



## รายงาน

### RFM to analyze customer behavior

#### คณะผู้จัดทำ

นายณัชพล หมื่นศรีชัย 65050253

นางสาวนฤกร สดวรรค 65050452

นางสาวสุกัญยาภรณ์ ฤดีชุติพร 65050916

#### เสนอ

รศ.ดร.ละออ บุญเกษม

รายงานฉบับนี้เป็นส่วนหนึ่งของรายวิชา Python Programming 05016385

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

คณะวิทยาศาสตร์ สาขาคณิตศาสตร์ประยุกต์

ภาคการศึกษาที่ 2 ปีการศึกษา 2567

## คำนำ

การจัดทำรายงาน “RFM to Analyze Customer Behavior” นี้ เป็นส่วนหนึ่งของวิชา Python Programming (05016385) ในหลักสูตรวิทยาศาสตรบัณฑิต สาขาคณิตศาสตร์ประยุกต์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง มีวัตถุประสงค์เพื่อเสริมสร้างความรู้ในการเขียนโปรแกรม Python และประยุกต์ใช้ RFM Analysis ในการวิเคราะห์พฤติกรรมลูกค้า

คณะผู้จัดทำเล็งเห็นว่าเรื่องนี้ น่าสนใจและเป็นประโยชน์ต่อการวิเคราะห์ข้อมูลเชิงลึกทางธุรกิจ ทางคณะผู้จัดทำต้องขอขอบพระคุณ รศ.ดร.ละออ บุญเกษม อาจารย์ที่ปรึกษา รายงานที่ได้ให้คำปรึกษา ข้อเสนอแนะ แนวคิด ตลอดจนการแก้ไขข้อบกพร่องต่างๆ ทางผู้จัดทำหวังว่า รายงานเล่มนี้จะเป็นประโยชน์กับผู้อ่าน หรือนักเรียน นักศึกษาที่กำลังหาข้อมูลเรื่องนี้อยู่ หากมีข้อเสนอแนะหรือข้อผิดพลาดประการใด ผู้จัดทำขอน้อมรับไว้และขออภัยมา ณ ที่นี้ด้วย

คณะผู้จัดทำเล็งเห็นว่าเรื่องนี้ น่าสนใจและเป็นประโยชน์ต่อการวิเคราะห์ข้อมูลเชิงลึกทางธุรกิจ ขอขอบพระคุณ รศ.ดร.ละออ บุญเกษม อาจารย์ที่ปรึกษา ที่ให้คำแนะนำและข้อเสนอแนะ คณะผู้จัดทำหวังว่ารายงานนี้จะเป็นประโยชน์ต่อผู้สนใจ และขอน้อมรับคำแนะนำด้วยความยินดี

คณะผู้จัดทำ

10 มกราคม 2568

## สารบัญ

คำนำ	ก
สารบัญ	ข
สารบัญรูป	ค
บทที่ 1 ที่มาและความสำคัญ	1
1.1 ที่มาและความสำคัญ	1
1.2 วัตถุประสงค์	1
1.3 ประโยชน์ที่คาดว่าจะได้รับ	1
บทที่ 2 เอกสารที่เกี่ยวข้อง ภาษา และเครื่องมือที่ใช้งาน	2
2.1 RFM Model	2
2.2 Python	6
2.3 ความแตกต่างระหว่าง Dashboard และ Report	12
2.4 ธุรกิจ E-Commerce	13
บทที่ 3 User Interface	16
3.1 หน้าแรก	16
3.2 ผลลัพธ์หลังจากอัปโหลดไฟล์ Data_sample	16
3.3 ผลลัพธ์หลังจากอัปโหลดไฟล์ OnlineRetail	17
บทที่ 4 Coding	25
บทที่ 5 อภิปรายผลการดำเนินงาน ข้อเสนอแนะ และแนวทางแก้ไข	28
5.1 สรุปผลการดำเนินงาน	28
5.2 ข้อเสนอแนะ	29
5.3 ปัญหาที่พบ และแนวทางการแก้ไขปัญหา	29
บรรณานุกรม	30

## สารบัญรูป

รูปที่ 1: กราฟ RFM Segments Map	3
รูปที่ 2: หน้าแรก	16
รูปที่ 3: ผลลัพธ์หลังจากอัปโหลดไฟล์ Data_sample	16
รูปที่ 4: ผลลัพธ์ในส่วนของแต่ละ Dashboard (1)	17
รูปที่ 5: ผลลัพธ์ในส่วนของแต่ละ Dashboard (2)	17
รูปที่ 6: ผลลัพธ์ในส่วนของแต่ละ Dashboard (3)	18
รูปที่ 7: ผลลัพธ์ในส่วนของแต่ละ Dashboard (4)	18
รูปที่ 8: ผลลัพธ์ในส่วนของแต่ละ Dashboard (5)	19
รูปที่ 10: ผลลัพธ์ในส่วนของแต่ละ Dashboard (7)	20
รูปที่ 11: ผลลัพธ์ในส่วนของแต่ละ Dashboard (8)	20
รูปที่ 12: ผลลัพธ์ในส่วนของแต่ละ Dashboard (9)	21
รูปที่ 13: ผลลัพธ์ในส่วนของแต่ละ Summarizing (1)	21
รูปที่ 14: ผลลัพธ์ในส่วนของแต่ละ Summarizing (2)	22
รูปที่ 15: ผลลัพธ์ในส่วนของแต่ละ Summarizing (3)	22
รูปที่ 16: ผลลัพธ์ในส่วนของแต่ละ Summarizing (4)	23
รูปที่ 17: ผลลัพธ์ในส่วนของแต่ละ Summarizing (5)	23
รูปที่ 18: ผลลัพธ์ในส่วนของแต่ละ Summarizing (6)	24

## บทที่ 1

### ที่มาและความสำคัญ

#### 1.1 ที่มาและความสำคัญ

ในปัจจุบัน ธุรกิจจะต้องเผชิญกับการแข่งขันที่สูงขึ้นและความต้องการของลูกค้าที่เปลี่ยนแปลงอยู่ตลอดเวลา การเข้าใจพฤติกรรมและการแบ่งกลุ่มลูกค้าอย่างมีประสิทธิภาพจึงกลายเป็นสิ่งสำคัญในการวางแผนการตลาดและสร้างความสัมพันธ์ที่ยั่งยืน ซึ่ง RFM (Recency, Frequency, Monetary) เป็นเครื่องมือที่ช่วยวิเคราะห์พฤติกรรมลูกค้าได้อย่างมีประสิทธิภาพ โดยสามารถระบุและจัดกลุ่มลูกค้าตามความสำคัญของการซื้อในอดีต ซึ่งช่วยให้ธุรกิจปรับกลยุทธ์ให้เหมาะสมกับความต้องการเฉพาะของลูกค้าแต่ละกลุ่ม

ดังนั้น ทางคณะผู้จัดทำจึงมีความสนใจที่จะพัฒนาโปรแกรมสำหรับวิเคราะห์พฤติกรรมลูกค้าด้วย RFM และเล็งเห็นว่าการใช้ Python จะช่วยให้การวิเคราะห์นี้รวดเร็วและแม่นยำยิ่งขึ้น ด้วยความสามารถของ Python ในการจัดการข้อมูลจำนวนมากและสร้างภาพข้อมูลที่ชัดเจน โปรแกรมนี้จะช่วยให้ธุรกิจสามารถวิเคราะห์ข้อมูลเชิงลึกได้ง่ายขึ้นผ่านกราฟและแดชบอร์ดต่างๆ ซึ่งจะช่วยในการตัดสินใจเชิงกลยุทธ์ได้อย่างมีประสิทธิภาพ

#### 1.2 วัตถุประสงค์

- 1.2.1. เพื่อศึกษาเพิ่มเติมเกี่ยวกับภาษา Python และนำความรู้ที่ได้เรียนมาประยุกต์ใช้
- 1.2.2. เพื่อวิเคราะห์และจัดกลุ่มลูกค้าตามพฤติกรรมการซื้อด้วย RFM
- 1.2.3. เพื่อสร้างเครื่องมือที่จะช่วยวิเคราะห์ข้อมูลเชิงลึก
- 1.2.4. เพื่อสร้างกราฟและแดชบอร์ดแสดงผลข้อมูล

#### 1.3 ประโยชน์ที่คาดว่าจะได้รับ

- 1.3.1. มีความเข้าใจเกี่ยวกับภาษา Python มากยิ่งขึ้นและมองเห็นแนวทางพัฒนาทักษะจากการใช้งานภาษา Python
- 1.3.2. เข้าใจความต้องการและแนวโน้มการซื้อของลูกค้ามากขึ้น
- 1.3.3. ช่วยให้ธุรกิจสามารถปรับกลยุทธ์การตลาดให้เหมาะสมกับแต่ละกลุ่มลูกค้า
- 1.3.4. ช่วยเพิ่มโอกาสในการสร้างรายได้และเพิ่มยอดขายให้กับธุรกิจ

## บทที่ 2

### เอกสารที่เกี่ยวข้อง ภาษา และเครื่องมือที่ใช้งาน

ในการจัดทำโปรแกรมวิเคราะห์พฤติกรรมลูกค้า ทางคณะผู้จัดทำได้ ทำการศึกษาเอกสารที่เกี่ยวข้อง ภาษา และเครื่องมือที่ใช้ ดังนี้

#### 2.1 RFM Model

RFM Model คือการทำ Customer Segmentation หรือ การแบ่งกลุ่มลูกค้าตาม พฤติกรรมการซื้อที่แตกต่างกัน โดยนำข้อมูลหลัก 3 อย่างมาวิเคราะห์ ได้แก่ Recency Frequency และ Monetary และให้ Score เพื่อจัดอันดับความสำคัญของกลุ่มลูกค้าได้อย่างแม่นยำ ดังนี้

R – RECENCY: ข้อมูลของลูกค้าที่มาซื้อสินค้าหรือใช้บริการครั้งล่าสุดเมื่อไหร่ เช่น ลูกค้า A มาซื้อสินค้าเมื่อวันที่ 21 ก.ค. 2566 เวลา 15.36 น. หากลูกค้าเพิ่งมาซื้อสินค้าหรือใช้บริการไปเมื่อไม่นานมานี้ Score ที่จะให้ลูกค้ารายนั้นยิ่งสูง เพราะลูกค้าเพิ่งจะมี Engage กับเราไป ทำให้ง่ายต่อการส่ง Conversation เพื่อให้ลูกค้าประทับใจและกลับมาซื้อสินค้าหรือใช้บริการซ้ำอีกครั้ง

F – FREQUENCY: ความถี่ของลูกค้าในการมาซื้อสินค้าหรือใช้บริการ เช่น ลูกค้า B มาซื้อสินค้า 2 ครั้งต่อสัปดาห์ แต่ลูกค้า C มาซื้อสินค้าทุกวันเป็นประจำ หากลูกค้าคนใดมาใช้บริการบ่อย Score ที่ให้ก็จะยิ่งสูง แต่ข้อควรระวังคือ ต้องดูจำนวนการซื้อที่เกิดขึ้นด้วย เพราะถ้าลูกค้ามาเป็นประจำแต่ไม่ได้ซื้อสินค้า นั่นเป็นสัญญาณที่บอกว่าลูกค้ายังไม่มี ความไว้วางใจในสินค้าและบริการ นักการตลาดต้องรีบหาทางสร้างความน่าเชื่อถือให้กับลูกค้าโดยเร็ว

M – MONETARY: ปริมาณการซื้อหรือใช้บริการของลูกค้า เช่น ลูกค้า D ซื้อสินค้าเพียง 2 ชิ้น แต่ลูกค้า E ซื้อสินค้า 8 ชิ้น ลูกค้าที่ซื้อสินค้าหรือใช้บริการที่มากกว่า ก็จะได้ Score ที่สูงกว่า แต่หากซื้อสินค้าหรือใช้บริการน้อย รวมไปถึงกลุ่มลูกค้าที่ไม่ได้ซื้อสินค้าหรือใช้บริการเลย แต่ยังมี Engage อยู่ เช่น แวะมาดูสินค้าที่เว็บไซต์ นำสินค้าเข้าตะกร้าแต่ไม่ได้กดชำระเงินก็ได้รับ Score น้อย

##### 2.1.1 เทคนิคการแบ่งกลุ่มลูกค้าด้วย กลยุทธ์ RFM

เทคนิคการคำนวณที่จะใช้ใน RFM Analysis คือการให้ Score ลูกค้าแต่ละคน ตั้งแต่ 1 จนถึง 5 ใน 3 แกน ได้แก่ Recency, Frequency และ Monetary โดยแต่ละแกนจะคำนวณอ้างอิงกับลูกค้าทุกคนในร้าน ดังนั้น การทำธุรกิจที่ดีจึงควรมีการเก็บข้อมูลของลูกค้า และนำมาวิเคราะห์ RFM Analysis เพื่อหากกลุ่มลูกค้าที่สำคัญต่อบริษัท เพราะจะช่วยให้สามารถใช้ทรัพยากรเงิน เวลา และกำลังคนที่มีอย่างจำกัดได้อย่างคุ้มค่า นอกจากนั้นข้อมูลที่ได้ยังสามารถทำให้นักการตลาดคิดหา กลยุทธ์ทางการตลาด หรือการทำแคมเปญต่างๆได้อย่างมีประสิทธิภาพมากยิ่งขึ้น

ตัวอย่างการให้ Score ของแต่ละแกน

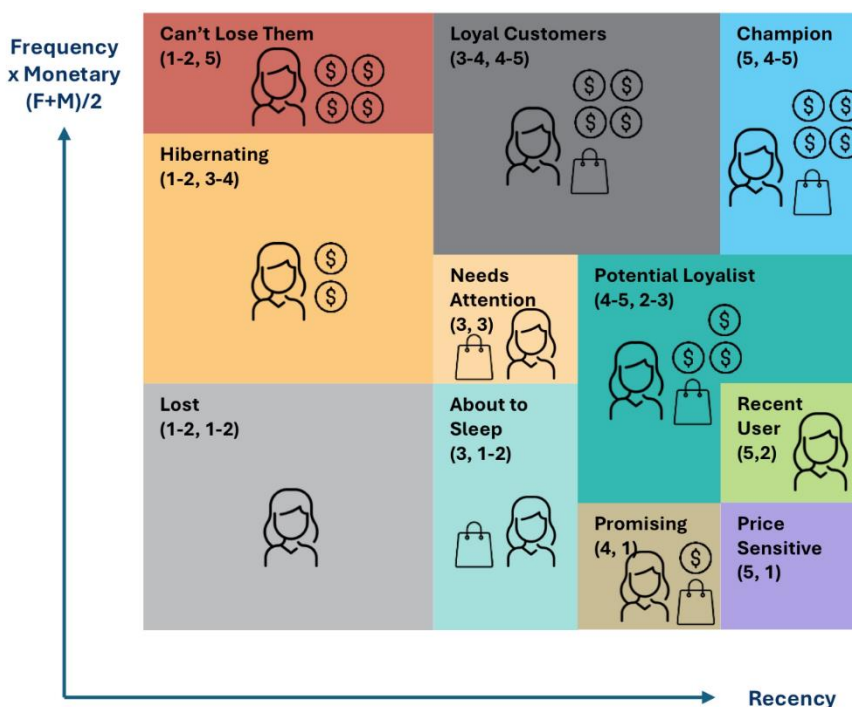
Recency : ลูกค้า A สั่งซื้อสินค้าล่าสุดวันนี้ จะได้ R Score 5 คะแนน ส่วนลูกค้า B สั่งซื้อสินค้าล่าสุดเมื่อ 3 ปีที่แล้ว จะได้ R Score 1 คะแนน

Frequency : ลูกค้า A สั่งซื้อสินค้าบ่อยที่สุดในร้าน จะได้ F Score 5 คะแนน ส่วนลูกค้า B สั่งซื้อสินค้าน้อยครั้งที่สุดในร้าน จะได้ F Score 1 คะแนน

Monetary : ลูกค้า A ใช้จ่ายเยอะที่สุดในร้าน จะได้ M Score 5 คะแนน ส่วนลูกค้า B ใช้จ่ายน้อยที่สุดในร้าน จะได้ M Score 1 คะแนน

เมื่อเราให้ R Score, F Score และ M Score กับลูกค้าทุกคนแล้ว เราจะนำข้อมูลลูกค้าแต่ละคน มาจัดเข้ากลุ่มด้วยตาราง 5x5 ช่อง โดยใช้ R แทนค่าในแกน X และ  $(F+M)/2$  แทนค่าในแกน Y

ตาราง RFM Analysis มีทั้งหมด 25 ช่อง (5x5) โดยอ้างอิงจาก Score ในแกน X และแกน Y



ของลูกค้าแต่ละคน เพื่อแบ่งกลุ่มความสำคัญของลูกค้า เมื่อหา Score ของลูกค้าแต่ละคนได้แล้ว ลูกค้าแต่ละคนก็จะถูกจัดเข้ากลุ่มนั้นๆโดยทันที

รูปที่ 1: กราฟ RFM Segments Map

การทำ RFM Analysis ให้มีประสิทธิภาพจำเป็นต้องเข้าใจกลุ่มลูกค้าเป้าหมายก่อน เพราะเราควร แยก Message ให้สัมพันธ์กับกลุ่มลูกค้าเป้าหมาย ถึงแม้ว่ากลุ่มคนเหล่านั้นจะเคยซื้อสินค้า และเป็น

ลูกค้าที่เคยใช้จ่ายเหมือนกัน แต่พฤติกรรมการซื้อที่ต่างกันทำให้เราควรแยกโจทย์หรือจุดประสงค์ในการพูดคุยไว้ด้วย ซึ่งจะทำให้การแบ่งกลุ่มลูกค้าออกเป็น 11 กลุ่มประกอบไปด้วย

1. Champions: กลุ่มลูกค้าที่สำคัญที่สุด เนื่องจากลูกค้ากลุ่มนี้เพิ่งซื้อสินค้าหรือใช้บริการไป นอกจากนั้นยังมีอัตราการซื้อซ้ำ และปริมาณการใช้จ่ายสูงที่สุดอีกด้วย

คำแนะนำ : ควรรักษาลูกค้ากลุ่มนี้ไว้ให้ดีที่สุด ห้ามปล่อยลูกค้ากลุ่มนี้ไปหาคู่แข่งเด็ดขาด

2. Loyal Customers: กลุ่มลูกค้าที่มีอัตราการใช้จ่าย และอัตราการซื้อซ้ำที่สูงกว่าลูกค้าทั่วไป ถึงแม้ว่าเมื่อเร็ว ๆ นี้ลูกค้ากลุ่มนี้จะได้ซื้อสินค้าหรือใช้บริการบ้างก็ตาม แต่ก็ถือว่าเป็นลูกค้ากลุ่ม 20% ตามหลักของ Pareto

คำแนะนำ : ทำความเข้าใจลูกค้ากลุ่มนี้ว่าแต่ละคนชอบอะไร เหตุผลที่เขาซื้อสินค้าหรือใช้บริการจากเราคืออะไร จากนั้นจึงทำการสื่อสารให้ตรงจุดกับลูกค้า และอย่าลืมมอบโปรโมชั่นหรือสิทธิประโยชน์พิเศษสำหรับลูกค้ากลุ่มนี้เสมอ

3. Potential Loyalist: กลุ่มลูกค้าที่ไม่ได้ซื้อสินค้าหรือใช้บริการบ่อย แต่ลูกค้ากลุ่มนี้เพิ่งมีการสั่งซื้อเข้ามา และมียอดใช้จ่ายที่สูงพอสมควร นั่นทำให้ลูกค้ากลุ่มนี้มีความเป็นไปได้สูงที่จะขยับขึ้นไปเป็น Loyal Customers

คำแนะนำ : ควรเร่งสร้างความประทับใจให้ลูกค้ากลุ่มนี้เชื่อมั่นว่าเราใส่ใจและให้ความสำคัญกับ Value proposition ที่เราเสนอ เพื่อให้เขาขยับขึ้นไปเป็น Loyal Customers

4. New Customers: กลุ่มลูกค้าที่เพิ่งสั่งซื้อไปเมื่อไม่นานมานี้ และเป็นออเดอร์แรกของลูกค้าด้วย ลูกค้ากลุ่มนี้จึงอาจจะยังไม่กล้าใช้จ่ายในมูลค่าที่สูงมากนัก จึงเป็นกลุ่มที่ง่ายที่สุดในการสร้าง Positive brand experience

คำแนะนำ : ควรเร่งสร้างความประทับใจให้ลูกค้ากลุ่มนี้รู้สึกว่าคุณค่าที่ซื้อสินค้าหรือใช้บริการของเรา และอย่าให้ลูกค้ากลุ่มนี้รู้สึกแฉ้งแฉ่องหรือสงสัย หากทางไปต่อไม่เจอหลังจากที่ซื้อสินค้ากับเรา

5. Promising: กลุ่มลูกค้าที่เพิ่งซื้อสินค้าหรือใช้บริการครั้งแรกไปไม่นานเท่าไร แต่มีอัตราการซื้อซ้ำ และมูลค่าการใช้จ่ายที่ยังต่ำอยู่ ลูกค้ากลุ่มนี้สนใจสินค้าและบริการของเราแต่ไม่มากพอที่จะเอาเข้ามาเป็นส่วนหนึ่งในชีวิตของเขา

คำแนะนำ : นำเสนอสินค้าหรือบริการที่ตรงกับ Personalization กับลูกค้าให้มากขึ้น แล้วทำ Upsell หรือ Re-Purchase แคมเปญอ้างอิงถึงสินค้าเดิมที่เคยซื้อไป



6. Need Attention: กลุ่มลูกค้าที่มียอดการสั่งซื้อ อัตราการซื้อซ้ำ และช่วงเวลาของการซื้อครั้งล่าสุด อยู่ที่ค่าเฉลี่ย หากเราไม่ Engage ลูกค้ากลุ่มนี้อาจจะทำให้เขาสูญเสียความสนใจที่มีต่อสินค้าหรือบริการของเราได้

คำแนะนำ : ให้ข้อเสนอแบบจำกัดเวลาเพื่อกระตุ้นการกลับมาซื้อสินค้าหรือใช้บริการซ้ำ หรือนำเสนอประโยชน์ของสินค้าที่เคยซื้อไปในแง่มุมอื่น เพื่อให้ลูกค้ากลุ่มนี้เกิดความสนใจในสินค้าหรือบริการของเรามากยิ่งขึ้น

7. About to Sleep: กลุ่มลูกค้าที่ไม่ได้มีการสั่งซื้อมาสักพักแล้ว ถึงแม้ว่ายอดการซื้อซ้ำและมูลค่าของออเดอร์ที่ผ่านมาของลูกค้ากลุ่มนี้ต่ำกว่าค่าเฉลี่ย แต่ยังถือว่าเป็นฐานลูกค้าของเราอยู่ หากไม่รีบดึงกลับมาลูกค้ากลุ่มนี้จะหายไปอย่างแน่นอน

คำแนะนำ : ให้ข้อเสนอแบบจำกัดเวลาสำหรับสินค้ายอดนิยม หรือทำ Re-Purchase แคมเปญอ้างอิงถึงสินค้าเดิมที่เคยซื้อไปสำหรับลูกค้ากลุ่มนี้

8. Can't Lose: กลุ่มลูกค้าที่เคยเป็น Champions มีการซื้อสินค้าหรือใช้บริการบ่อยมากๆ และมีมูลค่าการใช้จ่ายสูงที่สุดในร้าน แต่ไม่ได้กลับมาซื้อสินค้าหรือใช้บริการนานมากๆ แล้ว ซึ่งอาจเป็นไปได้ว่าลูกค้ากลุ่มนี้ไปซื้อสินค้าของคู่แข่ง หรือเกิดความไม่พอใจจากความผิดพลาดของสินค้าหรือบริการของเรา

คำแนะนำ : หาทางแก้ไขปัญหาที่เกิดขึ้น โดยการโทรสอบถามเป็นรายบุคคล หรือให้โปรโมชั่นพิเศษหลายๆกับพวกเขาเพื่อให้ลูกค้ากลุ่มนี้กลับมาให้ได้

9. At Risk

คือ กลุ่มลูกค้าที่เคยมียอดใช้จ่าย และการซื้อซ้ำที่สูงมาก แต่ไม่ได้กลับมาซื้อสินค้าหรือใช้บริการเลย ในช่วงที่ผ่านมา อาจเป็นไปได้ว่าในช่วงที่ผ่านมาจะมีอะไรบางอย่างที่ทำให้ลูกค้ากลุ่มนี้ลืมเราไป

คำแนะนำ : หาให้เจอว่ามีอะไรที่เปลี่ยนแปลงไป แล้วเราต้องทำอย่างไรเพื่อให้ลูกค้ากลุ่มนี้กลับมา อาจจะออก Exclusive promotion สำหรับลูกค้ากลุ่มนี้ และแนะนำให้รีบกลับไปดูการบริการ, คุณภาพของสินค้า และ Customer journey หากลูกค้าในกลุ่มนี้มีจำนวนเยอะมากๆ

10. Hibernating: กลุ่มลูกค้าที่เคยซื้อสินค้าหรือใช้บริการไปเมื่อนานมาแล้ว ปริมาณการซื้อและความถี่ในการซื้อน้อย แต่อาจจะคุ้มค่าและมีความเป็นไปได้ที่จะดึงลูกค้ากลุ่มนี้กลับมา เพราะลูกค้ากลุ่มนี้ยังจำเราได้และประวัติการสั่งซื้ออยู่ในระดับที่พอรับได้

คำแนะนำ : Recreate Brand Value กับลูกค้ากลุ่มนี้อีกครั้ง โดยเน้นจุดยืนถึงวิธีการแก้ไขปัญหาของเรา ทำโปรโมชั่นแบบจำกัดเวลาสำหรับสินค้ายอดนิยมหรือการทำ re-purchase แคมเปญอ้างอิงถึงสินค้าเดิมที่เคยซื้อไป

11. Lost: กลุ่มลูกค้าที่ไม่ได้กลับมาซื้อสินค้าหรือใช้บริการเลย และจำนวนกับมูลค่าของออเดอร์ที่เคยสั่งซื้อไปอยู่ในระดับที่ต่ำมาก เป็นไปได้ว่าลูกค้ากลุ่มนี้เลิกใช้สินค้าของเราไปแล้ว หรือหันไปใช้สินค้าหรือบริการจากคู่แข่ง หรือ อาจจะมีความรู้สึกที่เป็นลบต่อแบรนด์และสินค้าของเรา ซึ่งถือเป็นเรื่องยากมากที่จะ Engage ให้ลูกค้ากลุ่มนี้กลับมาอีกครั้ง

## 2.2 Python

Python เป็นภาษาการเขียนโปรแกรมที่เรียนรู้ได้ง่ายและทรงพลัง โดยมีโครงสร้างข้อมูลระดับสูงที่มีประสิทธิภาพ และมีวิธีการเขียนโปรแกรมเชิงวัตถุที่เรียบง่ายแต่มีประสิทธิภาพ ความสวยงามของไวยากรณ์และการกำหนดชนิดข้อมูลแบบไดนามิก พร้อมกับการที่ Python เป็นภาษาแบบตีความ (interpreted language) ทำให้เหมาะสำหรับการพัฒนาแอปพลิเคชันอย่างรวดเร็วและการเขียนสคริปต์ในหลากหลายแพลตฟอร์ม

### 2.2.1 สาเหตุที่ Python เป็นที่นิยม:

2.2.1.1 ง่ายต่อการใช้งาน: Python เหมาะกับหลากหลายงาน เช่น การเขียนโปรแกรม GUI, งานฐานข้อมูล, หรือเกม โดยไม่ต้องใช้เวลาในการพัฒนาเท่ากับการเขียนโปรแกรมใน C/C++/Java

2.2.1.2 เครื่องมือทรงพลัง: Python มีชนิดข้อมูลระดับสูงในตัว เช่น อาร์เรย์และดิกชันนารี ทำให้เหมาะกับงานที่ซับซ้อน

2.2.1.3 รองรับโมดูลและไลบรารี: Python ช่วยให้สามารถแบ่งโปรแกรมเป็นโมดูลที่น่ากลับมาใช้ใหม่ได้ และมีโมดูลมาตรฐานที่ครอบคลุม เช่น การจัดการไฟล์ การเชื่อมต่อเครือข่าย และ GUI

2.2.1.4 ภาษาที่แปลผล (Interpreted Language): Python ไม่ต้องผ่านกระบวนการคอมไพล์ ทำให้พัฒนาซอฟต์แวร์ได้รวดเร็ว และสามารถทดลองพีเจอร์ท่าง ๆ ได้แบบ interact

### 2.2.2 ข้อได้เปรียบของ Python:

2.2.2.1 เขียนโปรแกรมได้กระชับและอ่านง่าย: การใช้ข้อมูลระดับสูงช่วยลดความซับซ้อนในโค้ด ใช้การย่อหน้า (indentation) แทนเครื่องหมายปีกกา ({} เพื่อจัดกลุ่มคำสั่ง และไม่จำเป็นต้องประกาศตัวแปรหรืออาร์กิวเมนต์

2.2.2.2 การขยายความสามารถ: สามารถเพิ่มฟังก์ชันหรือโมดูลใหม่ใน Python ด้วยภาษา C เพื่อเพิ่มประสิทธิภาพ และใช้ Python เป็นส่วนเสริมในแอปพลิเคชันที่เขียนด้วย C

2.2.2.3 ใช้งานได้หลากหลาย: Python สามารถประยุกต์ใช้ในงานที่หลากหลายกว่าภาษา Awk หรือ Perl และยังคงใช้งานง่าย

## 2.2.3 More Control Flow Tools

### 2.2.3.1 if Statements

ประเภทคำสั่งที่รู้จักกันดีที่สุดคือคำสั่ง if ตัวอย่างเช่น:

```
x = int(input("Please enter an integer: "))

if x < 0:
    x = 0
    print('Negative changed to zero')
elif x == 0:
    print('Zero')
elif x == 1:
    print('Single')
else:
    print('More')
```

ใน Python: โครงสร้างคำสั่ง if สามารถมี elif ได้ตั้งแต่ 0 ครั้งขึ้นไป และส่วน else เป็นส่วนที่ไม่จำเป็น (optional) คำว่า elif ย่อมาจาก "else if" ซึ่งช่วยลดการเยื้องบรรทัด (indentation) ที่มากเกินไปเมื่อเทียบกับการใช้ if ซ้อนหลายระดับ การใช้ if ... elif ... elif ... สามารถใช้แทนโครงสร้าง switch หรือ case ในภาษาโปรแกรมอื่น ๆ ได้

### 2.2.3.2 for Statements

คำสั่ง for ในภาษา Python มีความแตกต่างเล็กน้อยจากภาษา C หรือ Pascal แทนที่จะวนซ้ำโดยใช้ความก้าวหน้าของตัวเลขเชิงเลขคณิตเสมอ (เช่นใน Pascal) หรือเปิดโอกาสให้ผู้ใช้กำหนดทั้งขั้นตอนการวนซ้ำและเงื่อนไขหยุด (เช่นใน C) คำสั่ง for ใน Python จะวนซ้ำผ่านรายการของสิ่งต่างๆ ในลำดับใดๆ (เช่นรายการหรือสตริง) โดยเรียงลำดับตามที่ปรากฏในลำดับนั้น

ตัวอย่างเช่น:

```
# Measure some strings:
words = ['cat', 'window', 'defenestrate']
```

```
for w in words:
    print(w, len(w))
```

การเขียนโค้ดที่แก้ไขคอลเลกชันในขณะที่กำลังวนซ้ำในคอลเลกชันเดียวกัน อาจทำให้เกิดความซับซ้อนและข้อผิดพลาดได้ง่าย วิธีที่ตรงไปตรงมามากกว่าคือการวนซ้ำผ่านสำเนาของคอลเลกชัน หรือสร้างคอลเลกชันใหม่แทน เช่น:

```
# Create a sample collection
users = {'Hans': 'active', 'Éléonore': 'inactive', '景太郎': 'active'}

# Strategy: Iterate over a copy
for user, status in users.copy().items():
    if status == 'inactive':
        del users[user]

# Strategy: Create a new collection
active_users = {}
for user, status in users.items():
    if status == 'active':
        active_users[user] = status
```

### 2.2.3.3 The range() Function

หากคุณจำเป็นต้องวนซ้ำผ่านลำดับของตัวเลข ฟังก์ชันที่มีอยู่ในตัว range() จะช่วยให้สะดวกมาก มันจะสร้างลำดับความก้าวหน้าเชิงเลข:

```
for i in range(5):
    print(i)
```

จุดสิ้นสุดที่กำหนดจะไม่เป็นส่วนหนึ่งของลำดับที่สร้างขึ้นเสมอ range(10) จะสร้างค่าทั้งหมด 10 ค่า ซึ่งเป็นดัชนีที่ถูกต้องสำหรับรายการที่มีความยาว 10 การเริ่มต้นของลำดับสามารถกำหนดให้เริ่มจากตัวเลขอื่นได้ หรือสามารถระบุการเพิ่มค่าที่แตกต่างออกไป (แม้แต่ค่าลบ บางครั้งจะเรียกว่า ‘step’):

```
list(range(5, 10))
list(range(0, 10, 3))
list(range(-10, -100, -30))
```

เพื่อวนซ้ำผ่านดัชนีของลำดับ คุณสามารถใช้การรวม range() และ len() ดังนี้:

```
a = ['Mary', 'had', 'a', 'little', 'lamb']
for i in range(len(a)):
    print(i, a[i])
```

ในหลายๆด้าน วัตถุที่คืนค่าจาก range() ทำตัวเหมือนกับรายการ (list) แต่ในความเป็นจริงมันไม่ใช่ มันคือลักษณะของวัตถุที่คืนค่ารายการที่ตามลำดับเมื่อคุณวนซ้ำผ่านมัน แต่ไม่ได้สร้างรายการจริงๆ ซึ่งช่วยประหยัดพื้นที่

เราจะเรียกวัดุดเช่นนี้ว่า iterable หมายถึง สามารถใช้เป็นเป้าหมายสำหรับฟังก์ชันและโครงสร้างที่คาดหวังบางสิ่งที่สามารถดึงเอารายการที่ตามมาจนกว่าจะหมด ตัวอย่างเช่น คำสั่ง for เป็นโครงสร้างประเภทนี้ ขณะที่ฟังก์ชันที่รับ iterable เป็นอินพุต เช่น sum()

```
sum(range(4)) # 0 + 1 + 2 + 3
```

#### 2.2.3.4 break and continue Statements

คำสั่ง **break** จะออกจากการวนซ้ำในลูป **for** หรือ **while** ที่อยู่ภายในที่สุด:

```
for n in range(2, 10):
    for x in range(2, n):
        if n % x == 0:
            print(f'{n} equals {x} * {n//x}')
            break
```

คำสั่ง **continue** จะดำเนินการต่อไปยังการวนซ้ำรอบถัดไปของลูป:

```
for num in range(2, 10):
    if num % 2 == 0:
        print(f"Found an even number {num}")
        continue
    print(f"Found an odd number {num}")
```

### 2.2.3.5 Defining Functions

เราสามารถสร้างฟังก์ชันที่เขียนลำดับฟีโบนัชชี (Fibonacci series) ไปจนถึงขีดจำกัดที่กำหนดได้ดังนี้:

```
def fib(n): # write Fibonacci series less than n
    """Print a Fibonacci series less than n."""
    a, b = 0, 1
    while a < n:
        print(a, end=' ')
        a, b = b, a+b
    print()

# Now call the function we just defined:
fib(2000)
```

คำสั่ง **def** ใช้ในการกำหนดฟังก์ชัน มันต้องตามด้วยชื่อฟังก์ชันและรายการพารามิเตอร์ในวงเล็บ คำสั่งที่เป็นเนื้อหาของฟังก์ชันจะเริ่มที่บรรทัดถัดไป และต้องมีการเยื้อง (indentation)

คำสั่งแรกในเนื้อหาของฟังก์ชันอาจเป็นข้อความสตริง (string literal) ซึ่งเป็น docstring ของฟังก์ชัน มีเครื่องมือที่ใช้ docstrings เพื่อสร้างเอกสารออนไลน์หรือเอกสารพิมพ์โดยอัตโนมัติ หรือให้ผู้ใช้สามารถท่องไปตามโค้ดได้แบบอินเทอร์แอคทีฟ การใส่ docstrings ในโค้ดที่คุณเขียนเป็นการปฏิบัติที่ดี ดังนั้นควรทำให้เป็นนิสัย

การดำเนินการของฟังก์ชันจะสร้างตารางสัญลักษณ์ (symbol table) ใหม่ที่ใช้สำหรับตัวแปรท้องถิ่น (local variables) ของฟังก์ชัน โดยเฉพาะการกำหนดค่าตัวแปรทั้งหมดในฟังก์ชันจะเก็บค่าในตารางสัญลักษณ์ท้องถิ่น ในขณะที่การอ้างอิงตัวแปรจะค้นหาตารางสัญลักษณ์ท้องถิ่นก่อน แล้ว

ตามด้วยตารางสัญลักษณ์ของฟังก์ชันที่ครอบคลุม (enclosing functions) จากนั้นไปที่ตารางสัญลักษณ์ทั่วโลก (global symbol table) และสุดท้ายที่ตารางของชื่อที่สร้างขึ้นในตัวแปล (built-in names) ดังนั้น ตัวแปลทั่วโลกและตัวแปลของฟังก์ชันที่ครอบคลุมจะไม่สามารถกำหนดค่าโดยตรงภายในฟังก์ชัน (ยกเว้นสำหรับตัวแปลทั่วโลกที่ต้องใช้คำสั่ง global หรือสำหรับตัวแปลของฟังก์ชันที่ครอบคลุมที่ต้องใช้คำสั่ง nonlocal) ถึงแม้ว่าจะสามารถอ้างอิงได้

พารามิเตอร์จริง (arguments) ในการเรียกฟังก์ชันจะถูกนำเข้ามาในตารางสัญลักษณ์ท้องถิ่นของฟังก์ชันที่ถูกเรียกเมื่อมันถูกเรียก ดังนั้นพารามิเตอร์จะถูกส่งผ่านโดยใช้การเรียกตามค่า (call by value) ซึ่งค่าคือการอ้างอิงออบเจกต์ (ไม่ใช่ค่าของออบเจกต์) เมื่อฟังก์ชันเรียกฟังก์ชันอื่น หรือเรียกตัวมันเองแบบ recursive ตารางสัญลักษณ์ท้องถิ่นใหม่จะถูกสร้างขึ้นสำหรับการเรียกนั้น การกำหนดฟังก์ชันจะเชื่อมโยงชื่อฟังก์ชันกับออบเจกต์ของฟังก์ชันในตารางสัญลักษณ์ปัจจุบัน ตัวอินเทอร์พรีเตอร์จะรู้จักออบเจกต์ที่ชี้ไปโดยชื่อนั้นว่าเป็นฟังก์ชันที่ผู้ใช้กำหนด ชื่ออื่นๆ ก็สามารถชี้ไปที่ออบเจกต์ของฟังก์ชันเดียวกันและสามารถใช้เพื่อเข้าถึงฟังก์ชันได้เช่นกัน

```
fib
<function fib at 10042ed0>
>>> f = fib
>>> f(100)
0 1 1 2 3 5 8 13 21 34 55 89
```

หากเขียนภาษาอื่นๆ คุณอาจจะสงสัยว่า fib ไม่ใช่ฟังก์ชัน แต่เป็นเพียงกระบวนการ (procedure) เพราะมันไม่คืนค่ากลับ ในความเป็นจริงแม้แต่ฟังก์ชันที่ไม่มีคำสั่ง return ก็ยังคงคืนค่าบางอย่าง ค่านั้นเรียกว่า None การเขียนค่าของ None จะถูกยกเลิกโดยตัวอินเทอร์พรีเตอร์ หากมันจะเป็นค่าที่เขียนออกมาเพียงค่าเดียว แต่คุณสามารถเห็นมันได้หากต้องการจริงๆ โดยการใช่ print():

```
fib(0)
>>> print(fib(0))
None
```

การเขียนฟังก์ชันที่คืนค่าลิสต์ของตัวเลขจากลำดับฟีโบนัชชีแทนที่จะพิมพ์มันออกมาเป็นเรื่องง่าย:

```
def fib2(n): # return Fibonacci series up to n
...     """Return a list containing the Fibonacci series up to n."""
...     result = []
```

```

... a, b = 0, 1
... while a < n:
...     result.append(a) # see below
...     a, b = b, a+b
... return result
...
>>> f100 = fib2(100) # call it
>>> f100             # write the result
[0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]

```

ตัวอย่างนี้โดยปกติจะแสดงให้เห็นฟีเจอร์ใหม่ของ Python:

คำสั่ง return: คืนค่าจากฟังก์ชัน return ที่ไม่มีอาร์กิวเมนต์จะคืนค่า None การหลุดออกจากฟังก์ชัน (โดยไม่ใช้คำสั่ง return) ก็คืนค่า None เช่นกัน

คำสั่ง result.append(a): เรียกเมธอดของออบเจกต์ลิสต์ result เมธอดคือลักษณะของฟังก์ชันที่ "เป็นของ" ออบเจกต์หนึ่ง และจะถูกเขียนในรูปแบบ obj.methodname โดยที่ obj คือวัตถุ (อาจเป็นนิพจน์) และ methodname คือชื่อของเมธอดที่ถูกกำหนดโดยประเภทของออบเจกต์นั้น ๆ ประเภทต่าง ๆ อาจกำหนดเมธอดที่แตกต่างกัน และเมธอดของประเภทต่าง ๆ อาจมีชื่อเดียวกันได้ โดยไม่เกิดความสับสน เมธอด append() ที่แสดงในตัวอย่างนี้ ถูกกำหนดไว้สำหรับออบเจกต์ประเภทลิสต์ มันจะเพิ่มองค์ประกอบใหม่ที่ท้ายลิสต์ ในตัวอย่างนี้มันเทียบเท่ากับการใช้ result = result + [a] แต่มีประสิทธิภาพมากกว่า

## 2.3 ความแตกต่างระหว่าง Dashboard และ Report

### 2.3.1 ลักษณะการใช้งาน

2.3.1.1 Dashboard: มักใช้สำหรับการแสดงข้อมูลที่สำคัญในรูปแบบภาพ (visual) เช่น กราฟ แผนภูมิ หรือแผนที่ ซึ่งช่วยให้ผู้ใช้สามารถดูข้อมูลสรุปได้อย่างรวดเร็ว โดยมักจะมีการอัปเดตข้อมูลแบบเรียลไทม์หรือในช่วงเวลาที่สั้น เพื่อการติดตามผลหรือการตัดสินใจที่รวดเร็ว

2.3.1.2 Report: เอกสารหรือไฟล์ที่ใช้สำหรับการนำเสนอข้อมูลอย่างละเอียด โดยมีข้อมูลเชิงลึก รวมถึงการวิเคราะห์และคำอธิบายเพิ่มเติม มักจะมีการจัดทำและสรุปในช่วงเวลาหนึ่ง เช่น รายงานประจำเดือนหรือประจำปี



### 2.3.2 เนื้อหาของข้อมูล

2.3.2.1 Dashboard: ให้ข้อมูลที่สรุปและมีภาพรวม (summary) มักจะมีกราฟหรือแผนภูมิที่แสดงตัวชี้วัดสำคัญ (KPIs) เช่น ยอดขายทั้งหมด จำนวนลูกค้า หรือผลประกอบการ เพื่อให้ผู้ใช้งานสามารถเห็นภาพรวมในแง่มุมต่างๆ ได้ในเวลาเดียวกัน

2.3.2.2 Report: จะมีข้อมูลที่ละเอียดและเจาะลึก เช่น ตัวเลขที่เป็นผลรวม หรือการวิเคราะห์ทางสถิติ โดยมักจะมีคำอธิบายประกอบการวิเคราะห์เพื่อให้ผู้อ่านสามารถเข้าใจข้อมูลและตัดสินใจได้จากข้อมูลที่มีความละเอียด

### 2.3.3 ความยืดหยุ่นในการปรับเปลี่ยน

2.3.3.1 Dashboard: สามารถปรับแต่งและอัปเดตข้อมูลได้ตลอดเวลาในลักษณะที่ยืดหยุ่นและทันสมัย โดยเน้นการตอบสนองกับการเปลี่ยนแปลงข้อมูลแบบเรียลไทม์

2.3.3.2 Report: มักจะต้องมีการจัดเตรียมและจัดทำล่วงหน้า ซึ่งเมื่อข้อมูลถูกสร้างแล้วจะไม่สามารถปรับเปลี่ยนได้ง่ายนัก และมักจะถูกจัดทำเป็นเอกสารที่ส่งหรือพิมพ์ออกมา

### 2.3.4 กลุ่มผู้ใช้งาน

2.3.4.1 Dashboard: ใช้สำหรับผู้ที่ต้องการติดตามข้อมูลหรือผลลัพธ์ในเวลาใกล้เคียง เช่น ผู้บริหารหรือทีมงานที่ต้องการดูข้อมูลสรุปที่อัปเดตตลอดเวลา

2.3.4.2 Report: ใช้สำหรับผู้ที่ต้องการข้อมูลเชิงลึกหรือต้องการทำการวิเคราะห์และประเมินผลในระยะยาว เช่น นักวิเคราะห์ข้อมูลหรือผู้ที่ต้องการทำการนำเสนอข้อมูลอย่างละเอียด

### 2.3.5 ลักษณะการนำเสนอข้อมูล

2.3.5.1 Dashboard: เน้นการแสดงผลข้อมูลในรูปแบบที่กระชับและเข้าใจง่าย โดยใช้กราฟ แผนภูมิ และตัวชี้วัดที่สำคัญที่สามารถสังเกตได้ง่าย

2.3.5.2 Report: ใช้ข้อความเพื่ออธิบายข้อมูลในเชิงลึกและรายละเอียด และสามารถใส่กราฟหรือแผนภูมิประกอบการอธิบายได้ แต่ข้อมูลส่วนใหญ่จะเน้นที่การแปลความหมายของตัวเลขและการวิเคราะห์

โดยสรุป Dashboard เป็นเครื่องมือที่ช่วยให้ผู้ใช้งานสามารถติดตามและเห็นภาพรวมของข้อมูลได้ทันทีในรูปแบบที่เข้าใจง่าย ส่วน Report เป็นเอกสารที่ใช้ในการวิเคราะห์และนำเสนอข้อมูลอย่างละเอียด พร้อมคำอธิบายเพื่อสนับสนุนการตัดสินใจ

## 2.4 ธุรกิจ E-Commerce

ธุรกิจ E-Commerce (อีคอมเมิร์ซ) คือ การทำธุรกรรมการซื้อขายสินค้าและบริการผ่านระบบอินเทอร์เน็ต ซึ่งไม่จำกัดแค่การซื้อขายสินค้า แต่รวมถึงการทำกิจกรรมทางธุรกิจต่างๆ เช่น การชำระเงิน การจัดส่งสินค้า การจัดการกับลูกค้า และการตลาดออนไลน์

#### 2.4.1 ประเภทของธุรกิจ E-Commerce

2.4.1.1 B2C (Business to Consumer) คือ การขายสินค้าหรือบริการจากธุรกิจ (ผู้ขาย) ไปยังลูกค้า (ผู้ซื้อ) ตัวอย่างเช่น เว็บไซต์ขายสินค้าออนไลน์อย่าง Amazon, Lazada, Shopee ที่ลูกค้าสามารถเลือกซื้อสินค้าที่ต้องการและชำระเงินออนไลน์

2.4.1.2 B2B (Business to Business) คือ การทำธุรกรรมระหว่างธุรกิจสองแห่ง ซึ่งอาจเกี่ยวข้องกับการซื้อขายวัตถุดิบหรือสินค้า ขายส่ง ตัวอย่างเช่น Alibaba ที่ทำธุรกรรมระหว่างผู้ผลิตและผู้ค้าส่ง

2.4.1.3 C2C (Consumer to Consumer) คือ การทำธุรกรรมระหว่างผู้บริโภค (ลูกค้า) ด้วยกัน เช่น การซื้อขายสินค้ามือสองผ่านแพลตฟอร์มออนไลน์อย่าง eBay หรือ Facebook Marketplace

2.4.1.4 C2B (Consumer to Business) คือ การที่ผู้บริโภคขายสินค้าหรือบริการให้กับธุรกิจ เช่น การทำการตลาดออนไลน์หรือการขายภาพถ่ายให้กับบริษัทที่ต้องการ

#### 2.4.2 ข้อดีของธุรกิจ E-Commerce

2.4.2.1 สะดวกและรวดเร็ว: ลูกค้าสามารถเลือกซื้อสินค้าหรือบริการได้ทุกที่ทุกเวลา ไม่จำเป็นต้องเดินทางไปยังร้านค้า

2.4.2.2 การเข้าถึงตลาดที่กว้างขึ้น: ธุรกิจสามารถเข้าถึงลูกค้าทั่วโลกผ่านอินเทอร์เน็ต

2.4.2.3 ลดต้นทุนการดำเนินธุรกิจ: ธุรกิจสามารถลดค่าใช้จ่ายจากการเปิดหน้าร้าน หรือค่าใช้จ่ายทางกายภาพอื่นๆ

2.4.2.4 ข้อมูลและการวิเคราะห์: ธุรกิจสามารถใช้ข้อมูลจากการซื้อขายและพฤติกรรมของลูกค้าเพื่อปรับกลยุทธ์การตลาด

#### 2.4.3 ข้อเสียของธุรกิจ E-Commerce

2.4.3.1 ความน่าเชื่อถือ: บางครั้งลูกค้าอาจรู้สึกไม่มั่นใจในการซื้อสินค้าผ่านทางออนไลน์ โดยเฉพาะสินค้าที่มีราคาสูงหรือมีความเสี่ยง

2.4.3.2 การแข่งขันสูง: มีคู่แข่งมากมายในตลาดออนไลน์ และการจะโดดเด่นในตลาดต้องใช้กลยุทธ์การตลาดที่ดี

2.4.3.3 ปัญหาการจัดส่งสินค้า: การจัดส่งสินค้ามีความสำคัญในการทำธุรกิจ E-Commerce และหากมีปัญหาเกี่ยวกับการจัดส่งอาจทำให้ลูกค้าผิดหวัง

2.4.3.4 ข้อจำกัดด้านเทคโนโลยี: บางกลุ่มลูกค้าอาจไม่มีการเข้าถึงอินเทอร์เน็ตหรือไม่สะดวกในการใช้เทคโนโลยี ทำให้ไม่สามารถใช้บริการได้

#### 2.4.4 เทรนด์และการพัฒนาในธุรกิจ E-Commerce

2.4.4.1 การใช้ AI และ Machine Learning: เพื่อพัฒนาการแนะนำสินค้าและบริการที่เหมาะสมกับลูกค้า

2.4.4.2 การชำระเงินผ่านมือถือ (Mobile Payments): การใช้แอปพลิเคชันต่างๆ เช่น PayPal, Line Pay หรือการชำระเงินผ่านธนาคาร

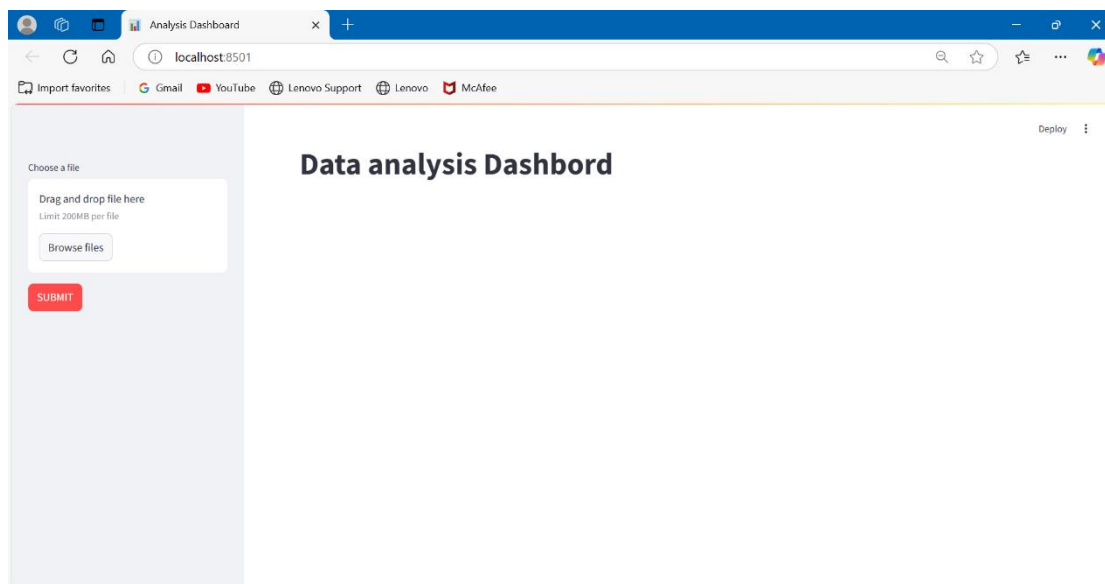
2.4.4.3 การใช้เทคโนโลยี Augmented Reality (AR): ที่ช่วยให้ลูกค้าสามารถทดลองสินค้าผ่านออนไลน์ได้

2.4.4.4 การส่งสินค้าผ่านระบบอัตโนมัติ: เช่น การใช้หุ่นยนต์หรือโดรนในการจัดส่งสินค้า

## บทที่ 3

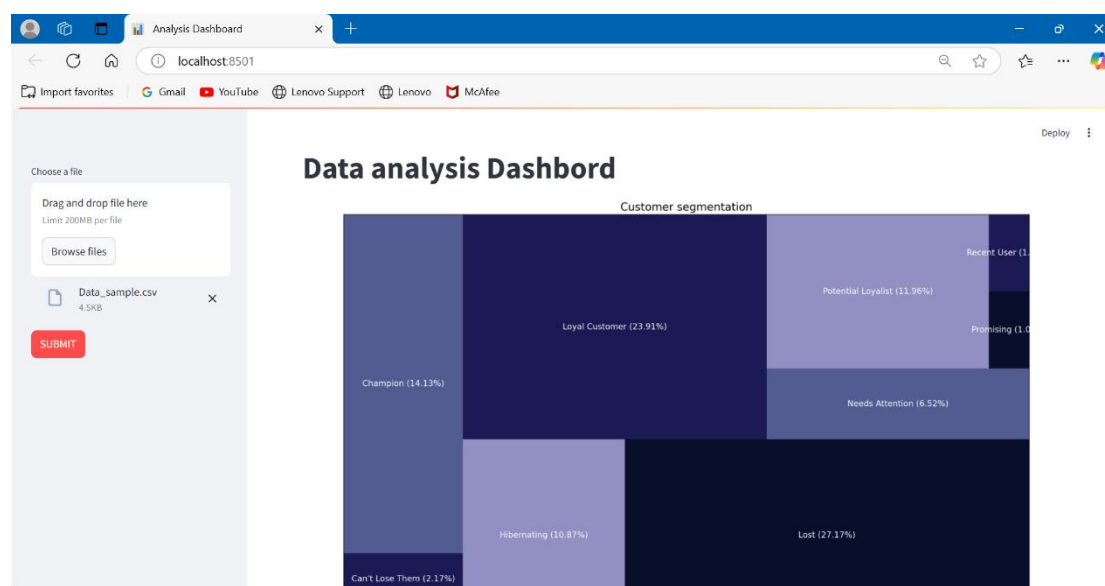
### User Interface

#### 3.1 หน้าแรก



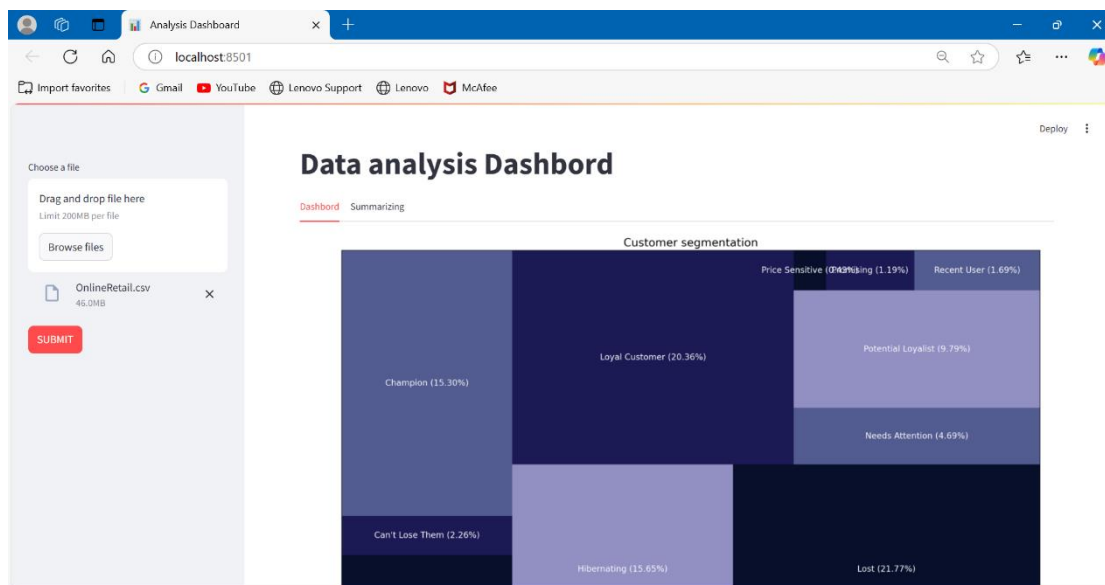
รูปที่ 2: หน้าแรก

#### 3.2 ผลลัพธ์หลังจากอัปโหลดไฟล์ Data\_sample



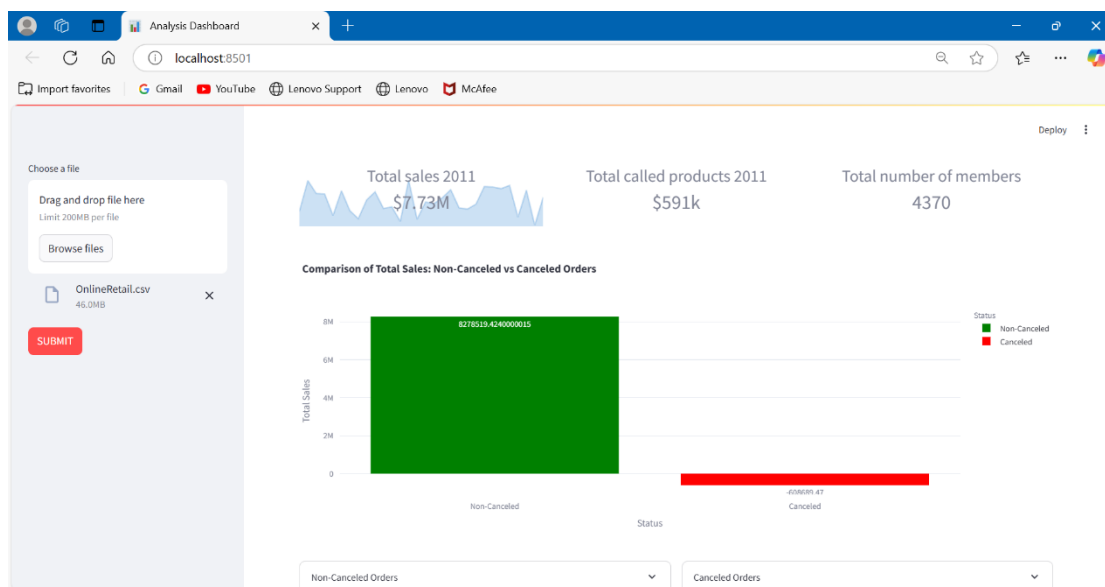
รูปที่ 3: ผลลัพธ์หลังจากอัปโหลดไฟล์ Data\_sample

### 3.3 ผลลัพธ์หลังจากอัปโหลดไฟล์ OnlineRetail

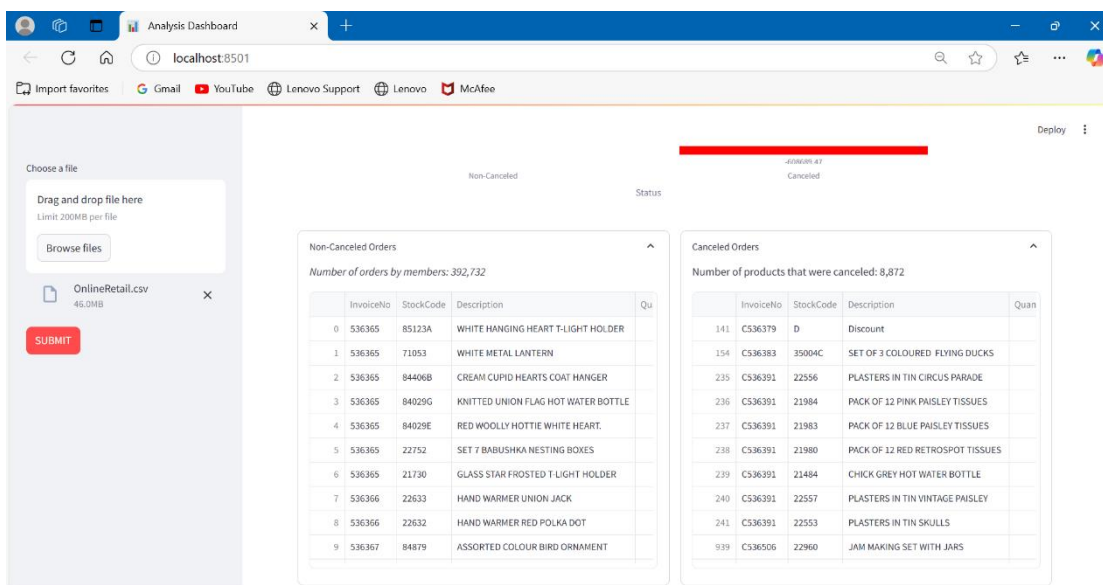


3.3.1 ผลลัพธ์ในส่วนของแท็บ Dashboard

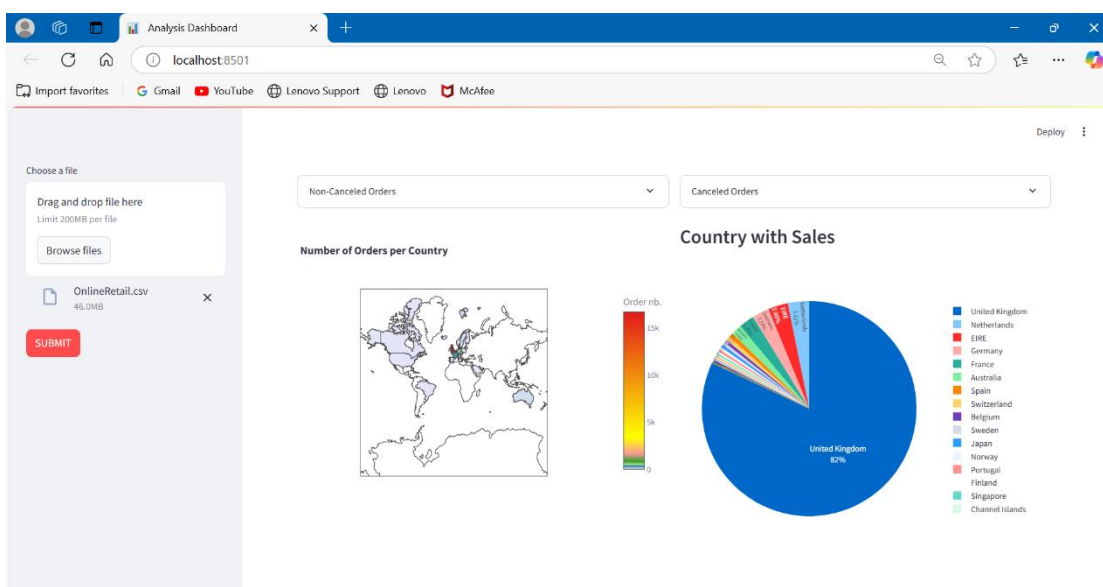
รูปที่ 4: ผลลัพธ์ในส่วนของแท็บ Dashboard (1)



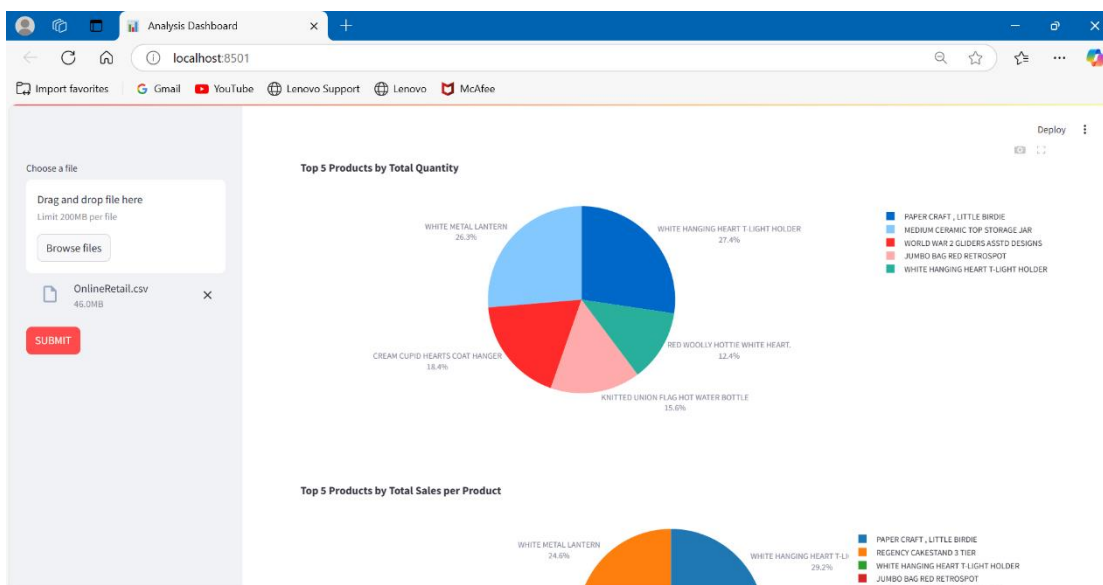
รูปที่ 5: ผลลัพธ์ในส่วนของแท็บ Dashboard (2)



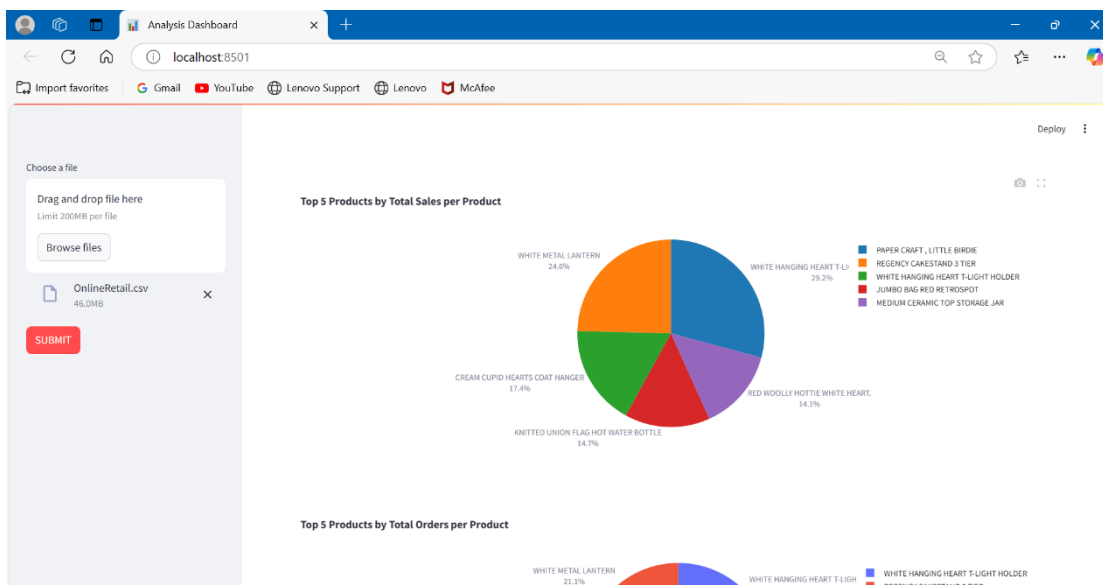
รูปที่ 6: ผลลัพธ์ในส่วนของแท็บ Dashboard (3)



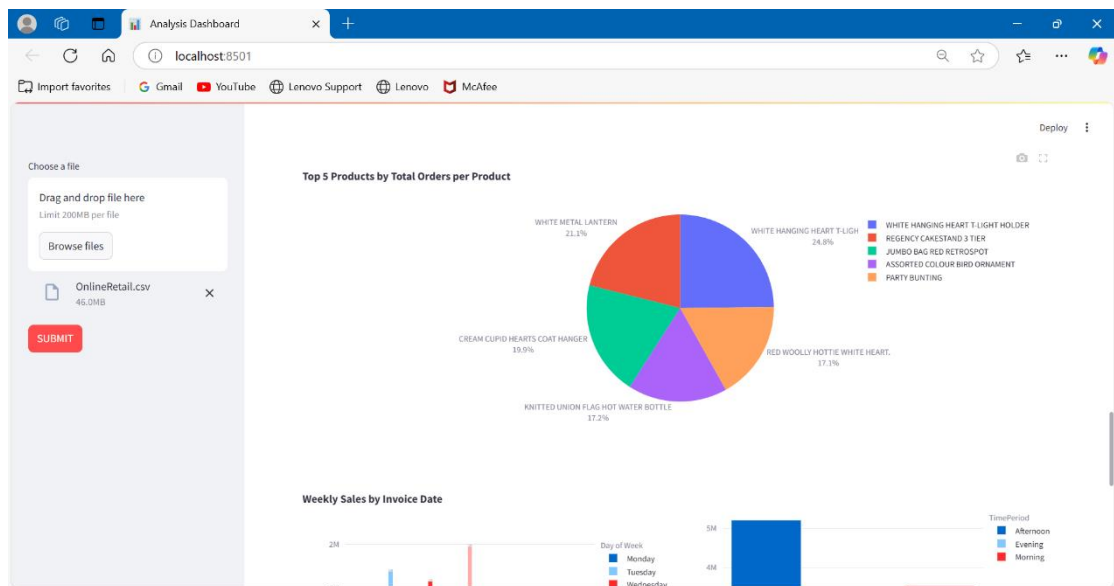
รูปที่ 7: ผลลัพธ์ในส่วนของแท็บ Dashboard (4)



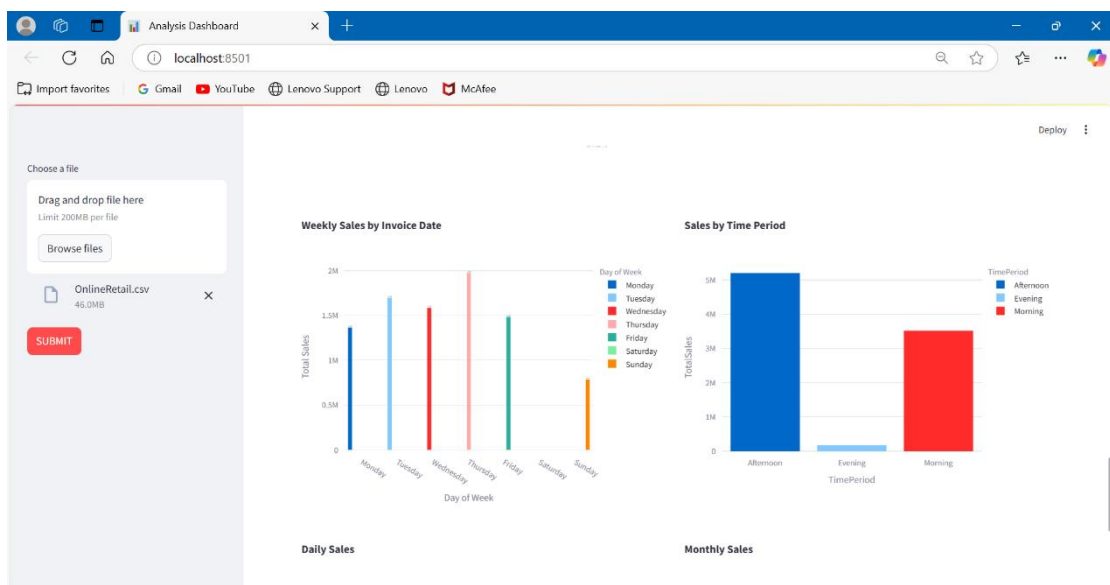
รูปที่ 8: ผลลัพธ์ในส่วนของแท็บ Dashboard (5)



รูปที่ 9: ผลลัพธ์ในส่วนของแท็บ Dashboard (6)

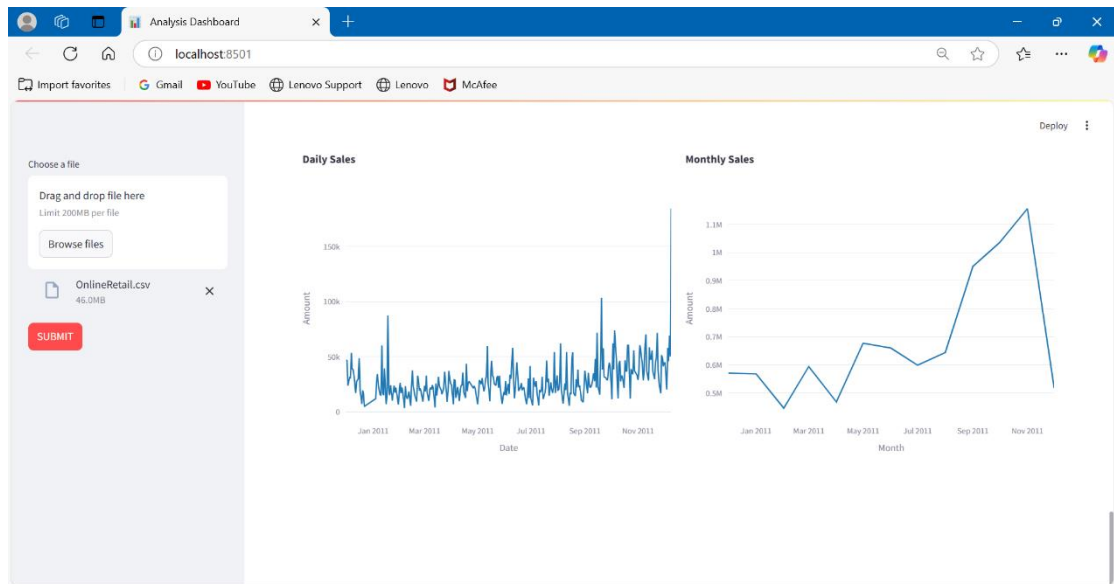


รูปที่ 10: ผลลัพธ์ในส่วนของแท็บ Dashboard (7)



รูปที่ 11: ผลลัพธ์ในส่วนของแท็บ Dashboard (8)





รูปที่ 12: ผลลัพธ์ในส่วนของแท็บ Dashboard (9)

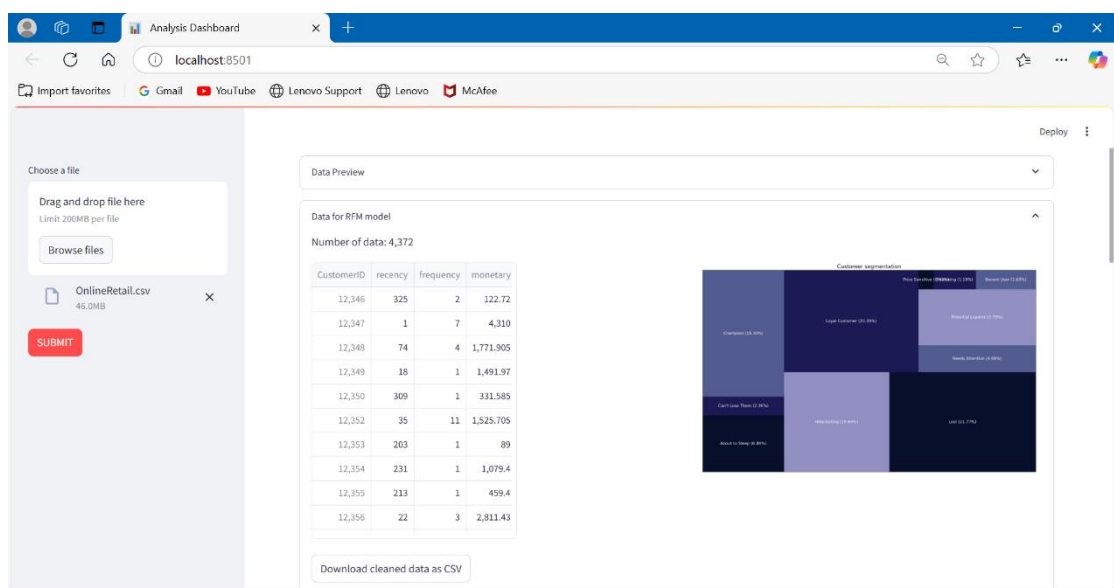
### 3.3.2 ผลลัพธ์ในส่วนของแท็บ Summarizing

The screenshot shows the "Data analysis Dashboard" with the "Summarizing" tab selected. It features a "Data Preview" dropdown and a "Data for RFM model" dropdown. Below these, a table displays the data for the RFM model, including segments and their associated metrics.

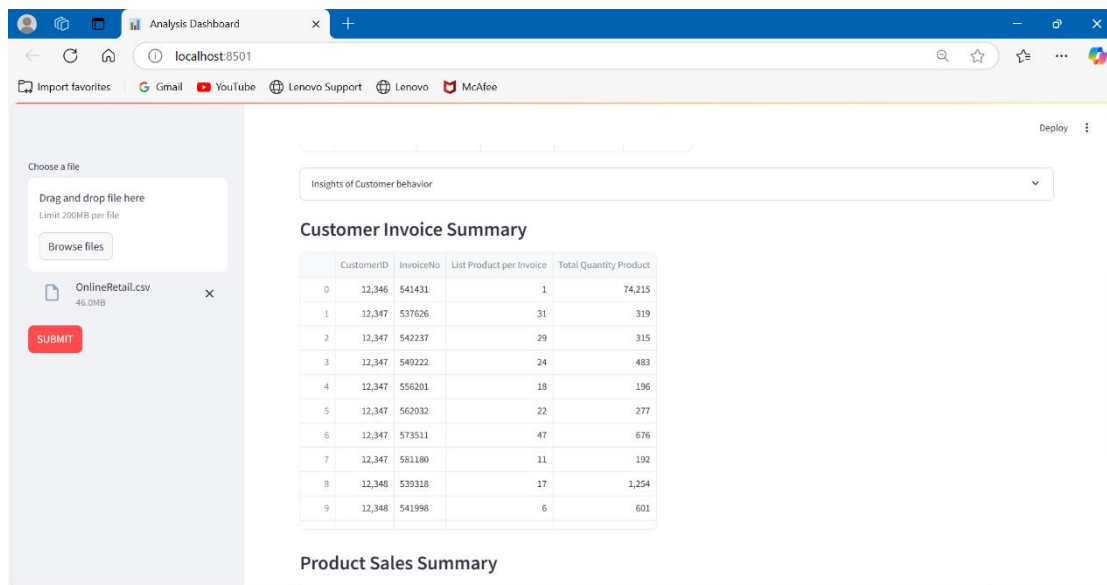
	Segment	Recency_Avg	Frequency_Avg	Monetary_Avg	Segment_Size
0	About to Sleep	51.3133	1.2533	265.6214	300
1	Can't Lose Them	122.7475	8	2,951.5169	99
2	Champion	4.3797	14.296	6,203.552	669
3	Hibernating	158.5965	2.9766	856.9848	684
4	Lost	222.7437	1.1996	235.0736	952
5	Loyal Customer	31.3876	7.5225	2,603.7247	890
6	Needs Attention	49.6049	2.3659	657.0501	205
7	Potential Loyalist	17.0047	2.1308	481.1204	428
8	Price Sensitive	5.6316	1	167.6803	19
9	Promising	22.1154	1	148.1627	52

รูปที่ 13: ผลลัพธ์ในส่วนของแท็บ Summarizing (1)

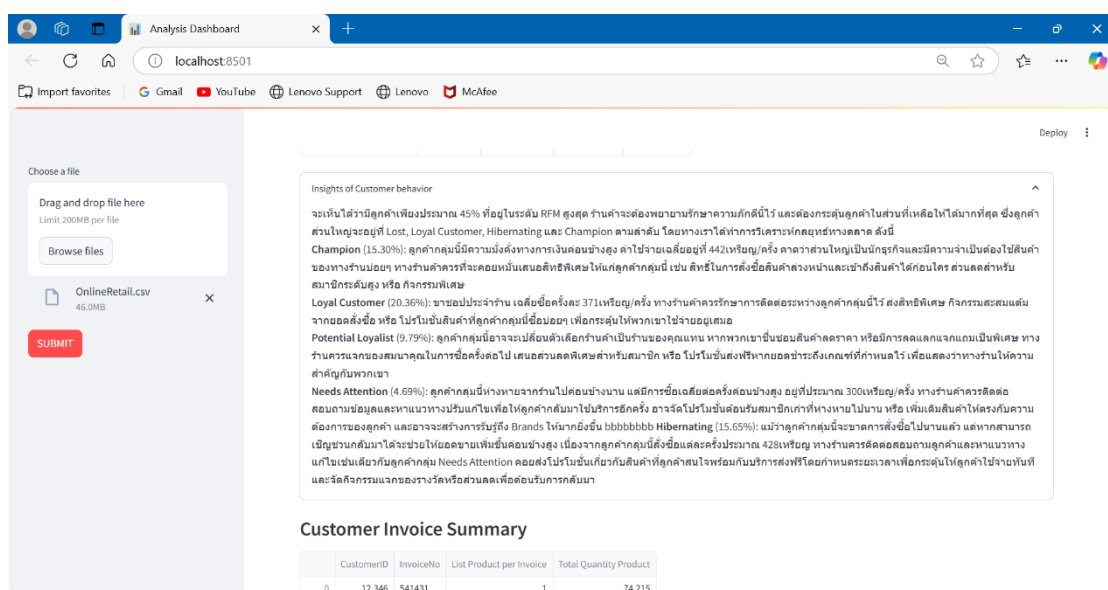
รูปที่ 14: ผลลัพธ์ในส่วนของแท็บ Summarizing (2)



รูปที่ 15: ผลลัพธ์ในส่วนของแท็บ Summarizing (3)



รูปที่ 16: ผลลัพธ์ในส่วนของแท็บ Summarizing (4)



รูปที่ 17: ผลลัพธ์ในส่วนของแท็บ Summarizing (5)

Choose a file

Drag and drop file here  
Limit 200MB per file

Browse files

OnlineRetail.csv  
46.0MB

SUBMIT

### Product Sales Summary

	Description	Total Quantity	TotalSales per Product	Total orders per product
0	INFLATABLE POLITICAL GLOBE	823	699.55	49
1	GROOVY CACTUS INFLATABLE	291	114.41	21
2	DOGGY RUBBER	192	40.32	29
3	HEARTS WRAPPING TAPE	5	3.25	3
4	SPOTS ON RED BOOKCOVER TAPE	16	6.72	5
5	ARMY CAMO BOOKCOVER TAPE	17	7.14	4
6	MINI FUNKY DESIGN TAPES	1,225	929.45	63
7	COLOURING PENCILS BROWN TUBE	2,373	1,138.99	122
8	COLOURING PENCILS BROWN TUBE	1,936	1,784.19	119
9	ASSTD DESIGN RACING CAR PEN	1,252	1,953	64

Deploy

รูปที่ 18: ผลลัพธ์ในส่วนของแท็บ Summarizing (6)

## บทที่ 4

### Coding

ตัวอย่างโค้ด:

```
import streamlit as st
import pandas as pd
import numpy as np
import random
from io import StringIO
import squarify
import plotly.graph_objs as go
import matplotlib.pyplot as plt
import plotly.express as px
```

ส่วนที่ 1 : นำเข้าไลบรารีที่จำเป็น

```
# Cleansing Data #
def CleansingData(uploaded_file):
    if uploaded_file.name == 'OnlineRetail.csv':
        ...
        Upload data
        ...
        for col in ['Quantity', 'UnitPrice']:
            ...
            Loop ลดระยะห่างของข้อมูล
            ...
        return Data_clean
    else:
        df = load_data(uploaded_file)
        return df
```

ส่วนที่ 2 : Cleansing Data

```

### Definition Convert RFM Score to segment label ###
def segment_label(RFMScore):
    if RFMScore in ['54', '55']:
        return "Champion"
    elif RFMScore == '52':
        return "Recent User"
    elif RFMScore == '51':
        return "Price Sensitive"
    elif RFMScore in ['42', '43', '52', '53']:
        return "Potential Loyalist"
    elif RFMScore == '41':
        return "Promising"
    elif RFMScore in ['34', '35', '44', '45']:
        return "Loyal Customer"
    elif RFMScore == '33':
        return "Needs Attention"
    elif RFMScore in ['31', '32']:
        return "About to Sleep"
    elif RFMScore in ['15', '25']:
        return "Can't Lose Them"
    elif RFMScore in ['13', '14', '23', '24']:
        return "Hibernating"
    elif RFMScore in ['11', '12', '21', '22']:
        return "Lost"
    else:
        return "Don't have segment label"
# End def #

```

ส่วนที่ 3 : if-else แบ่งช่วง RFM segment

```

#### Main layout ####
if submit:
    if uploaded_file is None:
        st.info("Upload a file through config")
        st.stop()

    if uploaded_file is not None:
        # To read file as bytes:
        bytes_data = uploaded_file.getvalue()

        # To convert to a string based IO:
        stringio = StringIO(uploaded_file.getvalue().decode("utf-8"))

        df = load_data(uploaded_file)
        RFMmodel(df)

```

ส่วนที่ 4 : ตรวจสอบการ submit เพื่อเริ่ม RFM Model

```

with b2:
    # Create a function to find a time period
    def time_of_day(hour):
        if 6 <= hour < 12:
            return 'Morning'
        elif 12 <= hour < 18:
            return 'Afternoon'
        elif 18 <= hour < 24:
            return 'Evening'
        else:
            return 'Night'

```

ส่วนที่ 5 : สร้างฟังก์ชันเพื่อค้นหาช่วงเวลา

## บทที่ 5

### อภิปรายผลการดำเนินงาน ข้อเสนอแนะ และแนวทางแก้ไข

จากการศึกษา Python เพื่อนำมาสร้างโมเดล RFM (Recency, Frequency, Monetary) ซึ่งเป็นเทคนิคการวิเคราะห์ข้อมูลลูกค้า และการสร้าง Dashboard เพื่อนำเสนอผลการดำเนินงานของธุรกิจ โดยใช้ข้อมูล E-Commerce บน [www.kaggle.com](https://www.kaggle.com) สามารถสรุปผลการดำเนินงาน ข้อเสนอแนะ ปัญหาที่พบและแนวทางการแก้ไข ได้ดังนี้

#### 5.1 สรุปผลการดำเนินงาน

การจัดทำโครงการนี้บรรลุวัตถุประสงค์ในการศึกษาเพิ่มเติมเกี่ยวกับภาษา Python และนำความรู้ที่ได้เรียนมาประยุกต์ใช้ผ่านการวิเคราะห์และจัดกลุ่มลูกค้าตามพฤติกรรมการซื้อขายด้วยโมเดล RFM ซึ่งผลลัพธ์ที่ได้ถูกนำมาสร้างกราฟและแดชบอร์ดเพื่อที่จะแสดงผลข้อมูลที่สามารถทำความเข้าใจได้ง่าย

โดยโครงการนี้แสดงข้อมูลบน Streamlit ซึ่งเป็น open-source app framework โดยมีการใช้ Libraries ดังนี้ streamlit, pandas, numpy, random, StringIO, squarify, plotly และ matplotlib โดยใช้ StringIO ในการเตรียมไฟล์ให้พร้อมใช้งาน ใช้ pandas ในการจัดการข้อมูลในรูปแบบตาราง (Data Frame) ใช้ numpy ในการจัดการข้อมูลเชิงตัวเลขและคำนวณทางคณิตศาสตร์ ในส่วนของที่เหลือจะใช้ในการสร้างและปรับแต่งกราฟ

ในส่วนของการ Coding ได้นำความรู้ที่ได้รับในระหว่างเรียนช่วงกลางภาคมาใช้ในการเขียนโปรแกรม ดังนี้ if-elseif statement, for loop, max-min function และการจัด format นอกจากนี้ยังมีการสร้างฟังก์ชันต่างๆ ได้แก่ load\_data(file) เพื่ออัปโหลดข้อมูล, CleansingData(uploaded\_file) เพื่อเตรียมข้อมูลให้พร้อมในการทำโมเดล RFM, segment\_label(RFMScore) เพื่อจัดลำดับลูกค้ารายบุคคลว่าควรอยู่ในกลุ่มใด, RFMmodel(df) เพื่อวิเคราะห์พฤติกรรมการซื้อของลูกค้าและจัดกลุ่มลูกค้าให้เหมาะสม แสดงโมเดลในลักษณะ treemap, time\_of\_day(hour) เพื่อคำนวณช่วงเวลาตอนเช้า บ่าย เย็น หรือกลางคืน

ผลลัพธ์ได้จากการทำโมเดล RFM ถูกนำมาวิเคราะห์เพื่อหาข้อมูลเชิงลึกและกลยุทธ์ทางการตลาด โดยไฟล์ OnlineRetail ได้ผลลัพธ์ออกมาคือ มีลูกค้าเพียงประมาณ 45% ที่อยู่ในระดับ RFM สูงสุด ซึ่งลูกค้าส่วนใหญ่จะอยู่ที่ Lost, Loyal Customer, Hibernating และ Champion ตามลำดับ ในการนำเสนอกลยุทธ์จะนำเสนอเพียงกลุ่มลูกค้าที่คุ้มค่าต่อการรักษาความสัมพันธ์ไว้ ซึ่งได้แก่ Champion (15.30%), Loyal Customer (20.36%), Potential Loyalist (9.79%), Need Attention (4.69%), Hibernating (15.65%) นอกจากนี้ยังมีการนำข้อมูลในไฟล์มาสรุปยอดขายของปีล่าสุด จำนวนเงินของการยกเลิกสินค้าปีล่าสุด อัปเดตจำนวนสมาชิกทั้งหมด สร้างกราฟ



ยอดขายรายวัน รายสัปดาห์ และรายเดือน กราฟสินค้า 5 อันดับแรกโดยจำแนกยอดรวมต่างๆ เป็นต้น

ในส่วนที่ทางคณะผู้จัดอยากพัฒนาเพิ่มเติมคือ การทำ Live Dashboard เพื่อที่จะให้ผู้ใช้งานสามารถตรวจสอบประสิทธิภาพในการขายของร้านค้าได้ รวมถึงการหาสินค้ายอดนิยมและดูข้อมูลของแต่ละประเทศที่มีการสั่งซื้อ

## 5.2 ข้อเสนอแนะ

5.2.1 การวิเคราะห์ข้อมูลที่ดี ผู้ที่รับผิดชอบในหน้าที่นี้ควรศึกษาสิ่งที่เกี่ยวข้องกับข้อมูลให้เข้าใจอย่างถ่องแท้ เนื่องจากข้อมูลที่ได้จากการวิเคราะห์จะถูกนำไปใช้งานต่อ หากเกิดความผิดพลาดในวิเคราะห์ข้อมูลอาจจะสร้างความเสียหายให้แก่องค์กรหรือหน่วยงานได้

5.2.2 การศึกษาผลงานที่เกี่ยวข้องกับเรื่องที่ทำและทำความเข้าใจอย่างลึกซึ้ง จะช่วยให้เราเข้าใจตรรกะ กระบวนการ และมองเห็นส่วนสำคัญได้ดียิ่งขึ้น ในอนาคต สิ่งนี้อาจจะช่วยให้เรากลายเป็นนักพัฒนาในเรื่องที่ทำก็ได้

## 5.3 ปัญหาที่พบ และแนวทางการแก้ไขปัญหา

5.3.1 การนำความรู้ที่ได้รับจากการเรียนวิชา Python Programming ในส่วนของกลางภาค ถูกนำมาใช้ในโครงงานนี้ค่อนข้างน้อย และการออกแบบโปรแกรมยังมีประสิทธิภาพไม่เพียงพอ ซึ่งในครั้งต่อไปทางคณะผู้จัดจะตรวจสอบการดำเนินงานให้รอบคอบยิ่งขึ้นเพื่อไม่ให้เกิดเหตุการณ์เช่นนี้อีก

5.3.2 บางฟังก์ชันของ Python ไม่สามารถใช้งานร่วมกับ Streamlit ได้ ทำการแก้ไขโดยศึกษาค้นคว้าเพิ่มเติมเพื่อหาฟังก์ชันหรือวิธีอื่นมาใช้งานแทน

5.3.3 การเกิดข้อผิดพลาดในระหว่างการรวม Code อันเนื่องมาจากความแตกต่างของการจัดวางโค้ด ตัวแปรที่ใช้ และเงื่อนไขของโค้ดในแต่ละส่วน นำมาซึ่งปัญหา เช่น ไม่ค้นพบข้อมูลที่เรียกใช้ ไม่แสดงผลลัพธ์จากคำสั่งเรียกใช้ ระยะเวลาในการแก้ไขที่ค่อนข้างนาน เป็นต้น ทางคณะผู้จัดทำได้เห็นพ้องต้องกันว่าควรกำหนดรายละเอียดให้ชัดเจนและหมั่นตรวจสอบโปรแกรมซึ่งกันและกัน

## บรรณานุกรม

### 1. สื่ออิเล็กทรอนิกส์

Doctor ease. “RFM Model คืออะไร ? เทคนิคการวิเคราะห์และแบ่งกลุ่มลูกค้าด้วย RFM Analysis”. [Online]. Available : <https://doctorease.co/what-is-rfm/>

Python. 2025. “Python 3.13.1 documentation”. [Online].  
<https://docs.python.org/3/>

Excel Millennium. 2016. "Report VS Dashboard มันต่างกันยังไงล่ะ". [Online].  
[https://excelmillenium.blogspot.com/2016/03/report-vs-dashboard.html?utm\\_source=chatgpt.com](https://excelmillenium.blogspot.com/2016/03/report-vs-dashboard.html?utm_source=chatgpt.com)

Foxbith. 2024. "E-commerce | อีคอมเมิร์ซ คืออะไร ? มีประเภทอะไรบ้าง". [Online].  
[https://www.foxbith.com/blog/what-is-ecommerce?utm\\_source=chatgpt.com](https://www.foxbith.com/blog/what-is-ecommerce?utm_source=chatgpt.com)

NIPA. "ธุรกิจ อีคอมเมิร์ซ คืออะไร? ทำไมถึงดึงดูดหลายคน เริ่มต้นธุรกิจนี้" [Online].  
[https://nipa.co.th/th/article/digital-marketing/what-e-commerce-business-model-is?utm\\_source=chatgpt.com](https://nipa.co.th/th/article/digital-marketing/what-e-commerce-business-model-is?utm_source=chatgpt.com)

### 2. Research

Jo-Ting Wei , Shih-Yen , Lin,Hsin-Hung Wu. 2010.

“A review of the application of RFM model”.

[https://www.researchgate.net/publication/228399859\\_A\\_review\\_of\\_the\\_application\\_of\\_RFM\\_model](https://www.researchgate.net/publication/228399859_A_review_of_the_application_of_RFM_model).