่ได้จากเทคนิคอาร์เอฟเอ็ม

1.6 สมมติฐานในการวิจัย

1.6.1 แบบจำลองที่สร้างด้วยอัลกอริธึม K-means มีประสิทธิภาพที่ดี เพราะในกรณีที่มีจำนวนข้อมูลมาก และมีจำนวนกลุ่มน้อย การจัดกลุ่มด้วย K-means อาจจะคำนวณได้เร็วกว่าแบบอื่นๆ

1.6.2 การใช้ Feature Engineering ก่อนสร้างแบบจำลองทำให้แบบจำลองมีประสิทธิภาพที่ดีกว่าการสร้างแบบจำลองที่ไม่มีขั้นตอน Feature Engineering

1.6.3 แอททริบิวต์ชื่อสินค้า เมื่อผ่านการวิเคราะห์ข้อความสามารถสร้างเป็นฟีเจอร์ใหม่ ที่ช่วยสะท้อนความคล้ายคลึงกันของสินค้าที่สั่งซื้อของลูกค้าแต่ละรายได้

1.6.4 วิธีการจัดกลุ่มด้วยเทคนิคอาร์เอฟเอ็มและการวิเคราะห์ข้อความมีประสิทธิภาพที่ดีกว่าวิธีการจัดกลุ่มด้วยเทคนิคอาร์เอฟเอ็มเพียงเทคนิคเดียว เพราะสามารถสะท้อนให้เห็นกลุ่มลูกค้าจากการซื้อขายสินค้า อีกทั้งยังสะท้อนให้เห็นกลุ่มสินค้าที่ใกล้เคียงกันอีกด้วย

บทที่ 2
เอกสารและงานวิจัยที่เกี่ยวข้อง

ในการวิจัยครั้งนี้ ผู้วิจัยได้ศึกษาเอกสารและงานวิจัยที่เกี่ยวข้อง และนำเสนอตามหัวข้อต่อไปนี้

1. การแบ่งกลุ่มลูกค้า (Customer Segmentation)
2. อัลกอริธึมที่ใช้ในการจัดกลุ่มของข้อมูล (Clustering Algorithms)
3. การสร้างฟีเจอร์ใหม่และตัดฟีเจอร์ที่ไม่เกี่ยวข้อง (Feature Engineering)
4. การวิเคราะห์ข้อความ (Text Analytics)
5. งานวิจัยที่เกี่ยวข้อง

2.1 การแบ่งกลุ่มลูกค้า (Customer Segmentation)

เมื่อบริษัทมีข้อมูลลูกค้าเป็นปริมาณมากขึ้นการจะพิจารณาข้อมูลลูกค้าทีละรายก็คงทำได้ไม่สะดวกนักและการมองว่าลูกค้าทุกรายมีลักษณะเหมือนกันหมดก็ไม่ถูกต้องเท่าไร ดังนั้นเพื่อให้บริษัทเข้าใจลูกค้าได้อย่างมากขึ้นจึงมีความจำเป็นต้องทำการแบ่งกลุ่มลูกค้า

การแบ่งกลุ่มลูกค้า (Customer Segmentation) คือ การแบ่งตลาดออกเป็นกลุ่มเล็ก ๆ โดยแต่ละกลุ่มจะมีความต้องการแตกต่างกัน มีลักษณะและพฤติกรรมเฉพาะตัว ซึ่งเราสามารถที่จะแบ่งกลุ่มออกไปได้ด้วยปัจจัยหลายแบบ ได้แก่

Geographic Segmentation การแบ่งกลุ่มโดยใช้หน่วยทางภูมิศาสตร์มาเป็นตัวแบ่ง เช่น สัญชาติ, เชื้อชาติ, รัฐ, จังหวัด, เขตพื้นที่ เป็นต้น

Demographic Segmentation การแบ่งกลุ่มโดยใช้ลักษณะเฉพาะ เช่น เพศ, อายุ, ลักษณะครอบครัว, รายได้, อาชีพ, ศาสนา เป็นต้น

Psychographic Segmentation การแบ่งกลุ่มโดยใช้ลักษณะเฉพาะทางสังคม เช่น ชนชั้นทางสังคม (Social Class), ลักษณะการใช้ชีวิตประจำวัน (Lifestyle) หรือลักษณะเฉพาะส่วนบุคคล เป็นต้น

Behavioral segmentation การแบ่งกลุ่มโดยใช้ลักษณะทางพฤติกรรม เช่น ความรู้ความเข้าใจในผลิตภัณฑ์, ทัศนคติ, การใช้งานและการตอบรับต่อผลิตภัณฑ์และบริการที่เสนอ

การเลือกเป้าหมาย (Targeting) คือ กระบวนการในการประเมินกลุ่มลูกค้าแต่ละกลุ่มที่เราแบ่งเอาไว้แล้วว่ากลุ่มไหนน่าจะตอบสนองต่อ Campaign ทางการตลาดได้ดีที่สุด การเลือกควรคำนึงถึง 3 ปัจจัย ได้แก่ ขนาดและการเติบโต (Size and Growth) ความน่าดึงดูด (Attractiveness) เป้าหมายทางธุรกิจ (Business Objectives)

2.2 อัลกอริธึมที่ใช้ในการจัดกลุ่มของข้อมูล (Clustering Algorithms)

Clustering คือ Machine Learning Model ประเภท Unsupervised ที่ไม่มี Target หรือไม่มีต้นแบบของผลลัพธ์ ซึ่งเป็น Model ที่นำไปใช้ในการจัดกลุ่มของข้อมูลที่ไม่เคยมีการจัดกลุ่มมาก่อน โดยจะแบ่งกลุ่มข้อมูลจากความคล้าย เช่น การจัดกลุ่มลูกค้าจากพฤติกรรมการซื้อสินค้าของลูกค้าที่มีลักษณะคล้ายกันจะเป็นกลุ่มลูกค้าประเภทเดียวกัน Algorithms ที่ใช้ในการทำ Clustering ได้แก่

- K-means clustering เป็นอัลกอริทึมเทคนิคการเรียนรู้โดยไม่มีผู้สอนที่ง่ายที่สุด โดยอัลกอริทึม K-Means จะตัดแบ่ง (Partition) วัตถุออกเป็น K กลุ่ม แทนค่าแต่ละกลุ่มด้วยค่าเฉลี่ยของกลุ่ม ซึ่งใช้เป็นจุดศูนย์กลาง (Centroid) ของกลุ่มในการวัดระยะห่างของข้อมูลในกลุ่มเดียวกัน หากข้อมูลไหนใกล้ค่าจุดศูนย์กลางตัวไหนที่สุดอยู่กลุ่มนั้น ต่อมาหาค่าเฉลี่ย (Mean) แต่ละกลุ่มให้เป็นค่าจุดศูนย์กลางใหม่ ทำซ้ำจนค่าเฉลี่ยหรือจุดศูนย์กลางในแต่ละกลุ่มจะไม่เปลี่ยนแปลง
- Hierarchical clustering เป็นเทคนิคที่นิยมใช้กันมากในการจัดกลุ่ม Case หรือจัดกลุ่มตัวแปร โดยในกรณีที่ใช้ในการแบ่ง Case นั้น จำนวน Case ต้องไม่มากนัก (จำนวน Case ควรต่ำกว่า 200 ถ้าตั้งแต่ 200 ขึ้นไปใช้ K-Means Cluster) และจำนวนตัวแปรต้องไม่มากเช่นกันไม่จำเป็นต้องทราบจำนวนกลุ่มมาก่อนไม่จำเป็นเป็นต้องทราบว่าตัวแปรใดหรือ Case ใดอยู่กลุ่มใดก่อน ตัวอย่างอัลกิรอทึมที่เป็นแบบไฮราคิคอล ได้แก่ อัลกอริทึมแอกโกเมอเรทีพ (Agglomerative) ซึ่งเทคนิคนี้เริ่มจากข้อมูลจะถูกนับเป็นหนึ่งคลัสเตอร์ จากนั้นจะคำนวณหาระยะ ถ้าใกล้กันจะถูกรวมกันและทำวนซ้ำจนกลายเป็นหนึ่งคลัสเตอร์ในที่สุด
- Biclustering เป็นเทคนิคที่สามารถจัดกลุ่มแถวและคอลัมน์ของเมทริกซ์ได้พร้อมกัน โดยการจัดกลุ่มแนวนอนจะมีลักษณะการจัดกลุ่มเหมือน Clustering ปกติ ส่วนการจัดกลุ่มแนวตั้งจะเป็นการจัดกลุ่มฟีเจอร์ ซึ่งช่วยแสดงให้เห็นฟีเจอร์ที่สำคัญของแต่ละกลุ่ม สำหรับลักษณะการทำงาน คือ จะสลับกันจัดกลุ่มแนวนอนและแนวตั้งผลัดกันจนได้ข้อมูลรวมกันเป็นกลุ่มก้อนมากที่สุด

2.3 การสร้างฟีเจอร์ใหม่และตัดฟีเจอร์ที่ไม่เกี่ยวข้อง (Feature Engineering)

คือ การบวนการใช้ความรู้ Domain Knowledge ในการสร้างฟีเจอร์ใหม่ขึ้นมา ตัดฟีเจอร์ที่ไม่เกี่ยวข้องทิ้งไป เพื่อช่วยทำให้อัลกอริทึมเรียนรู้ได้ดีขึ้น รวมถึงการสร้างคุณลักษณะใหม่จากคุณลักษณะเดิม อาจนำคุณลักษณะเดิมมาคำนวณเพื่อให้เป็นคุณลักษณะใหม่ การเติมข้อมูลที่ขาด (Imputation) การจัดการข้อมูลโดด (Handling Outliers) การแปลงข้อมูล (Log Transform) การใช้งานข้อมูลวันที่ (Extracting Date) การแปลงข้อมูล Category เป็นหลาย Column (One-Hot Encoding / Embedding) การจัดกลุ่มของข้อมูล (Grouping Operations) และ Scaling (Normalize, Standardize) เพื่อเป็นประโยชน์ต่อการจัดกลุ่ม

2.4 การวิเคราะห์ข้อความ (Text Analysis)

คือ กระบวนการที่กระทำกับข้อความ เพื่อค้นหารูปแบบ แนวทาง และความสัมพันธ์ที่ซ่อนอยู่ในชุดข้อความนั้น โดยอาศัยหลักสถิติ การเรียนรู้ของเครื่อง หลักคณิตศาสตร์ หลักการประมวลเอกสาร (Document Processing) หลักการประมวลผลข้อความ และหลักการประมวลผลภาษาธรรมชาติ (Natural Language Processing)

2.5 งานวิจัยที่เกี่ยวข้องกับการแบ่งกลุ่มลูกค้าด้วยอัลกอริธึมที่ใช้ในการจัดกลุ่มของข้อมูล

ในปัจจุบันมีงานวิจัยที่ศึกษาเทคนิคการแบ่งกลุ่มลูกค้าและคุณลักษณะเด่น โดยใช้เทคนิคการทำคลัสเตอร์ ตัวอย่างเช่น

2.5.1 บทความวิจัยเรื่อง Data mining for the online retail industry: A case study of RFM model-based customer segmentation using data mining โดย Daqing Chen, Sai Laing Sain & Kun Guo (1)

งานวิจัยนี้นำเสนอวิธีการแบ่งกลุ่มฐานลูกค้า เพื่อทำให้ธุรกิจเข้าใจลูกค้ามากยิ่งขึ้นและเพื่อเพิ่มประสิทธิภาพการดูแลลูกค้า แบบให้ลูกค้าเป็นศูนย์กลาง โดยใช้วิธีพิจารณา Customer Value จาก RFM Model ใช้ K-means algorithm และ Decision Tree algorithm เพื่อจำแนกลูกค้าออกเป็นกลุ่ม

การทดลองงานวิจัยนี้ใช้ Online Retail Dataset เลือกสนใจเฉพาะข้อมูลรายการสินค้าที่เกิดในประเทศอังกฤษ (United Kingdom) เท่านั้น มาใช้ในการวิเคราะห์ข้อมูล ผลลัพธ์ที่ได้ คือ ทำให้เห็นถึงกลุ่มลูกค้าที่มีลักษณะเฉพาะแตกต่างกัน และสามารถนำกลยุทธ์ทางการตลาดไปใช้เหมาะสมกับลูกค้าในแต่ละกลุ่ม

2.5.2 **บทความวิจัยเรื่อง** A Novel Approach for Customer Segmentation Based on Biclustering **โดย** Xiaohui Hu, Xiaosheng Wu, Jianlin Chen,Yu Xiao, Yun Xue, TieChen Li **และคณะ** (2)

งานวิจัยนี้นำเสนอวิธีการแบ่งกลุ่มฐานลูกค้า เพื่อที่จะช่วยเพิ่มประสิทธิภาพให้กับการบริหารลูกค้าสัมพันธ์ ใช้ chi-square statistical analysis ในการเลือก attribute ,ใช้ K-means algorithm เพื่อวัดค่าแต่ละ attribute , ใช้ DBSCAN algorithm เพื่อจำแนกลูกค้าออกเป็นกลุ่ม และใช้ Biclustering based on FP-tree algorithm เพื่อค้นหารายละเอียดเพิ่มเติมที่ซ่อนอยู่ในแต่ละ row และ column ของ matrix งานวิจัยนี้หากเปรียบเทียบกับวิธีแบ่งกลุ่มฐานลูกค้าแบบ biclustering based on Apriori และ biclustering on Fp-tree algorithm นั้น Fp-tree สามารถทำได้มีประสิทธิภาพมากกว่าสำหรับ dataset ที่มีขนาดใหญ่

การทดลองงานวิจัยนี้ใช้ airline company dataset ผลลัพธ์ที่ได้ คือ Biclustering algorithm สามารถทำการแบ่งกลุ่มลูกค้าได้อย่างแม่นยำมากยิ่งขึ้น หากเปรียบเทียบวิธีแบ่งกลุ่มฐานลูกค้าแบบ biclustering based on Apriori และ biclustering on Fp-tree algorithm นั้น Fp-tree ได้รับความนิยมมากกว่า ในอนาคตสามารถพัฒนางานวิจัยฉบับนี้ได้โดยเพิ่มการทำ Feature Selection เพื่มช่วยเลือก Feature ที่มีนัยสำคัญกับข้อมูลก่อนนำเข้าโมเดล

2.5.3 **บทความวิจัยเรื่อง** A biclustering-based method for market segmentation using customer pain points **โดย** Binda Wang, Yunwen Miao, Hongya Zhao, Jian Jin **และ** Yizeng Chen (3)

งานวิจัยฉบับนี้ศึกษาเกี่ยวกับการแบ่งกลุ่มลูกค้าจากข้อมูล Customer Pain Points โดยใช้เทคนิค Biclustering เพื่อระบุลูกค้าที่มีลักษณะเหมือนกันออกเป็นกลุ่มย่อยได้ (identify homogenous subgroups)

ผลลัพธ์ของงานวิจัยฉบับนี้ทำให้องค์กรสามารถแบ่งกลุ่มลูกค้า จากสาเหตุที่ทำให้ลูกค้าไม่พอใจผลิตภัณฑ์หรือบริการ และเมื่อทราบถึงปัญหาก็สามารถนำไปพัฒนาผลิตภัณฑ์หรือบริการในอนาคตได้

2.5.4 **บทความวิจัยเรื่อง** Customer Segmentation based on RFM model and Clustering Techniques With K-Means Algorithm **โดย** Ina Maryani **และคณะ** (4)

งานวิจัยฉบับนี้ศึกษาเกี่ยวกับการแบ่งกลุ่มฐานลูกค้า ของ Nine Reload Credit ทำ data mining โดยใช้ RFM model และใช้ Clustering technique การแบ่งกลุ่มฐานลูกค้านั้น จะช่วยให้ Company สามารถตัดสินใจทำ Marketing Strategy ได้อย่างเหมาะสมมากยิ่งขึ้น

ผลลัพธ์ของงานวิจัยฉบับนี้พบว่า Company สามารถแบ่งกลุ่มฐานลูกค้าได้ และทำให้สามารถรักษาฐานลูกค้าเดิมไว้ได้อีกด้วย งานวิจัยนี้ได้นำข้อมูล Transaction จำนวน 82,648 รายการ มาผ่านกระบวนการทำ RFM model จากนั้นทำ Clustering Analysis ด้วย K-Means สำหรับงานในอนาคตจะนำ Feature selection มาช่วยลด Computation Time และช่วยเพิ่มประสิทธิภาพให้กับโมเดล อีกทั้งทดลองใช้ Biclustering techniques ในขั้นตอนการทำ Clustering Analysis แทนที่จากเดิมใช้ K-Means

2.5.5 บทความวิจัยเรื่อง Customer Segmentation using K-means Clustering โดย Tushar Kansal และคณะ (5)

งานวิจัยฉบับนี้ศึกษาเกี่ยวกับการแบ่งกลุ่มฐานลูกค้า โดยใช้ clustering algorithms ทั้งหมด 3 algorithms คือ K-means, Agglomerative, และ Meanshift จากนั้นทำการเปรียบเทียบผลในแต่ละ Clusters ที่ได้จากการทำในแต่ละ algorithm

ผลการวิจัยโดยใช้ clustering algorithm สามารถจัดกลุ่มลูกค้าได้เป็น 5 กลุ่ม ได้แก่ Careless, Careful, Standard, Target และ Sensible customer อีกทั้ง mean shift clustering สามารถค้นพบเพิ่มอีก 2 กลุ่ม คือ High buyers and frequent visitors และ High buyers and occasional visitors เนื่องจาก dataset ในงานวิจัยนี้เป็นข้อมูลที่ unlabelled จึงเลือกสนใจข้อมูลที่เป็น internal clustering validation เพราะ สามารถใช้ในการเลือก clustering algorithm และจัดกลุ่มข้อมูลลงใน cluster ได้อย่างถูกต้อง ในอนาคตสามารถพัฒนางานวิจัยฉบับนี้ได้โดยการทำ feature extraction เพื่อเพื่อลด redundant data สำหรับที่จะนำไปใช้ในการวิเคราะห์

แม้ว่าจะมีงานวิจัยจำนวนไม่น้อย ที่ศึกษาเกี่ยวกับการจัดกลุ่มลูกค้า การศึกษาเหล่านั้นอาจแบ่งเป็น 2 แนวทางได้แก่ แนวทางที่ 1 ผู้วิจัยจะอาศัยข้อมูลจากใบสั่งซื้อ เพื่อจัดกลุ่มข้อมูลโดยอาศัยความคล้ายคลึงกันของสินค้าที่สั่งซื้อของลูกค้าแต่ละราย ส่วนในแนวทางที่ 2 ผู้ศึกษาจะจัดกลุ่มโดยอาศัยแบบจำลองอาร์เอฟเอ็ม ซึ่งเป็นการจัดกลุ่มลูกค้า โดยจำแนกตามศักยภาพของลูกค้าผ่านแบบจำลองอาร์เอฟเอ็ม โดยวิเคราะห์จากใบสั่งซื้อ การแบ่งกลุ่มเช่นนี้ อาจทำให้เราไม่สามารถมองเห็นภาพรวมของลูกค้าได้ชัดเจนในทุกๆแง่มุมได้ดีพอ ในการศึกษาที่ผ่านมา ผู้วิจัยจึงทำการจัดกลุ่มลูกค้า โดยอาศัยข้อมูลจากทั้งสองส่วน อันได้แก่ ข้อมูลการสั่งซื้อสินค้าในใบสั่งซื้อ ร่วมกับการใช้แบบจำลองอาร์เอฟเอ็ม เพื่อวัดศักยภาพของลูกค้า มีประกอบร่วมกัน เพื่อหาแนวทางในการสร้างแบบจำลองการจัดกลุ่มลูกค้า เพื่อให้มีประสิทธิภาพในการจัดกลุ่มสูงสุด นอกจากนี้ ในการวิเคราะห์ข้อมูลการสั่งซื้อสินค้าในใบสั่งซื้อ ผู้วิจัยยังใช้เทคนิคการวิเคราะห์

ข้อความร่วมด้วย เพื่อช่วยในการจัดกลุ่มสินค้า เพื่อให้แบบจำลองสามารถจัดกลุ่มได้อย่างมีความแม่นยำมากขึ้น

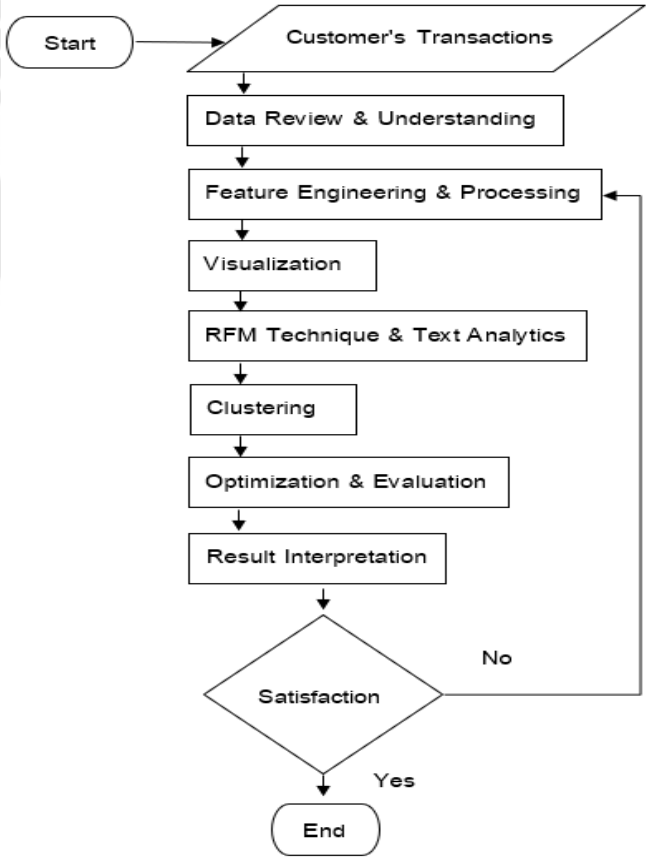
บทที่ 3
วิธีดำเนินการวิจัย

ในการวิจัยครั้งนี้ผู้วิจัยได้ดำเนินการตามขั้นตอนดังนี้

1. กระบวนการทำงานของแบบจำลอง
2. การนำเข้าข้อมูล ตรวจสอบข้อมูล และพิจารณาข้อมูล
3. การแปลงข้อมูลที่ได้ทำการเก็บรวบรวม
4. การวิเคราะห์ข้อมูลด้วยเทคนิคการแบ่งกลุ่มข้อมูล
5. การวัดผลลัพธ์จากการวิเคราะห์ข้อมูล

3.1 กระบวนการทำงานของแบบจำลอง

การทำวิจัยเรื่องการแบ่งกลุ่มลูกค้าจากข้อมูลการซื้อขายสินค้าโดยใช้เทคนิคคลัสเตอริง สามารถแสดงแผนผังลำดับขั้นตอนการดำเนินการได้ดังภาพประกอบที่ 1



ภาพประกอบ 1 แสดงกระบวนการทำงานของแบบจำลอง

จากภาพประกอบ 1 ได้อธิบายถึงกระบวนการสร้างแบบจำลองการแบ่งกลุ่มและการวัดผลลัพธ์จากการวิเคราะห์ข้อมูล โดยเริ่มจากขั้นตอนการนำเข้าข้อมูล (Load Data) การตรวจสอบข้อมูล การพิจารณานำข้อมูลมาใช้ในการวิเคราะห์ (Data Review & Understanding) การแปลงข้อมูลที่ได้ทำการเก็บรวบรวมมาให้กลายเป็นข้อมูลที่สามารถนำไปวิเคราะห์ได้ (Feature Engineering, Data Processing & New Features) การวิเคราะห์ข้อมูลด้วยเทคนิคการแบ่งกลุ่มข้อมูลแบบคลัสเตอริง(Clustering) และการวัดผลลัพธ์จากการวิเคราะห์ข้อมูล (Translated the results) หากผลลัพธ์ของแบบจำลองยังไม่เป็นที่น่าพอใจ จะกลับไปปรับการแปลงข้อมูล เพื่อให้ได้มาซึ่งแบบจำลองการแบ่งกลุ่มที่มีประสิทธิภาพดีที่สุด

3.2 การเก็บรวบรวมข้อมูล ตรวจสอบข้อมูล และพิจารณาข้อมูล

3.2.1 การเก็บรวบรวมข้อมูล

ในงานวิจัยนี้ได้ใช้ข้อมูลการซื้อขายสินค้าของร้านค้าปลีกในประเทศอังกฤษที่เกิดขึ้นในช่วงวันที่ 1 ธ.ค. 2553 ถึง 9 ธ.ค. 2554 ประกอบด้วย 8 แอททริบิวต์ มีจำนวนข้อมูลทั้งหมด 541,909 แถว ดังภาพประกอบที่ 2 และข้อมูลที่อยู่ในรูปแบบ File worksheet Microsoft Excel (.xlsx) ดังภาพประกอบที่ 3

Key Attributes information

Attribute Name	Summary	Description
InvoiceNo	Invoice number	Nominal, a 6-digit integral number uniquely assigned to each transaction. If this code starts with letter 'c', it indicates a cancellation
StockCode	Product (item) code	Nominal, a 5-digit integral number uniquely assigned to each distinct product
Description	Product (item) name	Nominal
Quantity	The quantities of each product (item) per transaction	Numeric
InvoiceDate	Invice Date and time	Numeric, the day and time when each transaction was generated
UnitPrice	Unit price	Numeric, Product price per unit in sterling
CustomerID	Customer number	Nominal, a 5-digit integral number uniquely assigned to each customer
Country	Country name	Nominal, the name of the country where each customer resides

ภาพประกอบ 2 แสดงแอททริบิวต์ที่ใช้สำหรับสร้างแบบจำลอง

InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	01-12-10 8:26	2.55	17850	United Kingdom
536365	71053	WHITE METAL LANTERN	6	01-12-10 8:26	3.39	17850	United Kingdom
536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	01-12-10 8:26	2.75	17850	United Kingdom
536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	01-12-10 8:26	3.39	17850	United Kingdom
536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	01-12-10 8:26	3.39	17850	United Kingdom
536365	22752	SET 7 BABUSHKA NESTING BOXES	2	01-12-10 8:26	7.65	17850	United Kingdom
536365	21730	GLASS STAR FROSTED T-LIGHT HOLDER	6	01-12-10 8:26	4.25	17850	United Kingdom
536366	22633	HAND WARMER UNION JACK	6	01-12-10 8:28	1.85	17850	United Kingdom
536366	22632	HAND WARMER RED POLKA DOT	6	01-12-10 8:28	1.85	17850	United Kingdom
536367	84879	ASSORTED COLOUR BIRD ORNAMENT	32	01-12-10 8:34	1.69	13047	United Kingdom
536367	22745	POPPY'S PLAYHOUSE BEDROOM	6	01-12-10 8:34	2.1	13047	United Kingdom
536367	22748	POPPY'S PLAYHOUSE KITCHEN	6	01-12-10 8:34	2.1	13047	United Kingdom
536367	22749	FELTCRAFT PRINCESS CHARLOTTE DOLL	8	01-12-10 8:34	3.75	13047	United Kingdom
536367	22310	IVORY KNITTED MUG COSY	6	01-12-10 8:34	1.65	13047	United Kingdom
536367	84969	BOX OF 6 ASSORTED COLOUR TEASPOONS	6	01-12-10 8:34	4.25	13047	United Kingdom
536367	22623	BOX OF VINTAGE JIGSAW BLOCKS	3	01-12-10 8:34	4.95	13047	United Kingdom
536367	22622	BOX OF VINTAGE ALPHABET BLOCKS	2	01-12-10 8:34	9.95	13047	United Kingdom
536367	21754	HOME BUILDING BLOCK WORD	3	01-12-10 8:34	5.95	13047	United Kingdom
536367	21755	LOVE BUILDING BLOCK WORD	3	01-12-10 8:34	5.95	13047	United Kingdom
536367	21777	RECIPE BOX WITH METAL HEART	4	01-12-10 8:34	7.95	13047	United Kingdom
536367	48187	DOORMAT NEW ENGLAND	4	01-12-10 8:34	7.95	13047	United Kingdom
536368	22960	JAM MAKING SET WITH JARS	6	01-12-10 8:34	4.25	13047	United Kingdom
536368	22913	RED COAT RACK PARIS FASHION	3	01-12-10 8:34	4.95	13047	United Kingdom

ภาพประกอบ 3 ตัวอย่างข้อมูลที่ใช้สำหรับสร้างแบบจำลอง

3.2.2 **ตรวจสอบข้อมูลและพิจารณาข้อมูลมาใช้ในการวิเคราะห์**

งานวิจัยนี้ใช้ภาษาไพทอน (Python) ในการวิเคราะห์ข้อมูลและทำ Machine Learning เริ่มต้นด้วยการนำเข้าโมดูลสำคัญสำหรับการสร้างแบบจำลอง ดังภาพประกอบที่ 4 ต่อมานำเข้าไฟล์ข้อมูลและข้อมูลที่ใช้สำหรับสร้างแบบจำลอง ดังภาพประกอบที่ 5

▾ Import modules

```
[ ]  import pandas as pd
     import numpy as np

     import time, warnings
     import datetime as dt

     #visualizations
     import matplotlib.pyplot as plt
     from pandas.plotting import scatter_matrix
     %matplotlib inline
     import seaborn as sns

     warnings.filterwarnings("ignore")
```

ภาพประกอบ 4 แสดงตัวอย่างโค้ดนำเข้าโมดูลสำคัญที่ใช้สำหรับสร้างแบบจำลอง

```
#load the dataset
retail_df = pd.read_excel("Online Retail.xlsx") #2010 - 2011
```

```
[27] retail_df.shape
```

```
(541909, 8)
```

```
[28] retail_df.head()
```

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom

ภาพประกอบ 5 แสดงตัวอย่างโค้ดนำเข้าไฟล์ข้อมูลและข้อมูลที่ใช้สำหรับสร้างแบบจำลอง

เริ่มทำ EDA เพื่อหาข้อมูลเชิงลึกจากข้อมูล Transaction Data โดยใช้ Pandas, Numpy และ Dataprep.eda ดังภาพประกอบที่ 6 - 16

```
retail_df.describe()
```

	Quantity	UnitPrice	CustomerID
count	541909.000000	541909.000000	406829.000000
mean	9.552250	4.611114	15287.690570
std	218.081158	96.759853	1713.600303
min	-80995.000000	-11062.060000	12346.000000
25%	1.000000	1.250000	13953.000000
50%	3.000000	2.080000	15152.000000
75%	10.000000	4.130000	16791.000000
max	80995.000000	38970.000000	18287.000000

ภาพประกอบ 6 แสดงค่าสถิติของข้อมูลแต่ละคอลัมน์

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
299983	A563186	B	Adjust bad debt	1	2011-08-12 14:51:00	-11062.06	NaN	United Kingdom
299984	A563187	B	Adjust bad debt	1	2011-08-12 14:52:00	-11062.06	NaN	United Kingdom

ภาพประกอบ 7แสดงตัวอย่างใบสั่งซื้อที่ถูกแก้ไข

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
141	C536379	D	Discount	-1	2010-12-01 09:41:00	27.50	14527.0	United Kingdom
154	C536383	35004C	SET OF 3 COLOURED FLYING DUCKS	-1	2010-12-01 09:49:00	4.65	15311.0	United Kingdom
235	C536391	22556	PLASTERS IN TIN CIRCUS PARADE	-12	2010-12-01 10:24:00	1.65	17548.0	United Kingdom

ภาพประกอบ 8 แสดงตัวอย่างใบสั่งซื้อที่ถูกยกเลิก

```
[58] from dataprep.eda import plot
     plot(retail_df)
```

Hide Stats and Insights

Dataset Statistics

Number of Variables	12
Number of Rows	541909
Missing Cells	136534
Missing Cells (%)	2.1%
Duplicate Rows	5268
Duplicate Rows (%)	1.0%
Total Size in Memory	217.7 MB
Average Row Size in Memory	421.2 B
Variable Types	Categorical: 7 Numerical: 4 DateTime: 1

Dataset Insights

CustomerID has 135080 (24.93%) missing values	Missing
Quantity is skewed	Skewed
UnitPrice is skewed	Skewed
TotalAmount is skewed	Skewed
InvoiceNo has a high cardinality: 25900 distinct values	High Cardinality
StockCode has a high cardinality: 4070 distinct values	High Cardinality
Description has a high cardinality: 4223 distinct values	High Cardinality
date has a high cardinality: 305 distinct values	High Cardinality
time has a high cardinality: 774 distinct values	High Cardinality
date has constant length 10	Constant Length

ภาพประกอบ 9 แสดงค่าสถิติและข้อมูลเชิงลึกของแต่ละคอลัมน์

```
[58] from dataprep.eda import plot
     plot(retail_df)
```

Hide Stats and Insights

Dataset Statistics

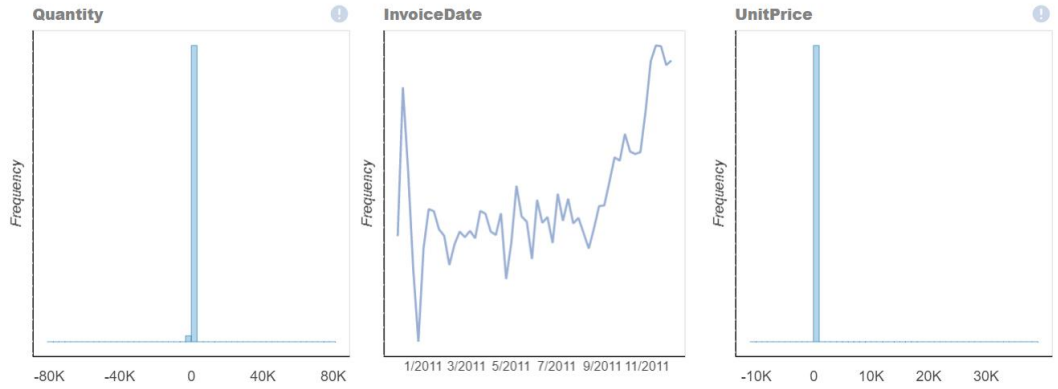
Number of Variables	12
Number of Rows	541909
Missing Cells	136534
Missing Cells (%)	2.1%
Duplicate Rows	5268
Duplicate Rows (%)	1.0%
Total Size in Memory	217.7 MB
Average Row Size in Memory	421.2 B
Variable Types	Categorical: 7 Numerical: 4 DateTime: 1

Dataset Insights

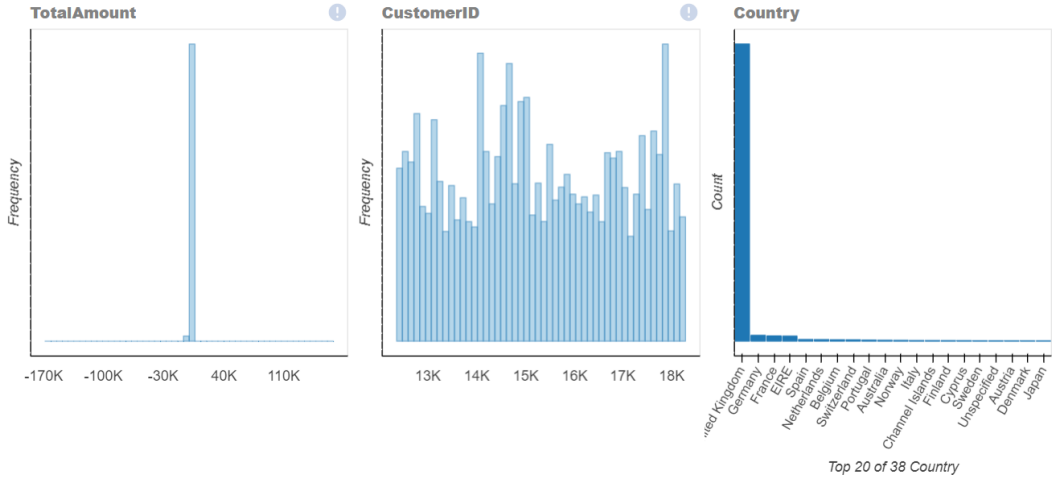
time has constant length 8	Constant Length
Quantity has 10624 (1.96%) negatives	Negatives
TotalAmount has 9290 (1.71%) negatives	Negatives

ภาพประกอบ 10 แสดงค่าสถิติและข้อมูลเชิงลึกของแต่ละคอลัมน์

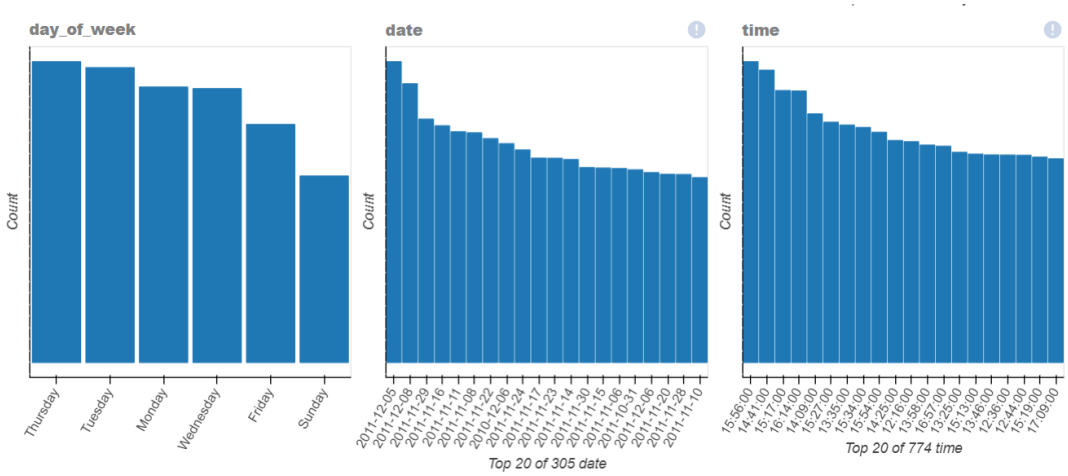
ภาพประกอบ 11 แสดงการตรวจสอบการกระจายข้อมูลของคอลัมน์รหัสใบสั่งซื้อ รหัสสต็อก และชื่อสินค้า



ภาพประกอบ 12 แสดงการตรวจสอบการกระจายข้อมูลของคอลัมน์ปริมาณการซื้อ วันที่ในใบสั่งซื้อและราคาต่อหน่วย

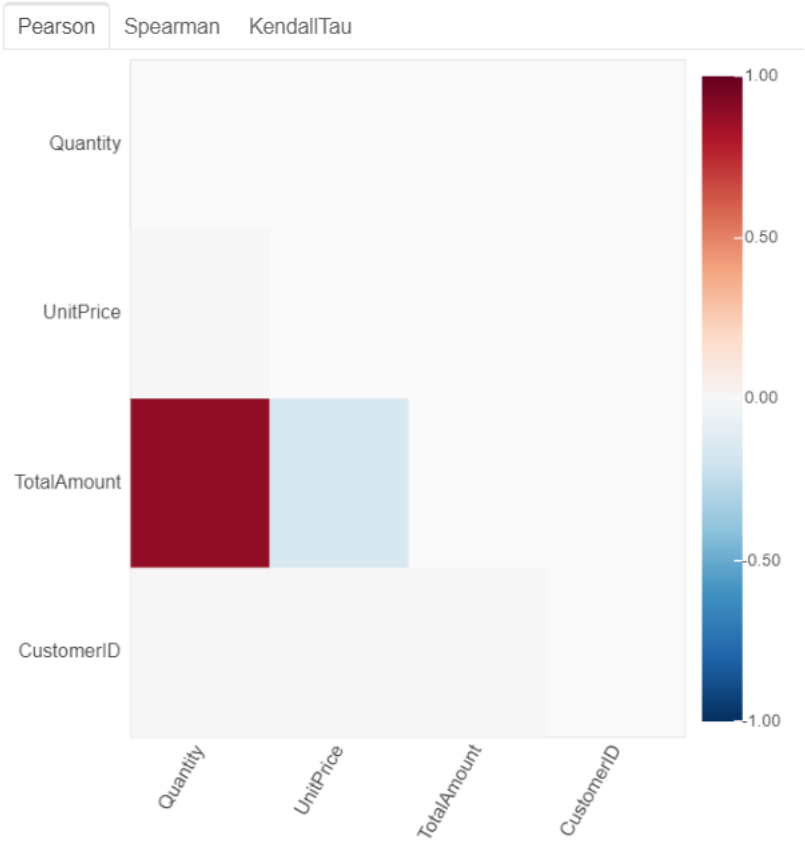


ภาพประกอบ 13 แสดงการตรวจสอบการกระจายข้อมูลของคอลัมน์จำนวนเงินรวม รหัสลูกค้าและประเทศ



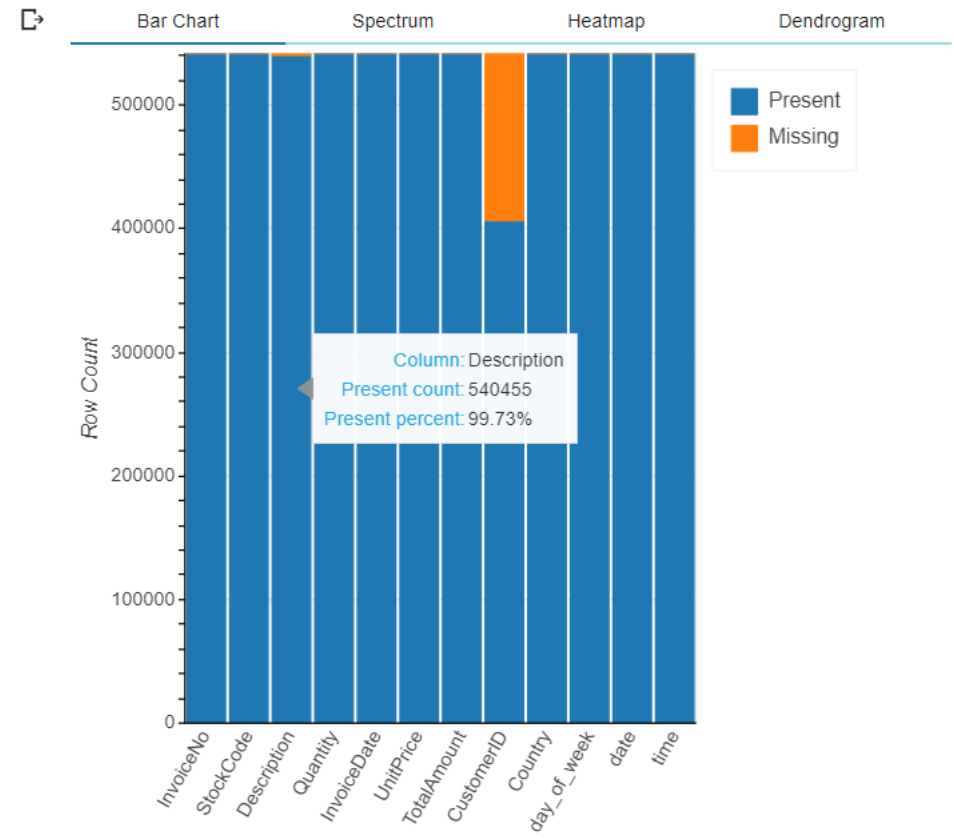
ภาพประกอบ 14 แสดงการตรวจสอบการกระจายข้อมูลของแต่ละวันในสัปดาห์และเวลา

```
[60] from dataprep.eda import plot_correlation
     plot_correlation(retail_df)
```



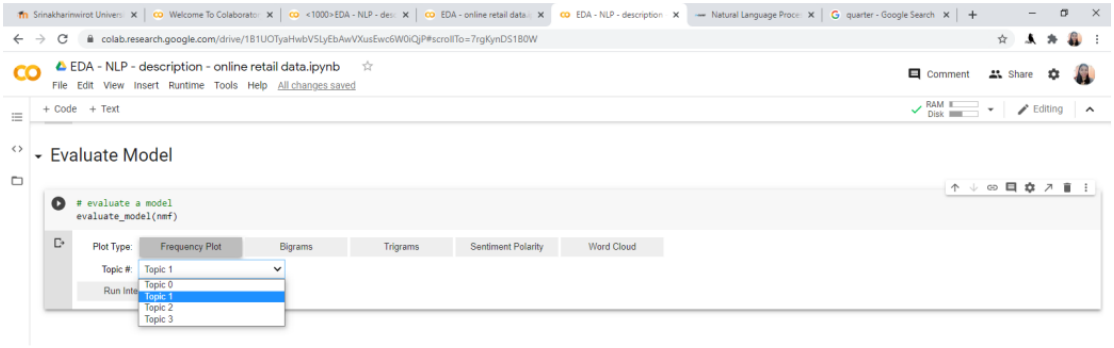
ภาพประกอบ 15 แสดงการวิเคราะห์ความสัมพันธ์ของแต่ละคอลัมน์

```
from dataprep.eda import plot_missing
plot_missing(retail_df)
```



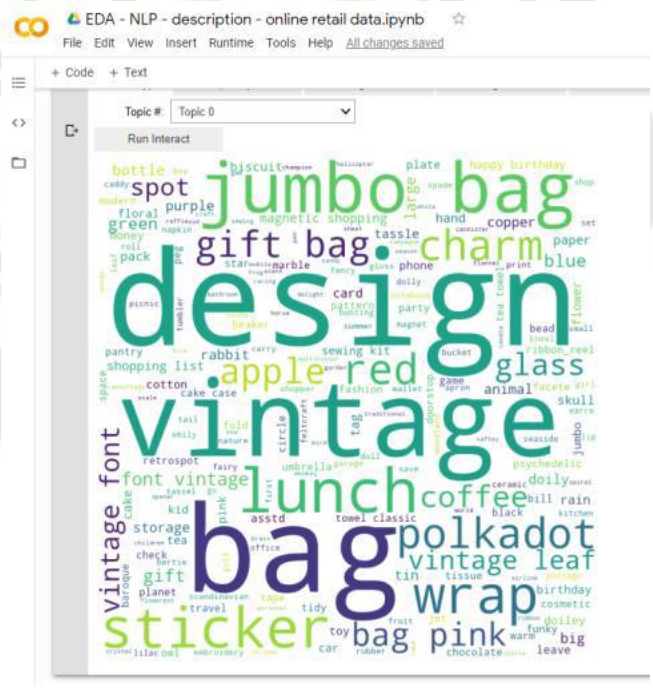
ภาพประกอบ 16 แสดงการตรวจสอบข้อมูลสูญหายของแต่ละคอลัมน์

ทำ EDA กับข้อมูลคอลัมน์ชื่อสินค้าซึ่งเป็นคอลัมน์ชื่อสินค้า เพื่อพยายามแบ่งกลุ่มสินค้าออกเป็นกลุ่ม แล้วตั้งชื่อหมวดหมู่ (Topic) ให้แต่ละกลุ่ม และหา Frequency Word โดยใช้ Pycaret ดังภาพประกอบที่ 17

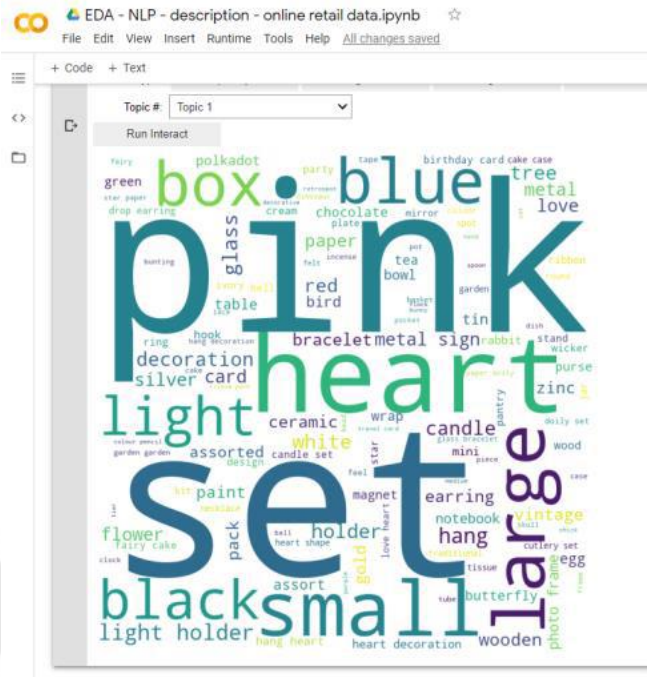


ภาพประกอบ 17 แสดงการเลือกหมวด

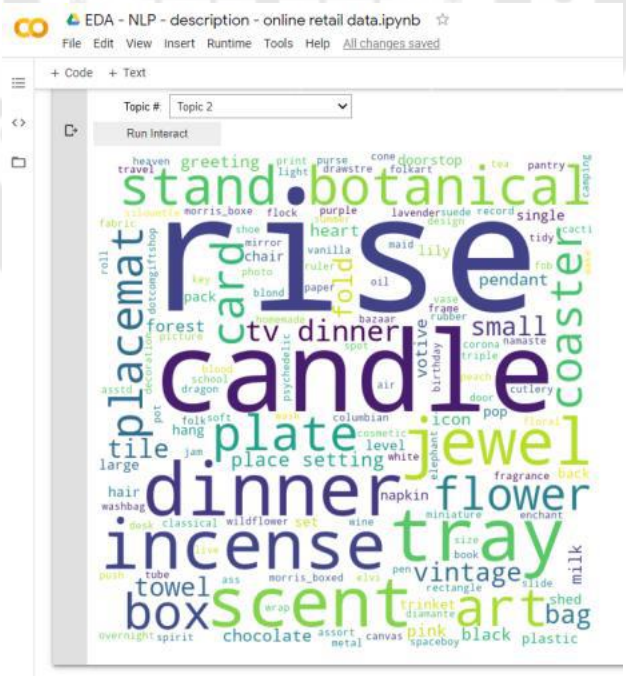
ผลลัพธ์จากการทำ EDA ชื่อสินค้าที่เกิด Transaction ตลอดทั้งปี แยกได้เป็น 4 หมวด ดังภาพประกอบที่ 18 - 21



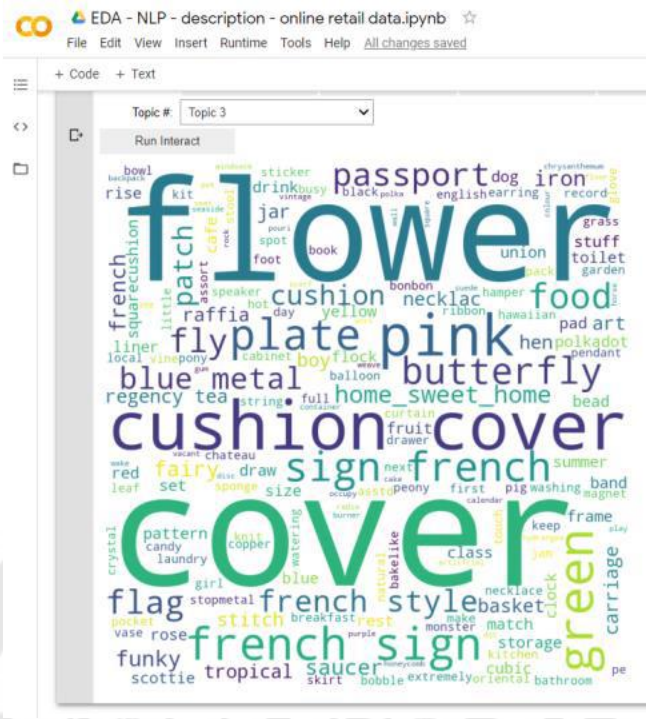
ภาพประกอบ 18 แสดงหมวดที่ 0



ภาพประกอบ 19 แสดงหมวดที่ 1



ภาพประกอบ 20 แสดงหมวดที่ 2



ภาพประกอบ 21 แสดงหมวดที่ 3

ผลลัพธ์ที่ได้จากการทำ Exploratory Data Analysis (EDA)

1. ช่วงเวลาที่เกิด Transaction เยอะที่สุดตั้งแต่ 12:00 pm - 17:00 pm (ช่วงบ่าย)
2. ต้นเดือนและปลายเดือนพฤศจิกายน พ.ศ. 2554 เป็นช่วงที่เกิด Transaction มากที่สุด
3. ลูกค้าจะซื้อสินค้ามากที่สุดในวันพฤหัสบดี, วันอังคาร, วันจันทร์, วันพุธ, วันศุกร์ และวันอาทิตย์ ตามลำดับ (โดยตลอดปีจะไม่มีรายการซื้อขายที่เกิดในวันเสาร์)
4. ลูกค้าที่ซื้อจำนวนเยอะอยู่ที่ประเทศ Netherlands และรองลงมาคืออยู่ที่ประเทศ Australia
5. ลูกค้าส่วนใหญ่อยู่ในประเทศ United Kingdom (UK)
6. 10 รายการสินค้าขายดี จากรายการซื้อขายตลอดทั้งปี คือ
    (1) PAPER CRAFT, LITTLE BIRDIE
    (2) REGENCY CAKESTAND 3 TIER
    (3) WHITE HANGING HEART T-LIGHT HOLDER
    (4) JUMBO BAG RED RETROSPOT
    (5) MEDIUM CERAMIC TOP STORAGE JAR
    (6) POSTAGE
    (7) PARTY BUNTING

(8) ASSORTED COLOUR BIRD ORNAMENT

(9) Manual

(10) RABBIT NIGHT LIGHT

7. 10 รายการสินค้าที่ขายได้ไม่ดี จากรายการซื้อขายตลอดทั้งปี คือ

(1) PADS TO MATCH ALL CUSHIONS

(2) HEN HOUSE W CHICK IN NEST

(3) SET 12 COLOURING PENCILS DOILEY

(4) BIRD BOX CHRISTMAS TREE DECORATION

(5) PINK CRYSTAL GUITAR PHONE CHARM

(6) PURPLE FRANGIPANI HAIRCLIP

(7) CAT WITH SUNGLASSES BLANK CARD

(8) HAPPY BIRTHDAY CARD TEDDY/CAKE

(9) 60 GOLD AND SILVER FAIRY CAKE CASES

(10) GREEN POP ART MAO CUSHION COVER

8. สินค้าขายดีในแต่ละไตรมาสได้แก่

MEDIUM CERAMIC TOP STORAGE JAR (Q1)

PICNIC BASKET WICKER 60 PIECES (Q2)

SET OF TEA COFFEE SUGAR TINS PANTRY (Q3)

PAPER CRAFT, LITTLE BIRDIE (Q3)

สินค้าขายไม่ดีในแต่ละไตรมาสได้แก่

DANISH ROSE TRINKET TRAYS (Q1)

POSTAGE (Q2)

CHRISTMAS CRAFT WHITE FAIRY (Q3)

SET OF 3 HANGING OWLS OLLIE BEAK (Q3)

9. 10 รายลูกค้าที่ซื้อเยอะที่สุดได้แก่รหัสลูกค้าดังต่อไปนี้

(1) 16446

(2) 24816401

(3) 30070457

(4) 24025062

(5) 3047321

(6) 13685339

(7) 21362047

(8) 20602478

(9) 4200795

(10) 12564016

3.3 การแปลงข้อมูลที่ได้ทำการเก็บรวบรวม

เนื่องด้วยข้อมูลในคอลัมน์วันที่ในใบสั่งซื้อยังอยู่ในรูปแบบที่ไม่พร้อมใช้งาน จึงจำเป็นต้องแบ่งออกเป็นคอลัมน์ชื่อวันของสัปดาห์และเวลา เพื่อที่จะสามารถนำไปทำในขั้นตอน EDA ได้ หากไม่ทำการแปลงข้อมูลก็จะไม่สามารถหาข้อมูลเชิงลึกจากข้อมูลในคอลัมน์ วันที่ในใบสั่งซื้อได้ ดังภาพประกอบที่ 22 -23

```
[55] retail_df.head()
```

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	TotalAmount	CustomerID	Country	day_of_week	date	time
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	15.30	17850.0	United Kingdom	Wednesday	2010-12-01	08:26:00
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	20.34	17850.0	United Kingdom	Wednesday	2010-12-01	08:26:00
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	22.00	17850.0	United Kingdom	Wednesday	2010-12-01	08:26:00
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	20.34	17850.0	United Kingdom	Wednesday	2010-12-01	08:26:00
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	20.34	17850.0	United Kingdom	Wednesday	2010-12-01	08:26:00

ภาพประกอบ 22 แสดงการแปลงข้อมูลวันที่ในใบสั่งซื้อ

```
retail_df.head()
```

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	TotalAmount	CustomerID	Country	day_of_week	date	time
19632	537879	22114	HOT WATER BOTTLE TEA AND SYMPATHY	12	2010-12-09 08:34:00	3.95	47.4	14243	United Kingdom	Thursday	2010-12-09	08:34:00
19633	537879	22835	HOT WATER BOTTLE I AM SO POORLY	8	2010-12-09 08:34:00	4.65	37.2	14243	United Kingdom	Thursday	2010-12-09	08:34:00
19634	537879	85150	LADIES & GENTLEMEN METAL SIGN	6	2010-12-09 08:34:00	2.55	15.3	14243	United Kingdom	Thursday	2010-12-09	08:34:00
19635	537879	85048	15CM CHRISTMAS GLASS BALL 20 LIGHTS	4	2010-12-09 08:34:00	7.95	31.8	14243	United Kingdom	Thursday	2010-12-09	08:34:00
19636	537879	21524	DOORMAT SPOTTY HOME SWEET HOME	2	2010-12-09 08:34:00	7.95	15.9	14243	United Kingdom	Thursday	2010-12-09	08:34:00

ภาพประกอบ 23 แสดงข้อมูลชื่อสินค้า

การสำรวจข้อมูลชื่อสินค้า โดยการแปลงให้กลายเป็นตัวอักษรพิมพ์เล็กและหาหน้าที่ของคำ ดังภาพประกอบที่ 24 และ 25

```
[49] input_str = "HOT WATER BOTTLE TEA AND SYMPATHY"
     input_str = input_str.capitalize()

     from textblob import TextBlob
     result = TextBlob(input_str)
     print(result.tags)
```

```
[('Hot', 'NNP'), ('water', 'NN'), ('bottle', 'NN'), ('tea', 'NN'), ('and', 'CC'), ('sympathy', 'NN')]
```

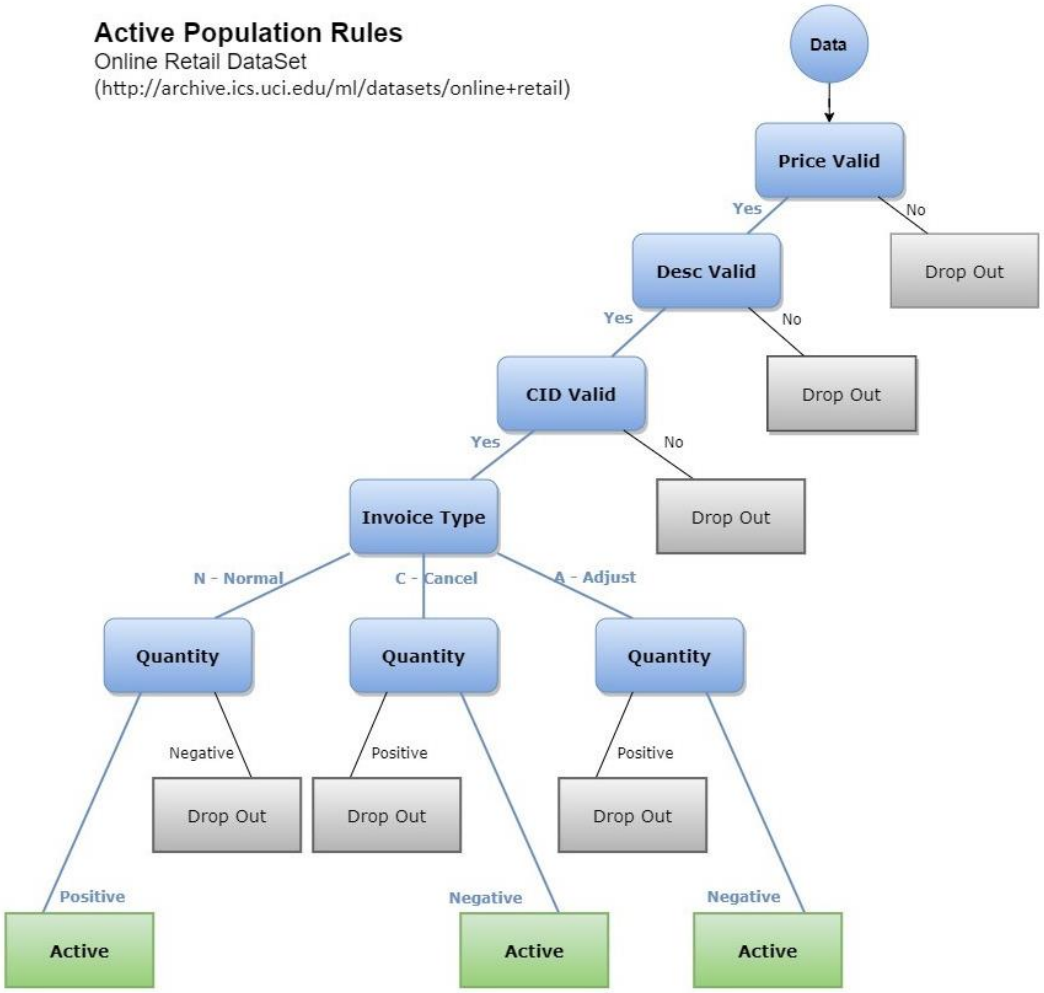
ภาพประกอบ 24 แสดงการแปลงข้อมูลชื่อสินค้า

```
[50] example = "HOT WATER BOTTLE TEA AND SYMPATHY"
     example = example.capitalize()
     word_tokens = word_tokenize(example)
     pos = nltk.pos_tag(word_tokens)
     selective_pos = ['NN','VB']
     selective_pos_words = []
     for word,tag in pos :
          if tag in selective_pos:
              selective_pos_words.append((word,tag))
     print(selective_pos_words)
```

```
[('water', 'NN'), ('bottle', 'NN'), ('tea', 'NN'), ('sympathy', 'NN')]
```

ภาพประกอบ 25 แสดงการแปลงข้อมูลชื่อสินค้า (ต่อ)

สรุปขั้นตอนการทำความสะอาดข้อมูลกับชุดข้อมูลที่ใช้ในการวิจัย ดังภาพประกอบที่ 26 เริ่มต้นจากการตรวจสอบราคาสินค้า ชื่อสินค้า รหัสลูกค้า ถ้าตรวจสอบแล้วถูกต้อง จะเก็บไว้ใช้ในงานวิจัย หากไม่ถูกต้องจะทำการตัดทิ้ง โดยได้ตัดข้อมูลใบคำสั่งซื้อที่มีการยกเลิกหรือแก้ไข 10,624 รายการ, ข้อมูลที่ไม่มีรหัสลูกค้า 133,361 รายการ โดยไม่นำมาร่วมทำการวิเคราะห์ในงานวิจัยนี้



ภาพประกอบ 26 แสดง Active Population Rules

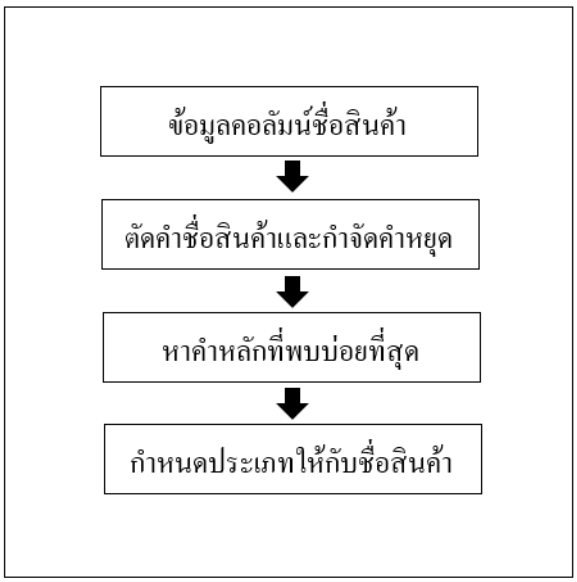
ที่ ม า : Evangelos T. (2020). Customer Segmentation using RFM Analysis in Python (ออนไลน์). สืบค้นจาก : https://www.linkedin.com/pulse/customer-segmentation-using-rfm-analysis-python-evangelos-tzimopoulos/ [8 พฤศจิกายน 2563]

ผู้วิจัยได้ทำการสร้างฟีเจอร์ใหม่ด้วยเทคนิคการวิเคราะห์ข้อความเพื่อสร้างกลุ่มสินค้า การจัดเตรียมข้อมูล ผู้วิจัยทำการกำหนดช่วงวันที่ของข้อมูลที่ใช้ในการวิเคราะห์ข้อมูลเท่ากับ 1 ปี ผู้วิจัยวิเคราะห์ข้อความจากชื่อสินค้า ใช้ข้อมูลคอลัมน์ชื่อสินค้า เลือกเฉพาะสินค้าที่ไม่ซ้ำโดยมีรหัสสินค้าเป็นตัวบ่งบอก ดังตารางที่ 1

ตาราง 1 ตัวอย่างชื่อสินค้า

ลำดับ	ชื่อสินค้าตัวอย่าง
1	WHITE HANGING HEART T-LIGHT HOLDER
2	WHITE METAL LANTERN
3	CREAM CUPID HEARTS COAT HANGER
…	STRAWBERRY CHARLOTTE BAG
3958	PAPER CRAFT, LITTLE BIRDIE

เนื่องจากชื่อสินค้าอยู่ในรูปแบบภาษาปกติของมนุษย์จึงจำเป็นต้องแปลงชื่อสินให้อยู่ในรูปแบบที่คอมพิวเตอร์สามารถเข้าใจ โดยผ่านกระบวนการวิเคราะห์ข้อความ สำหรับงานวิจัยนี้มีขั้นตอนดังภาพที่ 27



ภาพประกอบ 27 กระบวนการวิเคราะห์ข้อความ

ขั้นตอนการวิเคราะห์ข้อความดังภาพที่ 27 เริ่มจากนำข้อมูลจากคอลัมน์สินค้ามาตัดคำชื่อสินค้าภาษาอังกฤษและกำจัดคำหยุด โดยอาศัยพจนานุกรมคำศัพท์มาช่วยในกระบวนการตัดคำ สำหรับงานวิจัยนี้ผู้วิจัยใช้เครื่องมือประมวลภาษาธรรมชาติ โมดูล NLTK (Natural Language Toolkit) จากนั้นหาคำหลักที่พบบ่อยที่สุด สำหรับนำไปใช้กำหนดประเภทให้กับชื่อสินค้า ผู้วิจัยใช้

เทคนิค K-Means Clustering ในการแบ่งกลุ่ม เพื่อกำหนดประเภทให้กับชื่อสินค้า ใช้ Silhouette Score เป็นเทคนิคที่ใช้วัดตัวอย่างนั้นๆมีความเหมือนกับกลุ่มที่อยู่มากเพียงใด เมื่อเทียบกับกลุ่มอื่นๆ ผู้วิจัยเลือกจำนวนกลุ่มที่เหมาะสมที่สุดเท่ากับ 5 ถึงแม้จากการทดลอง จำนวนกลุ่มที่เท่ากับ 7 และ 9 จะได้คะแนนเยอะกว่า แต่ที่ไม่เลือก เพราะเมื่อแบ่งกลุ่มออกมาคลัสเตอร์บางกลุ่มมีองค์ประกอบน้อยมาก ต่อมาทำการรวมข้อมูลประเภทสินค้าโดยจัดกลุ่มตามแอตทริบิวต์รหัสลูกค้า ซึ่งค่าในแต่ละกลุ่มสินค้าของลูกค้าแต่ละราย คำนวณมาจากผลคูณของราคาต่อหน่วย (Unit Price) กับปริมาณที่ซื้อ (Quantity) จากนั้นทำการรวมคำสั่งซื้อเดียวกันไว้ในรายการเดียวและขั้นตอนต่อมาได้รวมลูกค้ารายเดียวกันไว้ในรายการเดียว สำหรับการตีความหมายของค่าที่ได้ในแต่ละกลุ่มสินค้าของลูกค้าแต่ละราย สามารถตีความได้ว่า ถ้าลูกค้าใช้จ่ายไปกับสินค้ากลุ่มไหนเยอะ แสดงว่าลูกค้ารายนั้นชื่นชอบสินค้าในกลุ่มสินค้านั้นเป็นพิเศษ จากตัวอย่างข้างต้น ลูกค้ารหัส 12347 ใช้จ่ายไปกับกลุ่มสินค้าที่ 4 มากที่สุด รองลงมาสินค้าในกลุ่มที่ 1 และซื้อน้อยที่สุด คือสินค้าในกลุ่มที่ 3 เป็นต้น

CustomerID	categ_0	categ_1	categ_2	categ_3	categ_4
12347	14.55429509	29.83668117	14.52455516	8.676178669	32.40828992
12348	58.04678284	41.95321716	0	0	0
12350	23.65430622	48.44497608	27.9007177	0	0
12352	53.72520468	12.89212046	3.37033144	14.3010058	15.71133762
12353	44.71910112	13.03370787	19.88764045	22.35955056	0

ภาพประกอบ 28 ตัวอย่างข้อมูลผลลัพธ์ของการวิเคราะห์ข้อความเพื่อสร้างกลุ่มสินค้า

การสร้างคุณลักษณะข้อมูลสำหรับแบบจำลองอาร์เอฟเอ็ม ผู้วิจัยทำการรวมข้อมูลการซื้อขายสินค้าโดยจัดกลุ่มตามแอตทริบิวต์รหัสลูกค้า จากนั้นสร้างตัวแปรใหม่ขึ้น โดยมีรายละเอียดการคำนวณดังนี้

1. จำนวนวันที่ลูกค้าซื้อล่าสุด (Recency) ดังสมการที่ 1

$$R_i = (D_n - D_l)^{-1} \quad (1)$$

R_i หมายถึง ค่ารีเซนซีของลูกค้าที่ i

D_n หมายถึง วันที่กำหนดให้เป็นวันที่ปัจจุบัน

D_l หมายถึง วันที่ลูกค้าซื้อสินค้าล่าสุด

เพื่อให้ค่า Recency มีทิศทางเดียวกับค่า ความถี่และค่าจำนวนเงิน โดยค่ามากหมายถึงการที่ลูกค้ามีศักยภาพมาก เราจึงใช้ส่วนกลับของเวลาครั้งล่าสุด แทนที่จะเป็นค่าเวลาล่าสุดตามปกติ

2. ความบ่อยในการซื้อ (Frequency) ดังสมการที่ 2

$$F_i = \sum_{i=m}^{n} Ii \quad (2)$$

F_i หมายถึง ค่าฟรีเควนซีของลูกค้าที่ i

I หมายถึง จำนวนใบสั่งซื้อ

3. คำนวณจำนวนเงินที่ลูกค้าใช้จ่าย (Monetary) ดังสมการที่ 3

$$M_i = \sum_{i=m}^{n} Ti \quad (3)$$

M_i หมายถึง โมนทารี ผลรวมยอดซื้อสินค้าทุกเครื่องเข้าด้วยกัน

T หมายถึง จำนวนเงินที่ลูกค้าจ่าย

ดังนั้นลูกค้าแต่ละรายจะมีค่า RFM เป็นของตนเอง ซึ่งคุณลักษณะตัวนี้ จะเป็นไปตามลูกค้าแต่ละราย มิใช่คุณลักษณะของแต่ละรายการสั่งซื้อ

3.4 การวิเคราะห์ข้อมูลด้วยเทคนิคการแบ่งกลุ่มข้อมูล

ในการศึกษานี้ ผู้วิจัยเปรียบเทียบแบบจำลอง 2 แบบ โดยแบบจำลองรูปแบบที่ 1 จะเป็นการจัดกลุ่มด้วยเทคนิคอาร์เอฟเอ็ม ส่วนในแบบจำลองที่ 2 จะเป็นการจัดกลุ่มข้อมูลจากใบสั่งซื้อร่วมกับการจัดกลุ่มศักยภาพของลูกค้าด้วยเทคนิคอาร์เอฟเอ็ม โดยแบบจำลองแต่ละแบบมีรายละเอียดดังนี้

แบบจำลองที่ 1 การจัดกลุ่มด้วยการใช้ข้อมูลอาร์เอฟเอ็ม

ผู้วิจัยนำฟีเจอร์ที่ได้จากขั้นตอนการวิเคราะห์ข้อมูลลูกค้าด้วยเทคนิคอาร์เอฟเอ็ม ดังตัวอย่างในภาพที่ 29 จากนั้นทำการจัดกลุ่มของข้อมูลด้วยอัลกอริทึมเคมีน

	Recency	Frequency	Monetary
0	0.007752	3	55.16
1	0.013333	2	250.00
2	0.013889	6	498.80
3	0.004902	1	19.90
4	0.004310	1	20.80
5	0.004673	1	30.00

ภาพประกอบ 29 ตัวอย่างข้อมูลการจัดกลุ่มด้วยเทคนิคอาร์เอฟเอ็ม

แบบจำลองที่ 2 การจัดกลุ่มด้วยการใช้ข้อมูลอาร์เอฟเอ็มและการวิเคราะห์ข้อความ

ผู้วิจัยนำฟีเจอร์ที่ได้จากขั้นตอนการวิเคราะห์ข้อมูลลูกค้าด้วยเทคนิคอาร์เอฟเอ็มร่วมกับการแบ่งกลุ่มสินค้าด้วยเทคนิควิเคราะห์ข้อความ ดังตัวอย่างในภาพที่ 30

	Recency	Frequency	Monetary	categ_0	categ_1	categ_2	categ_3	categ_4
0	0.007752	3	55.16	14.554295	29.836681	14.524555	8.676179	32.408290
1	0.013333	2	250.00	58.046783	41.953217	0.000000	0.000000	0.000000
2	0.013889	6	498.80	53.725205	12.892120	3.370331	14.301006	15.711338
3	0.004902	1	19.90	44.719101	13.033708	19.887640	22.359551	0.000000
4	0.004310	1	20.80	22.061330	31.176580	10.825459	16.235872	19.700760
5	0.004673	1	30.00	9.882455	53.286896	11.558555	0.000000	25.272094

ภาพประกอบ 30 ตัวอย่างข้อมูลการจัดกลุ่มด้วยเทคนิคอาร์เอฟเอ็มร่วมกับเทคนิควิเคราะห์ข้อความ

แบบจำลองที่ 3 การสร้างคะแนนอาร์เอฟเอ็มร่วมกับการจัดกลุ่มข้อมูลตามใบสั่งซื้อ

แบบจำลองที่ 4 การวัดประสิทธิภาพระหว่างแบบจำลองที่ 1 และแบบจำลองที่ 2

แบบจำลองที่ 5 การสร้างแบบจำลองเพื่อเปรียบเทียบอัลกอริทึมแอกโกเมอเรทีพกับเคมีน

ผู้วิจัยได้สร้างแบบจำลองโดยใช้อัลกอริทึมแอกโกเมอเรทีพขึ้นมา เพื่อเปรียบเทียบประสิทธิภาพกับแบบจำลองที่สร้างโดยใช้อัลกอริทึมเคมีน

3.5 การวัดประสิทธิภาพของผลลัพธ์จากการวิเคราะห์ข้อมูล

ผู้วิจัยใช้วิธีวิเคราะห์และประเมินผลลัพธ์เพื่อเปรียบเทียบกลุ่มข้อมูลจากทั้งสองแบบจำลองด้วยค่า Adjusted Rand Index (ARI) และค่า Normalized Mutual Information (NMI) ซึ่งเป็นค่าที่ใช้วัดความคล้ายคลึงกันระหว่างกลุ่มข้อมูล, ทำการเปรียบเทียบความสอดคล้องกันของข้อมูลทั้งสองแบบจำลอง (ระหว่างแบบจำลองที่ 1 กับแบบจำลองที่ 2) ด้วยค่า Confusion Matrix, เปรียบเทียบการคอมแพ็คระหว่างทั้งสองแบบจำลองด้วยค่า Silhouette ผลลัพธ์ที่ได้ดังตารางที่ 2

ตาราง 2 ค่า Silhouette score เพื่อเปรียบเทียบประสิทธิภาพแบบจำลอง

แบบจำลองที่ 1 (RFM)	แบบจำลองที่ 2 (RFM + Text Analytics)
0.3853	0.2222

จะเห็นได้ว่าค่า Silhouette ในแบบจำลองที่ 2 น้อยกว่าของแบบจำลองที่ 1 (คลัสเตอร์จะไม่คอมแพ็กเท่าแบบจำลองที่ 1) แต่สิ่งที่แบบจำลองที่ 2 จะได้มาทดแทน คือ การสะท้อนศักยภาพของลูกค้าและยังสามารถจำแนกลักษณะการสั่งซื้อสินค้าหลัก ในแต่ละกลุ่มได้อีกด้วย ลำดับถัดมาได้เปรียบเทียบแบบจำลองที่ใช้อัลกอริทึมแอกโกเมอเรทีพในการจัดกลุ่มกับแบบจำลองที่ใช้อัลกอริทึมเคมีนในการจัดกลุ่มด้วยค่า Silhouette ผลลัพธ์ที่ได้ดังตารางที่ 3

ตาราง 3 ค่า Silhouette score เพื่อเปรียบเทียบประสิทธิภาพอัลกอริทึม

	แบบจำลองที่ 1 (RFM)	แบบจำลองที่ 2 (RFM + Text Analytics)
K-means	0.3853	0.2222
Agglomerative	0.3180	0.1493

ผลลัพธ์ที่ได้แสดงให้เห็นว่าประสิทธิภาพของแบบจำลองที่สร้างโดยใช้อัลกอริทึมแอกโกเมอเรทีพต่ำกว่าของแบบจำลองที่สร้างโดยใช้เคมีน โดยสามาเหตุที่ทำให้ค่า Silhouette score ต่ำนั้น เนื่องมาจากอัลกอริทึมแอกโกเมอเรทีพจะทำการรวมคลัสเตอร์โดยไม่พยายามเพิ่มระยะทางภายในของคลัสเตอร์ (Intra cluster distance) ในมิติที่สูงขึ้น จึงทำให้คลัสเตอร์ที่ได้จากอัลกอริทึมนี้ไม่คอมแพ็กเท่ากับการจัดกลุ่มด้วยเคมีน จึงเป็นเหตุผลให้ผู้วิจัยเลือกใช้อัลกอริทึมในการวิจัย ซึ่งผลลัพธ์ที่ได้จากการวิจัยในครั้งนี้จะทำให้ทราบพฤติกรรมการซื้อของลูกค้าว่ามีกลุ่มลูกค้าใดบ้างที่ซื้อสินค้าเหมือนกันหรือคล้ายคลึงกัน อีกทั้งยังสะท้อนให้เห็นกลุ่มสินค้าที่ใกล้เคียงกันอีกด้วย เพื่อเป็นประโยชน์ต่อธุรกิจในอนาคตต่อไป

บทที่ 4
ผลการดำเนินงานวิจัย

ในการวิจัยเพื่อศึกษาวิธีการแบ่งกลุ่มลูกค้า ซึ่งใช้ข้อมูลเกี่ยวกับการซื้อขายสินค้าของร้านค้าปลีกในประเทศอังกฤษ โดยใช้เทคนิคการเรียนรู้ของเครื่อง เพื่อจัดกลุ่มของข้อมูลร่วมกับการวิเคราะห์ข้อความ ผู้วิจัยได้ดำเนินการวิจัยโดยการศึกษาตามขบวนการและขั้นตอนต่างๆ ตลอดจนการวัดประสิทธิภาพ เพื่อให้บรรลุจุดประสงค์ของการวิจัยที่ได้กำหนดไว้ ได้ดังนี้

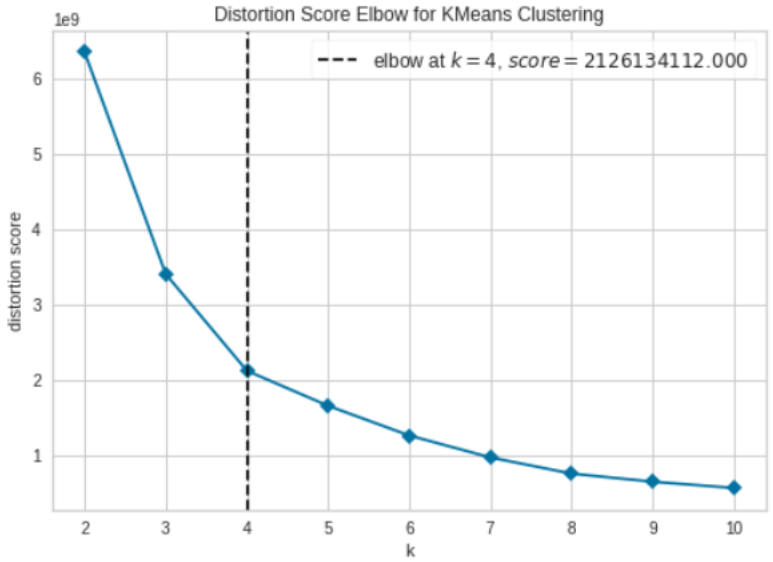
1. ผลลัพธ์ของการจัดกลุ่มด้วยเทคนิคอาร์เอฟเอ็ม
2. ผลลัพธ์ของการจัดกลุ่มด้วยเทคนิคอาร์เอฟเอ็มและการวิเคราะห์ข้อความ
3. ผลลัพธ์จากการเปรียบเทียบระหว่างการจัดกลุ่มด้วยเทคนิคอาร์เอฟเอ็มและการจัดกลุ่มด้วยเทคนิคอาร์เอฟเอ็มและการวิเคราะห์ข้อความ

ผลลัพธ์ของการจัดกลุ่มด้วยเทคนิคอาร์เอฟเอ็ม

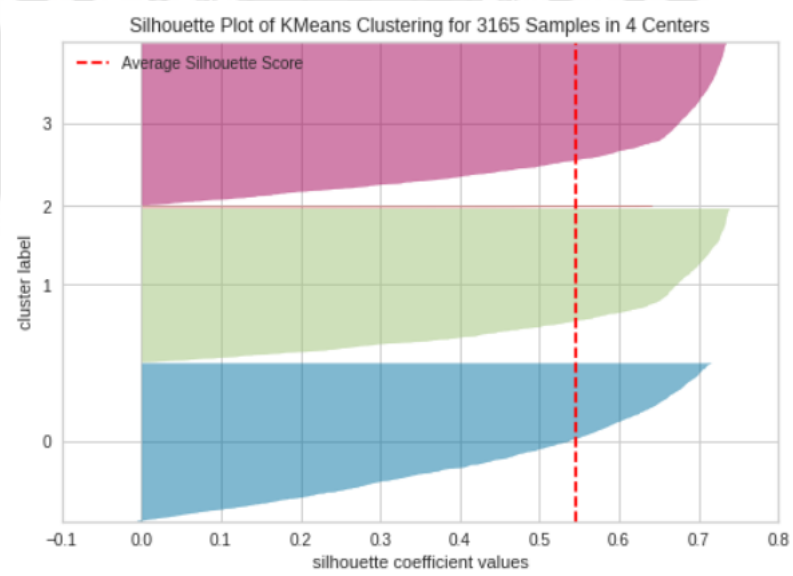
จากการสร้างแบบจำลองการจัดกลุ่มด้วยเทคนิคอาร์เอฟเอ็ม โดยใช้ข้อมูลที่ได้จากขั้นตอนการวิเคราะห์ข้อมูลลูกค้าด้วยเทคนิคอาร์เอฟเอ็ม ดังตัวอย่างในภาพที่ 31 จากนั้นทำการจัดกลุ่มของข้อมูลด้วยอัลกอริทึมเคมีน ซึ่งใช้วิธี Elbow Method ดังภาพที่ 32 และ Silhouette Analysis ดังภาพที่ 33 ในการหาจำนวนคลัสเตอร์ที่เหมาะสม ผลลัพธ์จากการจัดกลุ่มแสดงได้ดังภาพที่ 34

	Recency	Frequency	Monetary
0	0.007752	3	55.16
1	0.013333	2	250.00
2	0.013889	6	498.80
3	0.004902	1	19.90
4	0.004310	1	20.80
5	0.004673	1	30.00

ภาพประกอบ 31 ตัวอย่างข้อมูลการจัดกลุ่มด้วยเทคนิคอาร์เอฟเอ็ม



ภาพประกอบ 32 กราฟ Elbow Method ในการหาค่า Optimal cluster number



ภาพประกอบ 33 กราฟ Silhouette Analysis ในการหาค่า Optimal cluster number

CustomerID	Recency	Frequency	Monetary	Cluster
12347	129	3	55.16	Cluster 0
12348	75	2	250.00	Cluster 0
12352	72	6	498.80	Cluster 0
12353	204	1	19.90	Cluster 0
12354	232	1	20.80	Cluster 0

ภาพประกอบ 34 ตัวอย่างผลลัพธ์การจัดกลุ่มด้วยเทคนิคอาร์เอฟเอ็ม

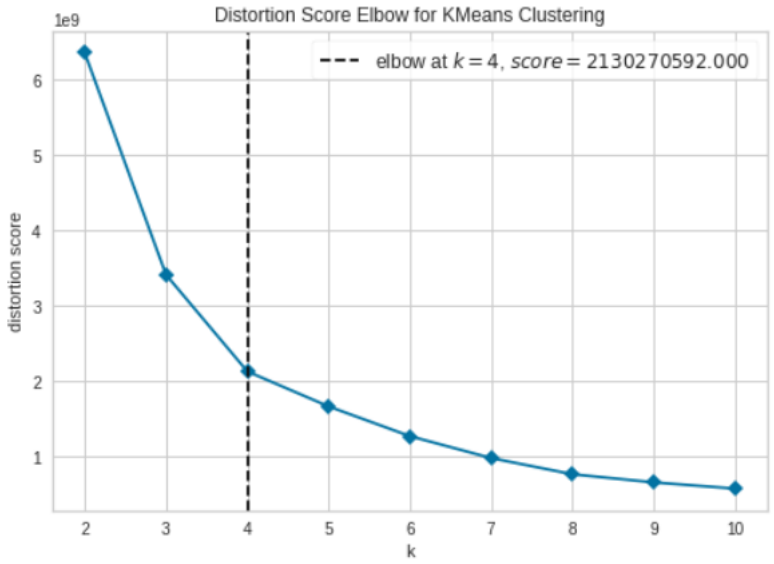
จากผลลัพธ์ของ Elbow Method และ Silhouette Analysis ผู้วิจัยเลือกจัดกลุ่มข้อมูลเป็น 5 คลัสเตอร์ โดยได้ผลลัพธ์ดังนี้ คลัสเตอร์ที่ 1 มีจำนวน 616 เรคคอร์ด คลัสเตอร์ที่ 2 มีจำนวน 408 เรคคอร์ด คลัสเตอร์ที่ 3 มีจำนวน 389 เรคคอร์ด คลัสเตอร์ที่ 4 มีจำนวน 1,283 เรคคอร์ด และคลัสเตอร์ที่ 5 มีจำนวน 459 เรคคอร์ด

ผลลัพธ์ของการจัดกลุ่มด้วยเทคนิคอาร์เอฟเอ็มและการวิเคราะห์ข้อความ

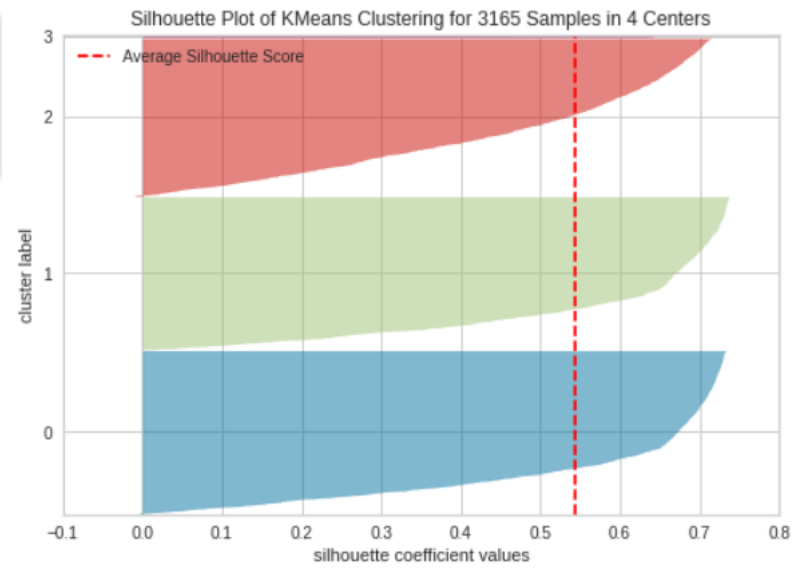
จากการสร้างแบบจำลองการจัดกลุ่มด้วยเทคนิคอาร์เอฟเอ็มและการวิเคราะห์ข้อความ โดยใช้ข้อมูลที่ได้จากขั้นตอนการวิเคราะห์ข้อมูลลูกค้าด้วยเทคนิคอาร์เอฟเอ็ม ดังตัวอย่างในภาพที่ 35 จากนั้นทำการจัดกลุ่มของข้อมูลด้วยอัลกอริทึมเคมีน ซึ่งใช้วิธี Elbow Method ดังภาพที่ 36 และ Silhouette Analysis ดังภาพที่ 37 ในการหาจำนวนคลัสเตอร์ที่เหมาะสม ผลลัพธ์จากการจัดกลุ่มแสดงได้ดังภาพที่ 38

	Recency	Frequency	Monetary	categ_0	categ_1	categ_2	categ_3	categ_4
0	0.007752	3	55.16	14.554295	29.836681	14.524555	8.676179	32.408290
1	0.013333	2	250.00	58.046783	41.953217	0.000000	0.000000	0.000000
2	0.013889	6	498.80	53.725205	12.892120	3.370331	14.301006	15.711338
3	0.004902	1	19.90	44.719101	13.033708	19.887640	22.359551	0.000000
4	0.004310	1	20.80	22.061330	31.176580	10.825459	16.235872	19.700760
5	0.004673	1	30.00	9.882455	53.286896	11.558555	0.000000	25.272094

ภาพประกอบ 35 ตัวอย่างข้อมูลการจัดกลุ่มด้วยเทคนิคอาร์เอฟเอ็มและการวิเคราะห์ข้อความ



ภาพประกอบ 36 กราฟ Elbow Method ในการหาค่า Optimal cluster number



ภาพประกอบ 37กราฟ Silhouette Analysis ในการหาค่า Optimal cluster number

CustomerID	Recency	Frequency	Monetary	categ_0	categ_1	categ_2	categ_3	categ_4	Cluster
12347	129	3	55.16	14.554295	29.836681	14.524555	8.676179	32.408290	Cluster 0
12348	75	2	250.00	58.046783	41.953217	0.000000	0.000000	0.000000	Cluster 0
12352	72	6	498.80	53.725205	12.892120	3.370331	14.301006	15.711338	Cluster 0
12353	204	1	19.90	44.719101	13.033708	19.887640	22.359551	0.000000	Cluster 0
12354	232	1	20.80	22.061330	31.176580	10.825459	16.235872	19.700760	Cluster 0

ภาพประกอบ 38 ตัวอย่างผลลัพธ์การจัดกลุ่มด้วยเทคนิคอาร์เอฟเอ็มและการวิเคราะห์ข้อความ

จากผลลัพธ์ของ Elbow Method และ Silhouette Analysis เช่นเดียวกับวิธีที่ 1 เราเลือกจัดกลุ่มข้อมูลเป็น 5 คลัสเตอร์ ได้ผลลัพธ์ดังนี้ คลัสเตอร์ที่ 1 มีจำนวน 813 เรคคอร์ด คลัสเตอร์ที่ 2 มีจำนวน 1,298 เรคคอร์ด คลัสเตอร์ที่ 3 มีจำนวน 627 เรคคอร์ด คลัสเตอร์ที่ 4 มีจำนวน 219 เรคคอร์ด และคลัสเตอร์ที่ 5 มีจำนวน 198 เรคคอร์ด

ผลลัพธ์จากการเปรียบเทียบระหว่างการจัดกลุ่มด้วยเทคนิคอาร์เอฟเอ็มและการจัดกลุ่มด้วยเทคนิคอาร์เอฟเอ็มและการวิเคราะห์ข้อความ

ผลลัพธ์ที่ได้จากการทดลองของแบบจำลองการจัดกลุ่มด้วยเทคนิคอาร์เอฟเอ็มและการจัดกลุ่มด้วยเทคนิคอาร์เอฟเอ็มและการวิเคราะห์ข้อความ จะเห็นได้ว่าการแบ่งกลุ่มด้วยเทคนิคอาร์เอฟเอ็มร่วมกับการวิเคราะห์ข้อความมีประสิทธิภาพดีกว่าวิธีที่ใช้ฟีเจอร์ที่ได้จากเทคนิคอาร์เอฟเอ็ม เพียงเทคนิคเดียว เพราะสามารถสะท้อนให้เห็นกลุ่มลูกค้าจากการซื้อขายสินค้า อีกทั้งยังสะท้อนให้เห็นกลุ่มสินค้าที่ใกล้เคียงกันอีกด้วย ประโยชน์ของงานวิจัยนี้ เมื่อธุรกิจเข้าใจพฤติกรรมต่างๆของลูกค้า ว่าลูกค้าแต่ละกลุ่ม ชื่นชอบสินค้าประเภทไหนเป็นพิเศษ ทีมจัดซื้อก็สามารถวางแผนสต็อกสินค้าให้พอดีกับความต้องการขายได้ อีกทั้งทีมการตลาดก็สามารถนำไปวางแผนการตลาดเชิงกลยุทธ์ได้อีกด้วย

บทที่ 5
สรุปผลการวิจัย อภิปรายผล และข้อเสนอแนะ

ในการวิจัยเพื่อศึกษาวิธีการแบ่งกลุ่มลูกค้า ซึ่งใช้ข้อมูลเกี่ยวกับการซื้อขายสินค้าของร้านค้าปลีกในประเทศอังกฤษ โดยใช้เทคนิคการเรียนรู้ของเครื่อง เพื่อจัดกลุ่มของข้อมูลร่วมกับการวิเคราะห์ข้อความ ผู้วิจัยได้ประเมินประสิทธิภาพของแบบจำลองแต่ละเทคนิค เพื่อนำมาเปรียบเทียบและสรุปผล โดยสามารถแบ่งหัวข้อในการสรุปผลได้ดังต่อไปนี้

1. สรุปผลการวิจัย
2. อภิปรายผลการวิจัย
3. ข้อเสนอแนะ

สรุปผลการวิจัย

ในการวิจัยนี้เป็นการศึกษาวิธีการแบ่งกลุ่มลูกค้าด้วยเทคนิคอาร์เอฟเอ็มร่วมกับเทคนิควิเคราะห์ข้อความ โดยศึกษา 2 วิธีดังนี้

วิธีที่ 1 การจัดกลุ่มด้วยเทคนิคอาร์เอฟเอ็ม

โดยใช้ฟีเจอร์ที่ได้จากขั้นตอนการวิเคราะห์ข้อมูลลูกค้าด้วยเทคนิคอาร์เอฟเอ็ม

วิธีที่ 2 การจัดกลุ่มด้วยเทคนิคอาร์เอฟเอ็มและการวิเคราะห์ข้อความ

ใช้ฟีเจอร์ที่ได้จากขั้นตอนการวิเคราะห์ข้อมูลลูกค้าด้วยเทคนิคอาร์เอฟเอ็มร่วมกับการแบ่งกลุ่มสินค้าด้วยเทคนิควิเคราะห์ข้อความ ผลการทดลองพบว่าวิธีการจัดกลุ่มด้วยเทคนิคอาร์เอฟเอ็มและการวิเคราะห์ข้อความมีประสิทธิภาพที่ดีกว่า เพราะสามารถสะท้อนให้เห็นกลุ่มลูกค้าจากการซื้อขายสินค้า อีกทั้งยังสะท้อนให้เห็นกลุ่มสินค้าที่ใกล้เคียงกันอีกด้วย

อภิปรายผลการวิจัย

การทดลองนี้ใช้เว็บแอพพลิเคชั่นจูปิเตอร์โน๊ตบุ๊ค สำหรับสร้างตัวแบบแบ่งกลุ่มลูกค้า มีจำนวนทั้งสิ้น 3,155 เรคคอร์ด ประกอบด้วย 2 วิธีดังนี้

วิธีที่ 1 การจัดกลุ่มด้วยเทคนิคอาร์เอฟเอ็ม ประกอบด้วยคุณลักษณะ ดังนี้

1) จำนวนวันที่ลูกค้าซื้อล่าสุด (Recency)

2) ความบ่อยในการซื้อ (Frequency)

3) จำนวนเงินที่ลูกค้าใช้จ่าย (Monetary)

จากการจัดกลุ่มของข้อมูลด้วยเทคนิคอาร์เอฟเอ็มจะเป็นการแบ่งกลุ่มของลูกค้าออกเป็น 5 คลัสเตอร์ โดยได้ผลลัพธ์ดังนี้ คลัสเตอร์ที่ 1 มีจำนวน 616 เรคคอร์ด คลัสเตอร์ที่ 2 มีจำนวน 408 เรคคอร์ด คลัสเตอร์ที่ 3 มีจำนวน 389 เรคคอร์ด คลัสเตอร์ที่ 4 มีจำนวน 1,283 เรคคอร์ด และคลัสเตอร์ที่ 5 มีจำนวน 459 เรคคอร์ด

วิธีที่ 2 การจัดกลุ่มด้วยเทคนิคอาร์เอฟเอ็มและการวิเคราะห์ข้อความประกอบด้วยคุณลักษณะดังนี้

1) จำนวนวันที่ลูกค้าซื้อล่าสุด (Recency)
2) ความบ่อยในการซื้อ (Frequency)
3) จำนวนเงินที่ลูกค้าใช้จ่าย (Monetary)
4) กลุ่มสินค้าที่ 1 (categ_0)
5) กลุ่มสินค้าที่ 2 (categ_1)
6) กลุ่มสินค้าที่ 3 (categ_2)
7) กลุ่มสินค้าที่ 4 (categ_3)

เช่นเดียวกับวิธีที่ 1 เราเลือกจัดกลุ่มข้อมูลเป็น 5 คลัสเตอร์ ได้ผลลัพธ์ดังนี้ คลัสเตอร์ที่ 1 มีจำนวน 813 เรคคอร์ด คลัสเตอร์ที่ 2 มีจำนวน 1,298 เรคคอร์ด คลัสเตอร์ที่ 3 มีจำนวน 627 เรคคอร์ด คลัสเตอร์ที่ 4 มีจำนวน 219 เรคคอร์ด และคลัสเตอร์ที่ 5 มีจำนวน 198 เรคคอร์ด

การเปรียบเทียบผลของการจัดกลุ่มทั้งสองแบบจำลอง

ในการเปรียบเทียบค่า Adjusted Rand Index และค่า Normalized Mutual Information ระหว่างแบบจำลองที่ 1 กับ 2 เราได้ค่าดังตารางที่ 2

ตาราง 4 ค่า Adjusted Rand index และค่า Normalized Mutual Information

ตัววัด	ค่า
Adjusted Rand index	0.5116
Normalized Mutual Information	0.3646

โดยมีค่า Confusion Matrix ดังนี้ โดย M1 M2 หมายถึงแบบจำลองที่ 1 และ 2 ตามลำดับ

ตาราง 5 ค่า Confusion Matrix

M1 / M2	0	1	2	3	4
0	159	217	210	3	224
1	100	0	0	1198	0
2	230	114	114	14	155
3	71	40	29	44	35
4	56	37	36	24	45

จากตารางที่ 2 และ 3 จะเห็นได้ว่า ผลที่ได้จากการแบ่งกลุ่ม ไม่สอดคล้องกันมากนัก ยกเว้น ในกลุ่ม (1, 3) ซึ่งสมาชิกในกลุ่มทั้งสองนี้ จะแทบสอดคล้องกันทั้งสิ้น ในส่วนกลุ่มอื่นๆ จะค่อนข้างกระจัดกระจายไปในคลัสเตอร์ต่างๆ เมื่อเราไปพิจารณาดูค่า จุดศูนย์กลางของแต่ละคลัสเตอร์ในภาพที่ 39 และ 40 ดังนี้

	Recency	Frequency	Monetary
0	0.006	2.698	114.283
1	0.010	1.515	154.736
2	0.013	1.103	49.540
3	0.004	4.788	281.780
4	0.008	1.987	92.061

ภาพประกอบ 39 จุดศูนย์กลางของแต่ละคลัสเตอร์ในแบบจำลองที่ 1

	Recency	Frequency	Monetary	categ_0	categ_1	categ_2	categ_3	categ_4
0	0.006	1.849	70.915	17.465	16.779	17.687	27.620	20.462
1	0.012	4.791	261.526	19.661	25.560	21.314	14.823	18.650
2	0.006	1.926	73.398	20.257	44.847	16.213	6.865	11.828
3	0.007	2.174	168.925	9.227	14.893	61.239	6.386	8.255
4	0.007	1.856	384.601	69.664	11.324	7.875	5.493	5.643

ภาพประกอบ 40 จุดศูนย์กลางของแต่ละคลัสเตอร์ในแบบจำลองที่ 2

จากผลของจุดศูนย์กลางที่ได้ จะเห็นว่า ค่า Recency ไม่มีบทบาทในการจัดกลุ่มมากนัก เนื่องจากค่า Recency ของจุดศูนย์กลางของทุกกลุ่มจะมีค่าใกล้เคียงกัน ในขณะที่ ฟีเจอร์ที่มีค่าแตกต่างกันอย่างเด่นชัดจะเป็นค่า Monetary ที่ในแต่ละกลุ่มของทั้งสองแบบจำลองมีค่าแยกกันอย่างชัดเจน โดยจะเห็นว่า จุดศูนย์กลางกลุ่ม 3 ในแบบจำลองที่ 1 จะมีค่าใกล้เคียงกับ จุดศูนย์กลางกลุ่ม 1 ในแบบจำลองที่ 2 ซึ่งสอดคล้องกับผลที่ได้จากการหา Confusion Matrix ในตารางที่ 3

ทั้งนี้ จะเห็นได้ว่า ในการจัดกลุ่มด้วยการใช้แบบจำลองที่ 2 นอกจากจะสะท้อนศักยภาพของลูกค้าแล้ว ยังสามารถจำแนกลักษณะการสั่งซื้อสินค้าหลัก ในแต่ละกลุ่มได้อีกด้วย เช่น ในกลุ่มที่ 0 จะเป็นพวกที่มีลักษณะ คือ สั่งสินค้าทั้ง 5 กลุ่มในปริมาณที่เท่าๆกัน แต่จะมียอดสั่งซื้อต่ำ และมียอดการใช้จ่ายต่ำ ในขณะที่ในกลุ่มที่มียอดการสั่งซื้อสูง ในกลุ่มที่ 4 จะนิยมสั่งซื้อสินค้าใน category_0 มากที่สุด ซึ่งจากข้อมูล Insight ในส่วนนี้ ทำให้เราสามารถวางแผนการสั่งสินค้าให้สอดคล้องกับความต้องการของลูกค้าที่มียอดการสั่งซื้อสูงได้

ข้อเสนอแนะ

1. การวิจัยนี้ในขั้นตอนการเก็บรวบรวมข้อมูล ตรวจสอบข้อมูล และพิจารณาข้อมูล เพื่อนำมาใช้ในการวิเคราะห์ ผู้วิจัยได้ทำการลบข้อมูลแถวที่ประกอบไปด้วย ข้อมูลที่มีการแก้ไข ข้อมูลที่ถูกยกเลิก และข้อมูลที่สูญหายทิ้งทั้งแถว ทำให้ข้อมูลในบางคอลัมน์ที่อยู่ภายในแถว ซึ่งเป็นข้อมูลที่ดีอาจถูกลบไปด้วย ดังนั้นหากมีวิธีที่ไม่ต้องลบข้อมูลทั้งแถว อาจช่วยเพิ่มประสิทธิภาพในการแบ่งกลุ่มให้ดียิ่งขึ้น

2. เนื่องจากการวิจัยนี้ได้ใช้อัลกอริทึมเคมีนในการแบ่งกลุ่ม ดังนั้นอาจมีอัลกอริทึมอื่นที่มีความสามารถในการแบ่งกลุ่มได้อย่างมีประสิทธิภาพมากกว่า

บรรณานุกรม

1. Chen D, Sain SL, Guo K. Data mining for the online retail industry: A case study of RFM model-based customer segmentation using data mining. Journal of Database Marketing & Customer Strategy Management. 2012;19(3):197-208.

2. Hu X, Zhang H, Wu X, Chen J, Xiao Y, Xue Y, et al., editors. A Novel Approach for Customer Segmentation Based on Biclustering. International Conference on Web Information Systems Engineering; 2013: Springer.

3. Wang B, Miao Y, Zhao H, Jin J, Chen Y. A biclustering-based method for market segmentation using customer pain points. Engineering Applications of Artificial Intelligence. 2016;47:101-9.

4. Maryani I, Riana D, Astuti RD, Ishaq A, Pratama EA, editors. Customer Segmentation based on RFM model and Clustering Techniques With K-Means Algorithm. 2018 Third International Conference on Informatics and Computing (ICIC); 2018: IEEE.

5. Kansal T, Bahuguna S, Singh V, Choudhury T, editors. Customer segmentation using k-means clustering. 2018 International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS); 2018: IEEE.

ภาคผนวก

รายละเอียดโค้ดการตรวจสอบข้อมูลและพิจารณาข้อมูลมาใช้ในการวิเคราะห์จากข้อมูลการซื้อขายสินค้า

```
# Commented out IPython magic to ensure Python compatibility.
import pandas as pd
import numpy as np

import time, warnings
import datetime as dt

#Visualizations
import matplotlib.pyplot as plt
from pandas.plotting import scatter_matrix
# %matplotlib inline
import seaborn as sns

warnings.filterwarnings("ignore")

"""# Get the Data"""

#load the dataset
retail_df = pd.read_excel("Online Retail.xlsx") #2010 - 2011

retail_df.shape

retail_df.head()

retail_df.describe()

"""# Prepare the Data"""

#Calculate the total sales of each product
TotalAmount = retail_df['Quantity'] * retail_df['UnitPrice']
retail df.insert(loc=6.column='TotalAmount'.value=TotalAmount)


#create a new column called weekday which contains weekday of invoice only
retail_df['day_of_week'] = retail_df['InvoiceDate'].dt.day_name()

#create a new column called date which contains the date of invoice only
retail_df['date'] = retail_df['InvoiceDate'].dt.date

#create a new column called time which contains the time of invoice only
retail_df['time'] = retail_df['InvoiceDate'].dt.time

retail_df.head()

pip install dataprep

#Analyze distributions with plot()

from dataprep.eda import plot
plot(retail_df)

#Analyze correlations with plot_correlation()

from dataprep.eda import plot_correlation
plot_correlation(retail_df)

plot(retail_df,"UnitPrice")

plot(retail_df,"Quantity")

plot(retail_df,"CustomerID")

plot(retail_df, "UnitPrice", "Quantity")


#Analyze missing values with plot_missing()

from dataprep.eda import plot_missing
plot_missing(retail_df)

#Remove rows with missing values
retail_df = retail_df.dropna()

# check missing values for each column
retail_df.isnull().sum().sort_values(ascending=False)

# change columns CustomerID - String to Int type
retail_df['CustomerID'] = retail_df['CustomerID'].astype('int64')

retail_df.describe().round(2)

#check the shape
retail_df.shape

#remove canceled orders or Quantity with negative values
retail_df = retail_df[retail_df['Quantity']>0]
retail_df.shape

#remove rows where customerID are NA
retail_df.dropna(subset=['CustomerID'],how='all',inplace=True)
retail_df.shape

#restrict the data to one full year
retail_df = retail_df[retail_df['InvoiceDate']>= "2010-12-09"]
retail_df.shape
```

```
94
95  print("Summary..")
96  #exploring the unique values of each attribute
97  print("Number of transactions: ", retail_df['InvoiceNo'].nunique())
98  print("Number of products bought: ",retail_df['StockCode'].nunique())
99  print("Number of customers:", retail_df['CustomerID'].nunique() )
100 print("Percentage of customers NA: ", round(retail_df['CustomerID'].isnull().sum() * 100 / len(retail_df),2),"%" )
101
102 retail_df.head()
103
104 plot(retail_df)
105
106 plot(retail_df, "Quantity")
107
108 plot(retail_df, "TotalAmount","Description")
109
110 sum_retail_df = retail_df.groupby(['Description']).sum().round(2)
111
112 retail_df.to_csv('clean_online_retail.csv')
113
114 sum_retail_df.to_csv('sum_online_retail.csv')
115
116 #remove canceled orders or Quantity with negative values
117 retail_df = retail_df[retail_df['Quantity']>0]
118 retail_df.shape
119
120 plot(retail_df, "TotalAmount", "Country")
121
122 plot(retail_df, "day_of_week")
123
124 plot(retail_df, "date")
125
126 plot(retail_df, "time")
127
128 # top 10 customers, max amounts paid
129 res_df = sum_retail_df.groupby(['CustomerID']).sum()
130 res_df.sort_values('TotalAmount',ascending=False,inplace=True)
131 final_df = res_df[(res_df['Quantity'] > 0) & (res_df['TotalAmount'] > 0)]
132 final_df.head(10)
133
134 # total sales of product at each month at each country I made filter on total sales >= 100
135 retail_df = retail_df[ (retail_df['Quantity'] > 0) & (retail_df['TotalAmount'] >0) ]
136 retail_df = retail_df[retail_df['TotalAmount'] >= 100 ]
137
138 retail_df.groupby(['Country','date','StockCode','Description']).sum()
139
140 totalsales_retail_df = retail_df.groupby(['Country','date','StockCode','Description']).sum()
141
142 totalsales_retail_df.to_csv('totalsales_online_retail.csv')
```

รายละเอียดโค้ดการสร้างฟีเจอร์จากเทคนิคอาร์เอฟเอ็ม

```
# RFM Analysis

#Last date available in our dataset
retail['InvoiceDate'].max()
now = dt.date(2011,12,9)
print(now)

#create a new column called date which contains the date of invoice only
retail['date'] = pd.DatetimeIndex(retail['InvoiceDate']).date

retail.head()

#group by customers and check last date of purshace
recency_df = retail.groupby(by='CustomerID', as_index=False)['date'].max()
recency_df.columns = ['CustomerID','LastPurshaceDate']
recency_df.head()

#calculate recency
recency_df['Recency'] = recency_df['LastPurshaceDate'].apply(lambda x: (now - x).days)

recency_df.head()

#drop LastPurchaseDate as we don't need it anymore
recency_df.drop('LastPurshaceDate',axis=1,inplace=True)

## Frequency

# drop duplicates
retail_copy = retail
retail_copy.drop_duplicates(subset=['InvoiceNo', 'CustomerID'], keep="first", inplace=True)

#calculate frequency of purchases
frequency_df = retail_copy.groupby(by=['CustomerID'], as_index=False)['InvoiceNo'].count()
frequency_df.columns = ['CustomerID','Frequency']
frequency_df.head()

## Monetary

#create column total cost
retail['TotalCost'] = retail['Quantity'] * retail['UnitPrice']

monetary_df = retail.groupby(by='CustomerID',as_index=False).agg({'TotalCost': 'sum'})
monetary_df.columns = ['CustomerID','Monetary']
monetary_df.head()

## Create RFM Table

#merge recency dataframe with frequency dataframe
temp_df = recency_df.merge(frequency_df,on='CustomerID')
temp_df.head()

#merge with monetary dataframe to get a table with the 3 columns
rfm_df = temp_df.merge(monetary_df,on='CustomerID')
#use CustomerID as index
rfm_df.set_index('CustomerID',inplace=True)
#check the head
rfm_df.head()

rfm_df.to_csv('rfm_df.csv',index=True)
```

รายละเอียดโค้ดการสร้างฟีเจอร์จากการวิเคราะห์ข้อความ

```
"""
____
## Insight on product categories
___
### Products Description

"""

is_noun = lambda pos: pos[:2] == 'NN'

def keywords_inventory(dataframe, colonne = 'Description'):
    stemmer = nltk.stem.SnowballStemmer("english")
    keywords_roots  = dict()  # collect the words / root
    keywords_select = dict()  # association: root <-> keyword
    category_keys   = []
    count_keywords  = dict()
    icount = 0
    for s in dataframe[colonne]:
        if pd.isnull(s): continue
        lines = s.lower()
        tokenized = nltk.word_tokenize(lines)
        nouns = [word for (word, pos) in nltk.pos_tag(tokenized) if is_noun(pos)]

        for t in nouns:
            t = t.lower() ; racine = stemmer.stem(t)
            if racine in keywords_roots:
                keywords_roots[racine].add(t)
                count_keywords[racine] += 1
            else:
                keywords_roots[racine] = {t}
                count_keywords[racine] = 1


    for s in keywords_roots.keys():
        if len(keywords_roots[s]) > 1:
            min_length = 1000
            for k in keywords_roots[s]:
                if len(k) < min_length:
                    clef = k ; min_length = len(k)
            category_keys.append(clef)
            keywords_select[s] = clef
        else:
            category_keys.append(list(keywords_roots[s])[0])
            keywords_select[s] = list(keywords_roots[s])[0]

    print("Nb of keywords in variable '{}': {}".format(colonne,len(category_keys)))
    return category_keys, keywords_roots, keywords_select, count_keywords

"""
"""

df_produits = pd.DataFrame(df_initial['Description'].unique()).rename(columns = {0:'Description'})



keywords, keywords_roots, keywords_select, count_keywords = keywords_inventory(df_produits)

"""
"""

list_products = []
for k,v in count_keywords.items():
    list_products.append([keywords_select[k],v])
list_products.sort(key = lambda x:x[1], reverse = True)

"""""""

liste = sorted(list_products, key = lambda x:x[1], reverse = True)
#_____________________________
plt.rc('font', weight='normal')
fig, ax = plt.subplots(figsize=(7, 25))
y_axis = [i[1] for i in liste[:125]]
x_axis = [k for k,i in enumerate(liste[:125])]
x_label = [i[0] for i in liste[:125]]
plt.xticks(fontsize = 15)
plt.yticks(fontsize = 13)
plt.yticks(x_axis, x_label)
plt.xlabel("Nb. of occurences", fontsize = 18, labelpad = 10)
ax.barh(x_axis, y_axis, align = 'center')
ax = plt.gca()
ax.invert_yaxis()
#_____________________________________________________________________________
plt.title("Words occurence",bbox={'facecolor':'k', 'pad':5}, color='w',fontsize = 25)
plt.show()

"""
### Defining product categories

"""

list_products = []
for k,v in count_keywords.items():
    word = keywords_select[k]
    if word in ['pink', 'blue', 'tag', 'green', 'orange']: continue
```

```
    if len(word) < 3 or v < 13: continue
    if ('+' in word) or ('/' in word): continue
    list_products.append([word, v])
#_____________________________________________________
list_products.sort(key = lambda x:x[1], reverse = True)
print('mots conservés:', len(list_products))

"""____
#### Data encoding
"""

liste_produits = df_cleaned['Description'].unique()
X = pd.DataFrame()
for key, occurence in list_products:
    X.loc[:, key] = list(map(lambda x:int(key.upper() in x), liste_produits))

"""
"""

threshold = [0, 1, 2, 3, 5, 10]
label_col = []
for i in range(len(threshold)):
    if i == len(threshold)-1:
        col = '.>{}'.format(threshold[i])
    else:
        col = '{}<.<{}'.format(threshold[i],threshold[i+1])
    label_col.append(col)
    X.loc[:, col] = 0

for i, prod in enumerate(liste_produits):
    prix = df_cleaned[ df_cleaned['Description'] == prod]['UnitPrice'].mean()
    j = 0
    while prix > threshold[j]:
        j+=1
        if j == len(threshold): break
    X.loc[i, label_col[j-1]] = 1

"""""""

print("{:<8} {:<20} \n".format('gamme', 'nb. produits') + 20*'-')
for i in range(len(threshold)):
    if i == len(threshold)-1:
        col = '.>{}'.format(threshold[i])
    else:
        col = '{}<.<{}'.format(threshold[i],threshold[i+1])
    print("{:<10}  {:<20}".format(col, X.loc[:, col].sum()))

"""
#### Creating clusters of products
"""

matrix = X.as_matrix()
for n_clusters in range(3,10):
    kmeans = KMeans(init='k-means++', n_clusters = n_clusters, n_init=30)
    kmeans.fit(matrix)
    clusters = kmeans.predict(matrix)
    silhouette_avg = silhouette_score(matrix, clusters)
    print("For n_clusters =", n_clusters, "The average silhouette_score is :", silhouette_avg)

"""""""

n_clusters = 5
silhouette_avg = -1
while silhouette_avg < 0.145:
    kmeans = KMeans(init='k-means++', n_clusters = n_clusters, n_init=30)
    kmeans.fit(matrix)
    clusters = kmeans.predict(matrix)
    silhouette_avg = silhouette_score(matrix, clusters)

    #km = kmodes.KModes(n_clusters = n_clusters, init='Huang', n_init=2, verbose=0)
    #clusters = km.fit_predict(matrix)
    #silhouette_avg = silhouette_score(matrix, clusters)
    print("For n_clusters =", n_clusters, "The average silhouette_score is :", silhouette_avg)

"""
#### Characterizing the content of clusters
"""

pd.Series(clusters).value_counts()

"""
** a / _Silhouette intra-cluster score_ **
"""

def graph_component_silhouette(n_clusters, lim_x, mat_size, sample_silhouette_values, clusters):
    plt.rcParams["patch.force_edgecolor"] = True
    plt.style.use('fivethirtyeight')
    mpl.rc('patch', edgecolor = 'dimgray', linewidth=1)
    #_____________________
    fig, ax1 = plt.subplots(1, 1)
    fig.set_size_inches(8, 8)
    ax1.set_xlim([lim_x[0], lim_x[1]])
    ax1.set_ylim([0, mat_size + (n_clusters + 1) * 10])
```

```
    y_lower = 10
    for i in range(n_clusters):
        #_____________________________________________________________________________
        # Aggregate the silhouette scores for samples belonging to cluster i, and sort them
        ith_cluster_silhouette_values = sample_silhouette_values[clusters == i]
        ith_cluster_silhouette_values.sort()
        size_cluster_i = ith_cluster_silhouette_values.shape[0]
        y_upper = y_lower + size_cluster_i
        cmap = cm.get_cmap("Spectral")
        color = cmap(float(i) / n_clusters)
        ax1.fill_betweenx(np.arange(y_lower, y_upper), 0, ith_cluster_silhouette_values,
                          facecolor=color, edgecolor=color, alpha=0.8)
        #_____________________________________________________
        # Label the silhouette plots with their cluster numbers at the middle
        ax1.text(-0.03, y_lower + 0.5 * size_cluster_i, str(i), color = 'red', fontweight = 'bold',
                bbox=dict(facecolor='white', edgecolor='black', boxstyle='round, pad=0.3'))
        #_____________________________________________________
        # Compute the new y_lower for next plot
        y_lower = y_upper + 10

#_____________________________________
# define individual silouhette scores
sample_silhouette_values = silhouette_samples(matrix, clusters)
#_____________
# and do the graph
graph_component_silhouette(n_clusters, [-0.07, 0.33], len(X), sample_silhouette_values, clusters)

"""** b/ _Word Cloud_**
"""

liste = pd.DataFrame(liste_produits)
liste_words = [word for (word, occurence) in list_products]

occurence = [dict() for _ in range(n_clusters)]

for i in range(n_clusters):
    liste_cluster = liste.loc[clusters == i]
    for word in liste_words:
        if word in ['art', 'set', 'heart', 'pink', 'blue', 'tag']: continue
        occurence[i][word] = sum(liste_cluster.loc[:, 0].str.contains(word.upper()))

"""""""

#_____________________________________________________________________
def random_color_func(word=None, font_size=None, position=None,
                      orientation=None, font_path=None, random_state=None):
    h = int(360.0 * tone / 255.0)
    s = int(100.0 * 255.0 / 255.0)
    l = int(100.0 * float(random_state.randint(70, 120)) / 255.0)
    return "hsl({}, {}%, {}%)".format(h, s, l)
#_____________________________________________________________________
def make_wordcloud(liste, increment):
    ax1 = fig.add_subplot(4,2,increment)
    words = dict()
    trunc_occurences = liste[0:150]
    for s in trunc_occurences:
        words[s[0]] = s[1]
    #________________________________________________________
    wordcloud = WordCloud(width=1000,height=400, background_color='white',
                          max_words=1628,relative_scaling=1,
                          color_func = random_color_func,
                          normalize_plurals=False)
    wordcloud.generate_from_frequencies(words)
    ax1.imshow(wordcloud, interpolation="bilinear")
    ax1.axis('off')
    plt.title('cluster {}'.format(increment-1))
#_____________________________________________________________________
fig = plt.figure(1, figsize=(14,14))
color = [0, 160, 130, 95, 280, 40, 330, 110, 25]
for i in range(n_clusters):
    list_cluster_occurences = occurence[i]

    tone = color[i] # define the color of the words
    liste = []
    for key, value in list_cluster_occurences.items():
        liste.append([key, value])
    liste.sort(key = lambda x:x[1], reverse = True)
    make_wordcloud(liste, i+1)

"""
** Principal Component Analysis_ **
"""

pca = PCA()
pca.fit(matrix)
pca_samples = pca.transform(matrix)

"""""""

fig, ax = plt.subplots(figsize=(14, 5))
sns.set(font_scale=1)
plt.step(range(matrix.shape[1]), pca.explained_variance_ratio_.cumsum(), where='mid',
         label='cumulative explained variance')
```

```
sns.barplot(np.arange(1,matrix.shape[1]+1), pca.explained_variance_ratio_, alpha=0.5, color = 'g',
            label='individual explained variance')
plt.xlim(0, 100)

ax.set_xticklabels([s if int(s.get_text())%2 == 0 else '' for s in ax.get_xticklabels()])

plt.ylabel('Explained variance', fontsize = 14)
plt.xlabel('Principal components', fontsize = 14)
plt.legend(loc='upper left', fontsize = 13);

"""""""

pca = PCA(n_components=50)
matrix_9D = pca.fit_transform(matrix)
mat = pd.DataFrame(matrix_9D)
mat['cluster'] = pd.Series(clusters)

import matplotlib.patches as mpatches

sns.set_style("white")
sns.set_context("notebook", font_scale=1, rc={"lines.linewidth": 2.5})

LABEL_COLOR_MAP = {0:'r', 1:'gold', 2:'b', 3:'k', 4:'c', 5:'g'}
label_color = [LABEL_COLOR_MAP[l] for l in mat['cluster']]

fig = plt.figure(figsize = (15,8))
increment = 0
for ix in range(4):
    for iy in range(ix+1, 4):
        increment += 1
        ax = fig.add_subplot(2,3,increment)
        ax.scatter(mat[ix], mat[iy], c= label_color, alpha=0.4)
        plt.ylabel('PCA {}'.format(iy+1), fontsize = 12)
        plt.xlabel('PCA {}'.format(ix+1), fontsize = 12)
        ax.yaxis.grid(color='lightgray', linestyle=':')
        ax.xaxis.grid(color='lightgray', linestyle=':')
        ax.spines['right'].set_visible(False)
        ax.spines['top'].set_visible(False)

        if increment == 9: break
    if increment == 9: break

#_______________________________________
# I set the legend: abreviation -> airline name
comp_handler = []
for i in range(5):
    comp_handler.append(mpatches.Patch(color = LABEL_COLOR_MAP[i], label = i))

plt.legend(handles=comp_handler, bbox_to_anchor=(1.1, 0.97),
           title='Cluster', facecolor = 'lightgrey',
           shadow = True, frameon = True, framealpha = 1,
           fontsize = 13, bbox_transform = plt.gcf().transFigure)

plt.show()

"""___
## Customer categories

### Formatting data

"""

corresp = dict()
for key, val in zip (liste_produits, clusters):
    corresp[key] = val
#________________________________________________________________
df_cleaned['categ_product'] = df_cleaned.loc[:, 'Description'].map(corresp)

"""___
#### 4.1.1 Grouping products

"""

for i in range(5):
    col = 'categ_{}'.format(i)
    df_temp = df_cleaned[df_cleaned['categ_product'] == i]
    price_temp = df_temp['UnitPrice'] * (df_temp['Quantity'] - df_temp['QuantityCanceled'])
    price_temp = price_temp.apply(lambda x:x if x > 0 else 0)
    df_cleaned.loc[:, col] = price_temp
    df_cleaned[col].fillna(0, inplace = True)
#_____________________________________________________________________________________________
df_cleaned[['InvoiceNo', 'Description', 'categ_product', 'categ_0', 'categ_1', 'categ_2', 'categ_3','categ_4']][:5]

CustomerIDcateg_product = df_cleaned[['CustomerID','InvoiceNo', 'Description', 'categ_product', 'categ_0', 'categ_1', 'categ_2',
'categ_3','categ_4']][::]

df = pd.DataFrame(CustomerIDcateg_product)
df

df.to_csv('CustomerIDcateg_product.csv',index=True)

```

รายละเอียดโค้ดการสร้างแบบจำลองการจัดกลุ่มด้วยเทคนิคอาร์เอฟเอ็ม เพื่อใช้ในการแบ่งกลุ่มลูกค้าจากข้อมูลการซื้อขายสินค้า

```
# ทำการ Import Libraries ที่ใช้ในการจัดกลุ่มด้วยเทคนิคอาร์เอฟเอ็ม
import numpy as np
import pandas as pd

# ทำการ Load the dataset ซึ่งฟีเจอร์ที่ได้จากขั้นตอนการวิเคราะห์ข้อมูลลูกค้าด้วยเทคนิคอาร์เอฟเอ็ม
raw_data = pd.read_excel("DatasetRFM.xlsx")
raw_data.head()

# ติดตั้ง PyCaret

pip install pycaret

# ทำการลบคอลัมน์ CustomerID
raw_data.drop('CustomerID', axis = 1, inplace= False)

# ทำการ import clustering module
from pycaret.clustering import *

# ทำการติดตั้ง intialize setup
exp_clu = setup(raw_data)

# ทำการสร้าง k-means model
kmeans = create_model('kmeans', num_clusters = 4)

# ทำการ Assign labels using trained model
kmeans_df = assign_model(kmeans)

# ใช้ CustomerID เป็น index
kmeans_df.set_index('CustomerID',inplace=True)

kmeans_df.head()

# ทำการ Save ผลลัพธ์ที่ได้จากการจัดกลุ่มลง Drive
kmeans_df.to_csv('ClusteringCategoryProductRFM.csv',index=True)

# ทำการแสดงผลของโมเดลด้วยกราฟแบบ 3 มิติ
plot_model(kmeans,plot = 'tsne')

# ทำการแสดงค่า Optimal Cluster ด้วย กราฟ Elbow
plot_model(kmeans, plot = 'elbow')

# ทำการแสดงค่า Optimal Cluster ด้วย กราฟ Silhouette
plot_model(kmeans, plot = 'silhouette')

# ทำการแสดงผลลัพธ์ของการแบ่งกลุ่มด้วยเทคนิค RFM ด้วย กราฟ Distribution
plot_model(kmeans, plot = 'distribution') #to see size of clusters
```

รายละเอียดโค้ดการสร้างแบบจำลองการจัดกลุ่มด้วยเทคนิคอาร์เอฟเอ็มและการวิเคราะห์ข้อความ เพื่อใช้ในการแบ่งกลุ่มลูกค้าจากข้อมูลการซื้อขายสินค้า

```
# ทำการ Import Libraries ที่ใช้ในการจัดกลุ่มด้วยเทคนิคอาร์เอฟเอ็มและการวิเคราะห์ข้อความ
import numpy as np
import pandas as pd

# ทำการ Load the dataset ซึ่งฟีเจอร์ที่ได้จากขั้นตอนการวิเคราะห์ข้อมูลลูกค้าด้วยเทคนิคอาร์เอฟเอ็มและการวิเคราะห์ข้อความ
raw_data = pd.read_excel("DatasetRFMcateg.xlsx")
raw_data.head()

# ติดตั้ง PyCaret

pip install pycaret

# ทำการลบคอลัมน์ CustomerID
raw_data.drop('CustomerID', axis = 1, inplace= False)

# ทำการ import clustering module
from pycaret.clustering import *

# ทำการติดตั้ง intialize setup
exp_clu = setup(raw_data)

# ทำการสร้าง k-means model
kmeans = create_model('kmeans', num_clusters = 4)

# ทำการ Assign labels using trained model
kmeans_df = assign_model(kmeans)

# ใช้ CustomerID เป็น index
kmeans_df.set_index('CustomerID',inplace=True)

# แสดงค่า 5 แถวแรกของข้อมูล
kmeans_df.head()

# ทำการ Save ผลลัพธ์ที่ได้จากการจัดกลุ่มลง Drive
kmeans_df.to_csv('ClusteringCategoryProductRFM.csv',index=True)

# ทำการแสดงผลของโมเดลด้วยกราฟแบบ 3 มิติ
plot_model(kmeans,plot = 'tsne')

# ทำการแสดงค่า Optimal Cluster ด้วย กราฟ Elbow
plot_model(kmeans, plot = 'elbow')

# ทำการแสดงค่า Optimal Cluster ด้วย กราฟ Silhouette
plot_model(kmeans, plot = 'silhouette')

# ทำการแสดงผลลัพธ์ของการแบ่งกลุ่มด้วยเทคนิค RFM ด้วย กราฟ Distribution
plot_model(kmeans, plot = 'distribution') #to see size of clusters
```

ประวัติผู้เขียน

ชื่อ-สกุล	นางเอกปรียา ใบสนิ
วัน เดือน ปี เกิด	8 กุมภาพันธ์ 2533
สถานที่เกิด	สงขลา
วุฒิการศึกษา	พ.ศ. 2551 บริหารธุรกิจบัณฑิต สาขาการจัดการเทคโนโลยีสารสนเทศ จาก มหาวิทยาลัยสงขลานครินทร์ พ.ศ. 2562 วิทยาศาสตรมหาบัณฑิต สาขาวิทยาการข้อมูล จาก มหาวิทยาลัยศรีนครินทรวิโรฒ