



รายงาน

เรื่อง ระบบสนับสนุนการตัดสินใจโดยใช้วิธีเมทริกซ์ตัดสินใจ
(Decision Helping System by Using Decision Matrix Method)

จัดทำโดย

นางสาวจิรนนท์	วรรณสาร	65050123
นางสาวปัญยวีร์	กิตติกุลยุทธ์	65050566
นายรศิราภ์	รวิรัฐธา	65050738
นางสาวสิรพัชร	หนูไธ	65050907

นำเสนอ

รศ.ดร.ละออ บุญเกษม

รายงานเล่มนี้เป็นส่วนหนึ่งของวิชา 05016385 PYTHON PROGRAMMING

สาขาคณิตศาสตร์ประยุกต์ ภาควิชาคณิตศาสตร์ คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ภาคเรียนที่ 2 ปีการศึกษา 2567

คำนำ

โครงการนี้เป็นส่วนหนึ่งของรายวิชา 05016385 การเขียนโปรแกรมด้วยภาษาไพทอน (PYTHON PROGRAMMING) ซึ่งมุ่งเน้นที่การศึกษาและออกแบบระบบที่ช่วยสนับสนุนการตัดสินใจ โดยใช้ความรู้ที่ได้รับจากรายวิชานี้เป็นพื้นฐานในการวิเคราะห์และพัฒนาระบบที่มีประสิทธิภาพและตอบสนองความต้องการของผู้ใช้งาน

ในการดำเนินงานของโครงการนี้ คณะผู้จัดทำได้รวบรวมข้อมูลจากเอกสารและทฤษฎีที่เกี่ยวข้อง เช่น ทฤษฎีเมทริกซ์การตัดสินใจ (Decision Matrix) และการวิเคราะห์สภาพแวดล้อมและศักยภาพ (SWOT Analysis) รวมถึงงานวิจัยที่เกี่ยวข้องกับการตัดสินใจ ข้อมูลเหล่านี้ถูกนำมาประยุกต์ใช้ร่วมกับความรู้ด้านการเขียนโปรแกรมภาษาไพทอน และไลบรารี Streamlit เพื่อพัฒนาโปรแกรมที่สามารถช่วยผู้ใช้งานเปรียบเทียบตัวเลือกต่าง ๆ อย่างมีเหตุผล และเสริมสร้างทักษะการวิเคราะห์ปัจจัยที่เกี่ยวข้องกับการตัดสินใจอย่างมีประสิทธิภาพ

คณะผู้จัดทำหวังเป็นอย่างยิ่งว่าโครงการนี้จะเป็นประโยชน์ต่อผู้อ่านทุกท่าน และหากมีข้อผิดพลาดประการใด คณะผู้จัดทำขอน้อมรับและขออภัยมา ณ ที่นี้

คณะผู้จัดทำ

สารบัญ

เรื่อง	หน้า
คำนำ	ก
สารบัญ	ข
สารบัญรูปภาพ	ง
สารบัญตาราง	จ
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญ	1
1.2 วัตถุประสงค์ของการศึกษา	1
1.3 ขอบเขตการศึกษา	2
1.4 ผลที่คาดว่าจะได้รับ	2
1.5 แผนการดำเนินการ	3
บทที่ 2 ทฤษฎีพื้นฐานและงานที่เกี่ยวข้อง	4
2.1 ข้อมูลเกี่ยวกับธุรกิจต้นแบบ	4
2.2 ทฤษฎีที่เกี่ยวข้อง	4
2.3 ภาษาไพทอน (Python)	16
2.4 โปรแกรม Visual Studio Code	16
2.5 ภาษาตกแต่งเว็บไซต์ CSS (Cascading Style Sheets: CSS)	18
2.6 ไลบรารี Streamlit	20
2.7 ไลบรารี Pandas	21
2.8 ไลบรารี Base64	23
2.9 ไลบรารี Matplotlib	24
2.10 ไลบรารี Seaborn	27
บทที่ 3 วิธีการดำเนินการ	29
3.1 การออกแบบและวางแผนโครงสร้างโปรแกรม	29
3.2 User Interface	30

เรื่อง	หน้า
บทที่ 4 การพัฒนาโปรแกรม	37
4.1 ชุดคำสั่งโปรแกรม	37
บทที่ 5 สรุปผล	46
5.1 สรุปผล	46
5.2 อภิปรายผล	46
5.3 ข้อเสนอแนะ	47
บรรณานุกรม	48

สารบัญรูปภาพ

รูปภาพที่	หน้า
รูปภาพที่ 2.1 แอปพลิเคชัน Decision Mentor	4
รูปภาพที่ 2.2 เมทริกซ์การตัดสินใจ (Decision Matrix)	5
รูปภาพที่ 2.3 การวิเคราะห์สภาพแวดล้อมและศักยภาพ (SWOT Analysis)	7
รูปภาพที่ 2.4 ภาษาไพทอน (Python)	16
รูปภาพที่ 2.5 แสดงรูปโปรแกรม Visual Studio Code	17
รูปภาพที่ 2.6 ภาษาตกแต่งเว็บไซต์ CSS (Cascading Style Sheets: CSS)	19
รูปภาพที่ 2.7 ไลบรารี Streamlit	21
รูปภาพที่ 2.8 ไลบรารี Pandas	23
รูปภาพที่ 2.9 การเข้ารหัสข้อมูล Base64	24
รูปภาพที่ 2.10 ไลบรารี Matplotlib	27
รูปภาพที่ 2.11 ไลบรารี Seaborn	28
รูปภาพที่ 3.1 แผนภาพการทำงานภาพรวมของทั้งระบบ	29
รูปภาพที่ 3.2 หน้าจอเริ่มต้น	30
รูปภาพที่ 3.3 หน้าจอการเลือกหมวดหมู่	31
รูปภาพที่ 3.4 หน้าจอการเพิ่มตัวเลือกที่จะพิจารณา	32
รูปภาพที่ 3.5 หน้าจอการเพิ่มเกณฑ์ที่ใช้ในการพิจารณาตัวเลือก	33
รูปภาพที่ 3.6 หน้าจอการกำหนดน้ำหนักและคะแนน	34
รูปภาพที่ 3.7 หน้าจอการกำหนดน้ำหนักและคะแนน (2)	35
รูปภาพที่ 3.8 หน้าจอแสดงผลลัพธ์	36

สารบัญตาราง

ตารางที่

หน้า

ตารางที่ 1.1 แผนการดำเนินการ

3

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญ

“เย็นนี้กินอะไรดี” คือคำถามที่มักจะได้รับคำตอบว่า “อะไรก็ได้” อยู่บ่อยครั้ง และตัวอย่างสถานการณ์ดังกล่าวนำไปสู่การตัดสินใจเรื่องปกติในชีวิตประจำวันที่อาจใช้เวลาอันยาวนานกว่าจะได้ข้อสรุป หากมองถึงประเด็นนี้ในภาพรวมแล้วนั้น จะพบว่าการตัดสินใจคือกระบวนการเลือกตัวเลือกที่ส่งผลต่อความสำเร็จหรือล้มเหลวของบุคคลหรือองค์กรในสถานการณ์ต่าง ๆ ไม่ว่าจะเป็นการเลือกซื้อสินค้า การบริหารจัดการโครงการ หรือการกำหนดกลยุทธ์ทางธุรกิจ

ในสังคมที่ขับเคลื่อนไปอย่างความเร่งรีบ กระบวนการตัดสินใจต้องเผชิญกับความซับซ้อนที่เพิ่มขึ้นจากปัจจัยต่าง ๆ เช่น จำนวนทางเลือกที่มากขึ้น ข้อมูลที่ต้องพิจารณาอย่างละเอียด ข้อจำกัดด้านเวลา ความรู้ที่จำกัด และความไม่แน่นอนของข้อมูล ปัจจัยเหล่านี้อาจส่งผลให้การตัดสินใจถูกชักนำด้วยอารมณ์ความรู้สึกมากกว่าข้อเท็จจริง ซึ่งอาจทำให้การตัดสินใจขาดความเป็นกลางและเหตุผลที่เหมาะสม

เพื่อแก้ไขปัญหาดังกล่าวและเพิ่มประสิทธิภาพในการตัดสินใจ เครื่องมือสนับสนุนการตัดสินใจจึงได้รับการพัฒนาขึ้น หนึ่งในวิธีการที่ได้รับการยอมรับอย่างกว้างขวางคือ เมทริกซ์การตัดสินใจ (Decision Matrix) ซึ่งเป็นเครื่องมือที่ช่วยจัดการและเปรียบเทียบตัวเลือกต่าง ๆ อย่างเป็นระบบ เมทริกซ์การตัดสินใจ (Decision Matrix) สามารถนำมาใช้ในสถานการณ์ที่มีตัวเลือกเพียงสองตัว หรือในกรณีที่มีตัวเลือกจำนวนมากก็สามารถจัดการได้อย่างมีประสิทธิภาพ

จากปัญหาที่พบและแนวทางการแก้ไขดังกล่าว 'ลั้งเล' จึงได้รับการพัฒนาขึ้น โดยมีวัตถุประสงค์เพื่อสนับสนุนผู้ใช้งานที่ต้องเผชิญกับความลังเลใจระหว่างตัวเลือกหลายตัว ซึ่งแต่ละตัวเลือกมีข้อดีและข้อเสียที่แตกต่างกัน ระบบ 'ลั้งเล' ได้นำแนวคิดของเมทริกซ์การตัดสินใจ (Decision Matrix) มาประยุกต์ใช้เพื่อช่วยประเมินและวิเคราะห์ตัวเลือกต่าง ๆ โดยอ้างอิงจากเกณฑ์ที่ชัดเจนและมีโครงสร้างอย่างเป็นระบบ ทั้งนี้ยังสามารถรองรับการใช้งานได้หลากหลายบริบท ทั้งในชีวิตประจำวันและในเชิงธุรกิจ

1.2 วัตถุประสงค์ของการศึกษา

- 1.2.1 เพื่อส่งเสริมการใช้ข้อมูลเชิงวิเคราะห์ในการตัดสินใจอย่างมีประสิทธิภาพ
- 1.2.2 เพื่อยกระดับการเรียนรู้และพัฒนาทักษะการตัดสินใจ
- 1.2.3 เพื่อช่วยให้ผู้ใช้งานมองเห็นภาพรวมและความเป็นไปได้ของแต่ละตัวเลือก
- 1.2.4 เพื่อให้ผู้ใช้งานสามารถนำระบบไปใช้เป็นแนวทางในการตัดสินใจ

1.3 ขอบเขตการศึกษา

- 1.3.1 ศึกษาและประยุกต์ใช้หลักการตัดสินใจเชิงวิเคราะห์โดยใช้เมทริกซ์การตัดสินใจ (Decision Matrix) เพื่อเปรียบเทียบตัวเลือกต่าง ๆ อย่างมีระบบ
- 1.3.2 ใช้ความรู้เกี่ยวกับการวิเคราะห์สภาพแวดล้อมและศักยภาพ (SWOT Analysis) ในการวิเคราะห์ปัจจัยภายในและภายนอก เพื่อช่วยกำหนดเกณฑ์และน้ำหนักที่สำคัญในเมทริกซ์การตัดสินใจ (Decision Matrix)
- 1.3.3 พัฒนาระบบโดยใช้ภาษาไพทอน (Python) และไลบรารี Streamlit บนเว็บแอปพลิเคชัน

1.4 ผลที่คาดว่าจะได้รับ

- 1.4.1 ผู้ใช้สามารถนำข้อมูลทีวิเคราะห์ได้ไปปรับใช้ในสถานการณ์จริง
- 1.4.2 ผู้ใช้สามารถมองเห็นภาพรวมของปัจจัยสำคัญที่เกี่ยวข้องกับตัวเลือกต่าง ๆ ได้อย่างชัดเจน
- 1.4.3 ผู้ใช้จะได้ฝึกวิเคราะห์ปัจจัยต่าง ๆ ในการตัดสินใจอย่างมีเหตุผล
- 1.4.4 ผู้ใช้สามารถเปรียบเทียบตัวเลือกต่าง ๆ ได้อย่างมีเหตุผลผ่าน Decision Matrix และ SWOT Analysis ช่วยลดความลังเลและเพิ่มความมั่นใจในการตัดสินใจ

1.5 แผนการดำเนินการ

ตารางที่ 1.1 แผนการดำเนินการ

การวางแผนการดำเนินงาน	ผู้ได้รับมอบหมาย	พ.ศ. 2567						
		สัปดาห์ ที่ 1	สัปดาห์ ที่ 2	สัปดาห์ ที่ 3	สัปดาห์ ที่ 4	สัปดาห์ ที่ 5	สัปดาห์ ที่ 6	สัปดาห์ ที่ 7
1.การวางแผน	จิรนนท์, ปุณยวีร์, รศิราภ, สิริพัชร							
2.ศึกษาค้นคว้า และเก็บรวบรวม ข้อมูล	จิรนนท์, ปุณยวีร์, รศิราภ, สิริพัชร							
4.การพัฒนา ระบบ	จิรนนท์, ปุณยวีร์, รศิราภ, สิริพัชร							
5.การจัดทำ เอกสารรายงาน	จิรนนท์, ปุณยวีร์, รศิราภ, สิริพัชร							

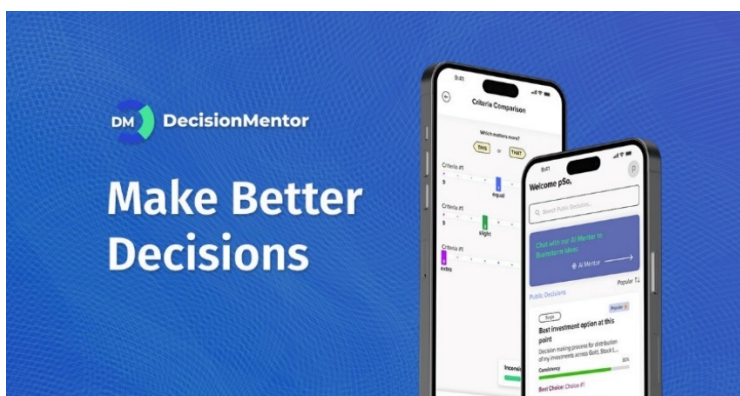
บทที่ 2

ทฤษฎีพื้นฐานและงานที่เกี่ยวข้อง

2.1 ข้อมูลเกี่ยวกับธุรกิจต้นแบบ

2.1.1 Decision Mentor

แอปพลิเคชันที่ช่วยผู้ใช้งานในการตัดสินใจในสถานการณ์ที่ซับซ้อน โดยใช้เครื่องมือและวิธีการวิเคราะห์ที่เป็นระบบ เช่น Decision Matrix, What-If Analysis, และ Multi-Criteria Decision Analysis (MCDA) แอปพลิเคชันนี้ออกแบบมาเพื่อช่วยให้ผู้ใช้งานสามารถเปรียบเทียบตัวเลือกต่าง ๆ ตามเกณฑ์ที่กำหนด และค้นหาตัวเลือกที่เหมาะสมที่สุดสำหรับความต้องการของตนเอง



รูปภาพที่ 2.1 แอปพลิเคชัน Decision Mentor

2.2 ทฤษฎีที่เกี่ยวข้อง

2.2.1 เมทริกซ์การตัดสินใจ (Decision Matrix)

Decision Matrix หรือ เมทริกซ์การตัดสินใจ เป็นเครื่องมือที่ช่วยให้ผู้ใช้งานสามารถเปรียบเทียบและตัดสินใจเกี่ยวกับตัวเลือกต่าง ๆ ได้อย่างมีประสิทธิภาพ โดยใช้เกณฑ์การประเมินที่กำหนดไว้ล่วงหน้า ตารางนี้ถูกใช้อย่างแพร่หลายในหลายด้าน เช่น ธุรกิจ วิศวกรรม การจัดการโครงการ และการแก้ปัญหาเชิงกลยุทธ์

องค์ประกอบของ Decision Matrix

- 1) ตัวเลือก (Alternatives) คือ รายการของทางเลือกที่เราต้องการเปรียบเทียบ เช่น ผลิตภัณฑ์ แผนงาน หรือกลยุทธ์
- 2) เกณฑ์ (Criteria) คือ สิ่งที่ใช้วัดและประเมินตัวเลือก เช่น ราคา คุณภาพ เวลา หรือผลกระทบต่อสิ่งแวดล้อม

- 3) น้ำหนักของเกณฑ์ (Weights) คือ สิ่งที่แสดงถึงความสำคัญของแต่ละเกณฑ์ (ค่าตั้งแต่ 0 ถึง 1 หรือ 0 ถึง 100%)
- 4) คะแนนของตัวเลือก (Scores) คือ คะแนนที่ตัวเลือกแต่ละตัวได้รับในแต่ละเกณฑ์ (อยู่ในช่วง 1-10 หรือ 1-100)
- 5) คะแนนรวม (Weighted Scores) คือ ผลรวมของคะแนนที่คำนวณจากคะแนนในแต่ละเกณฑ์คูณกับน้ำหนักของเกณฑ์นั้น ๆ

การใช้งาน Decision Matrix


- 1) การเปรียบเทียบตัวเลือกหลายตัว ใช้ในกรณีที่มีหลายตัวเลือกและต้องการตัดสินใจเลือกตัวเลือกที่เหมาะสมที่สุด
- 2) การวิเคราะห์ข้อมูลเชิงปริมาณและคุณภาพ ช่วยให้สามารถเปรียบเทียบตัวเลือกที่มีข้อมูลหลากหลายมิติได้ในรูปแบบที่มีโครงสร้าง
- 3) การเพิ่มความโปร่งใส การใช้เกณฑ์และน้ำหนักที่ชัดเจนช่วยลดอคติและความไม่แน่นอนในกระบวนการตัดสินใจ

ขั้นตอนการสร้าง Decision Matrix

- 1) ระบุทางเลือกและเกณฑ์ที่เกี่ยวข้อง
- 2) กำหนดน้ำหนักให้แต่ละเกณฑ์ตามความสำคัญ
- 3) ให้คะแนนแต่ละตัวเลือกตามเกณฑ์
- 4) คำนวณคะแนนรวมของตัวเลือกแต่ละตัวโดยใช้สูตร

$$\text{คะแนนรวม (Weighted Scores)} = \sum (\text{คะแนนของเกณฑ์} * \text{น้ำหนักของเกณฑ์})$$

- 5) เปรียบเทียบคะแนนรวมและเลือกตัวเลือกที่มีคะแนนสูงสุด

<div> <div>  พุ่มทริกซ์ PUGH MATRIX </div> <div> เครื่องมือช่วยประเมินตัวเลือกตามเกณฑ์ตามความสำคัญ เพื่อช่วยในการตัดสินใจเลือกตัวเลือกที่เหมาะสมที่สุด ให้ผลออกมาเป็น </div> <div>  SERVICE DESIGN THAILAND </div> </div>									
ทางเลือกเปรียบเทียบ	น้ำหนักเกณฑ์ (Baseline)	น้ำหนัก 1-5 (Weight) X	แบบที่ 1	แบบที่ 2	แบบที่ 3	แบบที่ 4	แบบที่ 5	แบบที่ 6	
ทางเลือกที่ 1:									
ทางเลือกที่ 2:									
ทางเลือกที่ 3:									
ทางเลือกที่ 4:									
ทางเลือกที่ 5:									
ทางเลือกที่ 6:									
ทางเลือกที่ 7:									
ทางเลือกที่ 8:									
ทางเลือกที่ 9:									
ทางเลือกที่ 10:									
ทางเลือกที่ 11:									
ทางเลือกที่ 12:									

รูปภาพที่ 2.2 เมทริกซ์การตัดสินใจ (Decision Matrix)

2.2.2 การวิเคราะห์สภาพแวดล้อมและศักยภาพ (SWOT Analysis)

SWOT Analysis เป็นกระบวนการวิเคราะห์สภาพปัจจุบันขององค์กรหรือหน่วยงาน เพื่อระบุ จุดแข็ง (Strengths), จุดอ่อน (Weaknesses), โอกาส (Opportunities), และ อุปสรรค (Threats) ที่อาจมีผลกระทบต่อการดำเนินงานและการบรรลุเป้าหมายในอนาคต โดยใช้ข้อมูลทั้งจาก ปัจจัยภายใน และ ปัจจัยภายนอก เพื่อช่วยให้องค์กรเข้าใจสถานการณ์รอบด้านและกำหนดกลยุทธ์ที่เหมาะสม โดยสิ่งที่ต้องนำมาพิจารณา คือ

1) Strengths (จุดแข็ง)

ความได้เปรียบที่องค์กรมีเหนือคู่แข่ง หรือ สิ่งที่องค์กรทำได้ดี เช่น ทรัพยากรที่มีคุณภาพ, ความชำนาญเฉพาะทาง, หรือชื่อเสียงในตลาด

2) Weaknesses (จุดอ่อน)

ข้อเสียเปรียบหรือสิ่งที่องค์กรควรปรับปรุง ตัวอย่างเช่น ขาดการบริหารจัดการที่มีประสิทธิภาพ, การขาดทรัพยากรที่สำคัญ, หรือกระบวนการที่ล้าสมัย

3) Opportunities (โอกาส)

โอกาสจากปัจจัยภายนอกที่สามารถใช้เพื่อขยายธุรกิจหรือเพิ่มความสามารถในการแข่งขัน เช่น การเติบโตของตลาด, การเปลี่ยนแปลงของเทคโนโลยี, หรือการลดลงของคู่แข่ง

4) Threats (อุปสรรค)

ปัจจัยภายนอกที่อาจส่งผลกระทบต่อการดำเนินงานขององค์กร ตัวอย่างเช่น การแข่งขันที่รุนแรง, การเปลี่ยนแปลงของกฎหมาย, หรือความไม่แน่นอนทางเศรษฐกิจ

หลักการสำคัญของ SWOT Analysis

การวิเคราะห์สภาพแวดล้อมและศักยภาพ (SWOT Analysis) มุ่งเน้นที่การสำรวจ สภาพการณ์ภายใน และ สภาพการณ์ภายนอก โดยมีเป้าหมายเพื่อ

- 1) รู้เรา โดยการวิเคราะห์จุดแข็งและจุดอ่อนขององค์กรจากมุมมองภายใน
- 2) รู้เขา โดยการวิเคราะห์โอกาสและอุปสรรคจากสภาพแวดล้อมภายนอก
- 3) ประเมินสถานการณ์ โดยการรวมปัจจัยทั้งหมดเข้าด้วยกันเพื่อให้ผู้บริหารสามารถกำหนดวิสัยทัศน์และกลยุทธ์ที่เหมาะสม

กระบวนการวิเคราะห์ SWOT

- 1) รวบรวมข้อมูล

รวบรวมข้อมูลข้อมูลภายใน เช่น ความสามารถขององค์กร, ทรัพยากร, และกระบวนการ นอกจากนี้ต้องมีการรวบรวมข้อมูลภายนอก เช่น แนวโน้มตลาด, คู่แข่ง, และสภาพแวดล้อมทางเศรษฐกิจ

2) แยกข้อมูลตามปัจจัย

แยกข้อมูลตามปัจจัย โดยการระบุจุดแข็งและจุดอ่อนจากข้อมูลภายใน หรือระบุโอกาสและอุปสรรคจากข้อมูลภายนอก

3) จัดทำ SWOT Matrix

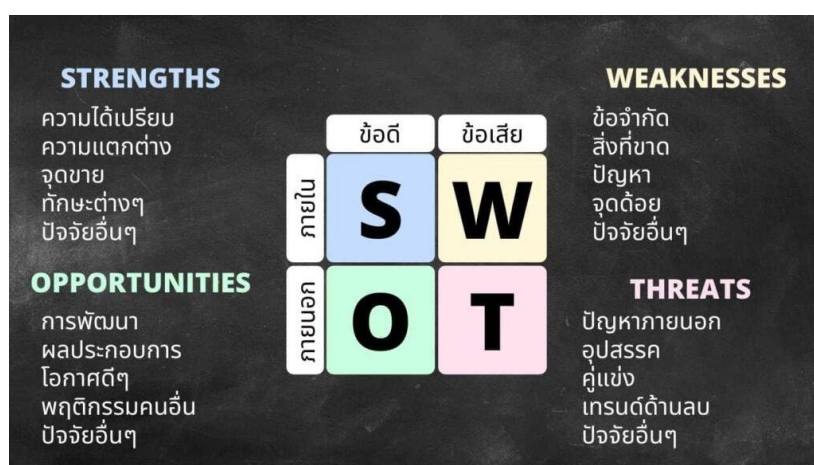
จัดเรียงข้อมูลในรูปแบบตาราง 4 ช่อง (Strengths, Weaknesses, Opportunities, Threats) เพื่อช่วยในการมองเห็นความสัมพันธ์ระหว่างปัจจัยต่าง ๆ

4) กำหนดกลยุทธ์

ใช้ข้อมูล SWOT เพื่อสร้างกลยุทธ์ เช่น การใช้จุดแข็งเพื่อคว้าโอกาส (SO Strategy) หรือการแก้จุดอ่อนเพื่อรับมือกับอุปสรรค (WT Strategy)

ประโยชน์ของการวิเคราะห์สภาพแวดล้อมและศักยภาพ (Swot Analysis)

- 1) ช่วยให้องค์กรเข้าใจจุดแข็งและจุดอ่อนของตนเอง
- 2) ช่วยระบุโอกาสในตลาดและเตรียมพร้อมรับมือกับอุปสรรค
- 3) สนับสนุนการตัดสินใจเชิงกลยุทธ์และการพัฒนาวิสัยทัศน์ในอนาคต
- 4) เป็นเครื่องมือที่ช่วยให้การกำหนดกลยุทธ์องค์กรเป็นไปอย่างเหมาะสมและสอดคล้องกับสภาพแวดล้อม



รูปภาพที่ 2.3 การวิเคราะห์สภาพแวดล้อมและศักยภาพ (SWOT Analysis)

2.2.3 อาหาร

อาหารและโภชนาการมีผลสำคัญต่อสุขภาพ โดยสุขภาพที่ดีหมายถึง สภาพะที่ร่างกายและจิตใจสมบูรณ์ปราศจากโรคภัย การรับประทานอาหารที่ถูกต้องตามหลักโภชนาการช่วยให้ร่างกายได้รับสารอาหารที่ครบถ้วนและเกิดประโยชน์สูงสุด อาหารที่ดีควรหุงต้มอย่างถูกวิธีและประกอบด้วยสารอาหารครบถ้วน ได้แก่ โปรตีน คาร์โบไฮเดรต ไขมัน เกลือแร่ วิตามิน และน้ำ

การเลือกอาหารที่มีคุณค่าทางโภชนาการและเหมาะสมกับความต้องการของร่างกายเป็นสิ่งสำคัญ อาหารแต่ละมื้อควรประกอบด้วยอาหารหลายชนิดเพื่อให้ได้รับสารอาหารครบถ้วน การบริโภคอาหารหลัก 5 หมู่ในสัดส่วนที่เหมาะสมช่วยให้ร่างกายแข็งแรงและเจริญเติบโตได้ดี

การใช้ SWOT กับอาหาร

1) Strengths (จุดแข็ง)

ปัจจัยภายในที่เป็นจุดเด่นของอาหาร : รสชาติอร่อย (Taste), คุณค่าทางโภชนาการสูง (Nutritional Value), วัตถุดิบสดใหม่ (Fresh Ingredients), เมนูเฉพาะที่หาทานที่อื่นไม่ได้ (Unique Menu)

2) Weaknesses (จุดอ่อน)

ปัจจัยภายในที่เป็นข้อด้อยของอาหาร : ราคาแพง (High Cost), ระยะเวลาการเตรียมนาน (Preparation Time), ส่วนผสมบางอย่างอาจไม่เหมาะกับกลุ่มคนแพ้อาหาร (Allergen Concerns), บรรจุภัณฑ์ที่ไม่เหมาะสม (Packaging Issues)

3) Opportunities (โอกาส)

ปัจจัยภายนอกที่เอื้ออำนวยต่ออาหาร : แนวโน้มการรักสุขภาพของผู้บริโภค (Health Trends), ความต้องการอาหารปลอดสารพิษหรือออร์แกนิก (Organic Food Demand), การเพิ่มยอดขายผ่านการตลาดออนไลน์ (Online Marketing), เทรนด์อาหารเฉพาะกลุ่ม เช่น อาหารวีแกน (Vegan Options)

4) Threats (อุปสรรค)

ปัจจัยภายนอกที่เป็นอุปสรรคต่ออาหาร : คู่แข่งในตลาดที่เสนออาหารประเภทเดียวกัน (Market Competition), การเพิ่มขึ้นของต้นทุนวัตถุดิบ (Rising Ingredient Costs), การเปลี่ยนแปลงของกฎระเบียบด้านอาหาร (Regulatory Changes), ความเปลี่ยนแปลงของพฤติกรรมผู้บริโภค (Consumer Behavior Changes)

2.2.4 การแต่งกาย

เสื้อผ้าและเครื่องแต่งกายเป็นสิ่งจำเป็นในชีวิตประจำวันของมนุษย์ ไม่เพียงแต่ช่วยปกปิดร่างกายและป้องกันสภาพอากาศทั้งร้อนและหนาว แต่ยังเป็นสัญลักษณ์ของความสวยงาม ความมั่นใจ และการแสดงออกทางบุคลิกภาพ นอกจากนี้ยังมีประโยชน์ในด้านการใช้งาน เช่น สวมหมวกเพื่อป้องกันแสงแดด หรือติดก๊ิบเพื่อเสริมความสวยงาม

ในอดีต มนุษย์เริ่มต้นด้วยการใช้วัสดุธรรมชาติ เช่น ใบไม้และหนังสัตว์ เพื่อทำเสื้อผ้าและเครื่องประดับร่างกาย ต่อมาเกิดการพัฒนาการทอผ้า การตัดเย็บเสื้อผ้า และการสร้างสรรค์เครื่องแต่งกายที่หลากหลายจนกลายเป็นสิ่งสำคัญในชีวิตประจำวันของเราในปัจจุบัน

ในชีวิตประจำวัน เสื้อผ้าและเครื่องแต่งกายเป็นส่วนหนึ่งที่มนุษย์ต้องใช้อยู่เสมอ ประกอบด้วย

1) เสื้อผ้า

เสื้อกล้าม เสื้อแขนสั้น เสื้อแขนยาว กางเกงขาสั้น กางเกงขายาว กระโปรง ชุดชั้นใน ชุดนอน ชุดกีฬา ชุดว่ายน้ำ และชุดนักเรียน

2) เครื่องแต่งกาย

รองเท้าหลากหลายชนิด (หนัง, ผ้าใบ, รองเท้าแตะ) ถุงเท้า หมวก เข็มขัด นาฬิกา แว่นตา และเครื่องประดับอื่น ๆ เช่น กิ๊บ โบ ยางรัดผม สร้อยคอ ต่างหู

ประโยชน์ของเสื้อผ้าและเครื่องแต่งกาย

เสื้อผ้าและเครื่องแต่งกายมีประโยชน์หลายประการที่ช่วยตอบสนองความต้องการของมนุษย์ในชีวิตประจำวัน ได้แก่

- 1) ช่วยปกปิดร่างกายเพื่อความสะดวกสบายและป้องกันอันตรายจากสภาพแวดล้อม
- 2) ช่วยให้ร่างกายอบอุ่นหรือปรับตัวให้เหมาะสมกับสภาพอากาศ
- 3) สร้างความมั่นใจและช่วยเสริมบุคลิกภาพของผู้สวมใส่
- 4) เพิ่มความสวยงามและการตกแต่งร่างกายให้ดูโดดเด่น
- 5) สร้างความประทับใจแก่ผู้อื่น เมื่อแต่งกายได้เหมาะสมกับโอกาส สถานที่ และฤดูกาล

การใช้ SWOT กับเครื่องแต่งกาย

1) Strengths (จุดแข็ง)

ปัจจัยภายในที่เป็นจุดเด่นของการแต่งกาย : ความสะดวกสบายในการสวมใส่ (Comfort), สไตล์ที่เหมาะสมกับบุคลิกและโอกาส (Style Appropriateness), คุณภาพของวัสดุ (Material Quality), การสร้างความมั่นใจ (Confidence Boost)

2) Weaknesses (จุดอ่อน)

ปัจจัยภายในที่เป็นข้อด้อยของการแต่งกาย : ราคาแพง (High Cost), การดูแลรักษาที่ยุ่งยาก (Difficult Maintenance), ความเข้ากันได้ที่จำกัดกับชุดอื่น ๆ (Limited Mix-and-Match), ความไม่เหมาะสมกับทุกสถานการณ์ (Occasion-Limited)

3) Opportunities (โอกาส)

ปัจจัยภายนอกที่เอื้ออำนวยต่อการแต่งกาย : เทรนด์แฟชั่นใหม่ ๆ (Fashion Trends), โอกาสในการเลือกใช้แบรนด์ที่เข้าถึงได้ง่ายขึ้น (Accessible Brands), ความต้องการการแต่งกายที่เป็นมิตรต่อสิ่งแวดล้อม (Sustainable Fashion), การปรับตัวเข้ากับวัฒนธรรมสากล (Cultural Adaptation)

4) Threats (อุปสรรค)

ปัจจัยภายนอกที่เป็นอุปสรรคต่อการแต่งกาย : ความเปลี่ยนแปลงของเทรนด์แฟชั่นที่รวดเร็ว (Fast-Changing Trends), คู่แข่งในตลาดแฟชั่นที่หลากหลาย (Intense Market Competition), ปัญหาสิ่งแวดล้อมจากการผลิตเสื้อผ้า (Environmental Concerns), การเปลี่ยนแปลงของกฎระเบียบทางการแต่งกาย (Dress Code Policies)

2.2.5 การจัดการ

การจัดการ (Management) หมายถึง การให้กลุ่มบุคคลในองค์กรเข้ามาทำงานร่วมกันเพื่อบรรลุวัตถุประสงค์ร่วมกันขององค์กร ซึ่งประกอบด้วย การวางแผน การจัดการองค์กร การสรรบุคลากร การนำหรือสั่งการ และการควบคุมองค์กรหรือความพยายามที่จะบรรลุวัตถุประสงค์ร่วมกัน

การจัดการทรัพยากรบุคคล ประกอบด้วย การใช้งานตามตำแหน่ง ตลอดจนการจัดวางทรัพยากรบุคคล ทรัพยากรการเงิน ทรัพยากรเทคโนโลยี และทรัพยากรธรรมชาติ ทั้งยังช่วยบริหารให้กับองค์กรต่างๆ ให้มีความก้าวหน้าด้วย

SWOT Analysis ในการจัดการ

1) Strengths (จุดแข็ง)

ปัจจัยภายในที่เป็นจุดเด่นของการจัดการ : ความชัดเจนในเป้าหมายองค์กร, การมีผู้นำที่มีความสามารถ, ทีมงานที่มีทักษะและความเชี่ยวชาญ, ระบบการสื่อสารภายในที่ดี, การใช้เทคโนโลยีที่ทันสมัย

2) Weaknesses (จุดอ่อน)

ปัจจัยภายในที่เป็นข้อด้อยของการจัดการ : การบริหารทรัพยากรที่ไม่มีประสิทธิภาพ, ขาดการวางแผนระยะยาว, ความล่าช้าในการดำเนินงาน, ขาดการพัฒนาทักษะของบุคลากร, ปัญหาความขัดแย้งภายในองค์กร

3) Opportunities (โอกาส)

ปัจจัยภายนอกที่เอื้ออำนวยต่อการจัดการ : การสนับสนุนจากรัฐบาล, การเปลี่ยนแปลงของเทคโนโลยี, การเปิดตลาดใหม่หรือพันธมิตรทางธุรกิจ, กระแสความสนใจด้านการพัฒนาองค์กรอย่างยั่งยืน, ความต้องการของตลาดที่เพิ่มขึ้น

4) Threats (อุปสรรค)

ปัจจัยภายนอกที่เป็นอุปสรรคต่อการจัดการ : การเปลี่ยนแปลงทางเศรษฐกิจ, การแข่งขันในตลาดที่รุนแรง, การเปลี่ยนแปลงของกฎระเบียบ, การขาดแคลนทรัพยากร, การเปลี่ยนแปลงของพฤติกรรมผู้บริโภค

2.2.6 การประกันภัย

ประกันภัย คือ สัญญาความคุ้มครองที่ช่วยบริหารความเสี่ยงภัย หากเกิดเหตุการณ์ที่ไม่คาดคิดต่าง ๆ โดยผู้เอาประกันภัยจะชำระเบี้ยประกันภัยให้กับบริษัทประกันภัย เพื่อแลกกับการได้รับความคุ้มครองจากบริษัทประกันภัยเมื่อมีความเสียหายเกิดขึ้น ซึ่งบริษัทประกันภัยจะชดเชยค่าสินไหมทดแทนให้กับผู้เอาประกันภัยตามสัญญาที่ระบุไว้ในกรมธรรม์ ซึ่งในปัจจุบันประเภทของประกันภัย แบ่งออกเป็น 3 ประเภท ดังนี้

1) การประกันวินาศภัย (Non-Life Insurance) เป็นประกันภัยที่ให้ความคุ้มครองต่อทรัพย์สินหรือทรัพย์สินที่เกี่ยวข้องต่อความเสียหายจากเหตุการณ์วินาศภัย รวมไปถึงความเสียหายที่เกิดจากการกระทำของบุคคล เช่น การประกันอัคคีภัย การประกันรถยนต์ ประกันภัยทางทะเลและขนส่ง ประกันสุขภาพ ฯลฯ

2) การประกันชีวิต (Life Insurance) การประกันภัยที่ให้ความคุ้มครองเป็นเงินก้อนในกรณี ที่ผู้เอาประกันภัยเสียชีวิต ซึ่งมีหลากหลายประเภท ขึ้นอยู่กับวัตถุประสงค์ของการประกันชีวิตและ ความต้องการของผู้เอาประกัน

3) ประกันภัยเกี่ยวกับความรับผิดตามกฎหมายต่อบุคคลภายนอก (Liability Insurance) เป็นประกันที่ให้ความคุ้มครองความรับผิดตามกฎหมายที่มีต่อบุคคลภายนอก สำหรับความบาดเจ็บ ทางร่างกายหรือเสียชีวิต และ/หรือ ความเสียหายต่อทรัพย์สิน ที่มาจากการกระทำของผู้เอาประกันที่ ทำผิดพลาด อุบัติเหตุ หรือเป็นเหตุไม่ได้คาดคิด ทางบริษัทประกันจะเข้ามาดูแลทันที ทั้งค่าใช้จ่ายใน การต่อสู้คดี การชดเชยทรัพย์สินผู้อื่น รวมทั้งชดเชยค่าเสียหายต่อชีวิต เป็นต้น

การใช้ SWOT กับ การประกันภัย

1) Strengths (จุดแข็ง)

ปัจจัยภายในที่เป็นจุดเด่นของการประกันภัย : ความน่าเชื่อถือของบริษัทประกันภัย , ผลิตภัณฑ์หลากหลาย, การบริการหลังการขายที่ดี, ระบบการบริหารความเสี่ยงที่มี ประสิทธิภาพ, การใช้เทคโนโลยีในกระบวนการ

2) Weaknesses (จุดอ่อน)

ปัจจัยภายในที่เป็นข้อด้อยของการประกันภัย : ค่าเบี้ยประกันที่สูง, เงื่อนไขใน กรมธรรม์ที่ซับซ้อน, การเคลมที่ยุ่งยาก, การพึ่งพาตัวแทนประกันมากเกินไป, การปรับตัวต่อ เทคโนโลยีที่ล่าช้า

3) Opportunities (โอกาส)

ปัจจัยภายนอกที่เอื้ออำนวยต่อการประกันภัย : การเพิ่มขึ้นของความต้องการความ คุ้มครอง, การสนับสนุนจากภาครัฐ, การขยายตัวของตลาดดิจิทัล, การเพิ่มความตระหนักใน ความสำคัญของการวางแผนการเงิน, การเกิดของผลิตภัณฑ์ใหม่

4) Threats (อุปสรรค)

ปัจจัยภายนอกที่เป็นอุปสรรคต่อการประกันภัย : การแข่งขันที่รุนแรงในตลาด, ความเสี่ยงจากภัยธรรมชาติหรือโรคระบาด, การเปลี่ยนแปลงของกฎหมาย, พฤติกรรม ผู้บริโภคที่เปลี่ยนแปลง, ความเสี่ยงจากเทคโนโลยี

2.2.7 การตลาดออนไลน์

Digital Marketing หรือ การตลาดดิจิทัล ก็คือการตลาดบนโลกดิจิทัล ที่มีอุปกรณ์อิเล็กทรอนิกส์เป็นสื่อกลาง มีการใช้ช่องทางออนไลน์ต่างๆ และใช้ Martech Tools มาช่วยเหลือในการทำตลาดบนโลกออนไลน์ เพื่อให้มีประสิทธิภาพสูงสุด เป็นผลดีต่อสินค้าหรือบริการ รวมไปถึงแบรนด์

Online Marketing หรือ การตลาดออนไลน์ เป็น วิธีการหนึ่งของ Digital Marketing โดย Online Marketing จะอาศัยอินเทอร์เน็ตในการสื่อสารเกี่ยวกับสินค้าหรือบริการของแบรนด์ให้เข้าถึงลูกค้าผ่านทางช่องทางต่างๆทั้ง เว็บไซต์ หรือ Social Media Platform เป็นต้น

องค์ประกอบหลักในการทำ Digital Marketing เริ่มต้นตั้งแต่การตั้งเป้าหมาย การรู้จักกลุ่มลูกค้า รู้ว่าลูกค้าต้องการสิ่งใด เลือกกลยุทธ์และอุปกรณ์ช่วยเหลือที่เหมาะสม ใช้ช่องทางออนไลน์ต่างๆที่เข้าถึงเป้าหมายได้ดีที่สุด และต้องนำข้อมูลมาวิเคราะห์และวัดผล เพื่อนำไปพัฒนาทั้งสินค้าหรือบริการ และการทำการตลาด

การใช้ SWOT กับการตลาดออนไลน์

1) Strengths (จุดแข็ง)

ปัจจัยภายในที่เป็นจุดเด่นของการตลาดออนไลน์ : การเข้าถึงกลุ่มเป้าหมายได้กว้างขวาง, ต้นทุนต่ำกว่าการตลาดแบบดั้งเดิม, การวิเคราะห์ข้อมูลที่แม่นยำ, ความสามารถในการปรับเปลี่ยนกลยุทธ์ได้รวดเร็ว, การสร้างปฏิสัมพันธ์กับลูกค้า

2) Weaknesses (จุดอ่อน)

ปัจจัยภายในที่เป็นข้อด้อยของการตลาดออนไลน์ : ความจำเป็นในการใช้เทคโนโลยี, ความอึดตัวของโฆษณา Fatigue), การสร้างเนื้อหาที่มีคุณภาพ, การแข่งขันที่รุนแรง, ความเสี่ยงด้านความปลอดภัย

3) Opportunities (โอกาส)

ปัจจัยภายนอกที่เอื้ออำนวยต่อการตลาดออนไลน์ : การเพิ่มขึ้นของผู้ใช้อินเทอร์เน็ต, การเปลี่ยนแปลงพฤติกรรมผู้บริโภค, เครื่องมือและแพลตฟอร์มที่หลากหลาย, เทคโนโลยีใหม่ ๆ, การตลาดแบบเฉพาะบุคคล (Personalization)

4) Threats (อุปสรรค)

ปัจจัยภายนอกที่เป็นอุปสรรคต่อการตลาดออนไลน์ : การเปลี่ยนแปลงของ อัลกอริทึม, ข้อกำหนดด้านความเป็นส่วนตัว, ความผันผวนของค่าโฆษณา, ความไม่แน่นอนทางเศรษฐกิจ, การแสดงความคิดเห็นเชิงลบหรือรีวิวออนไลน์

2.2.3 การลงทุน

การลงทุนคือการนำทรัพย์สินที่มีมูลค่ามาใช้เพื่อสร้างผลตอบแทนที่มากขึ้นในอนาคต ซึ่งอาจเป็นการลงทุนในสินทรัพย์ที่จับต้องได้ เช่น อสังหาริมทรัพย์ หรือสินทรัพย์ทางการเงิน เช่น หุ้นและพันธบัตร ผู้ลงทุนมีเป้าหมายในการสร้างรายได้หรือเพิ่มมูลค่าของเงินทุนในระยะยาว การลงทุนต่างจากการเก็บเงินทั่วไปที่ไม่เพียงแค่อักรักรักษามูลค่า แต่ยังเน้นการเติบโตและผลตอบแทนด้วย

การลงทุนสามารถแบ่งออกเป็นหลายประเภท ตามระดับความเสี่ยงและผลตอบแทน ตั้งแต่การฝากเงินในธนาคาร ไปจนถึงการลงทุนในหุ้นและอสังหาริมทรัพย์

1) ฝากเงินในธนาคาร

เป็นการลงทุนที่มีความเสี่ยงต่ำสุด มักมีผลตอบแทนในรูปแบบดอกเบี้ยที่ค่อนข้างต่ำ การฝากเงินในธนาคารเป็นทางเลือกที่เหมาะสมสำหรับผู้ที่ต้องการรักษาความปลอดภัยของเงินทุนและไม่ต้องการเสี่ยงมาก

2) พันธบัตรรัฐบาล

เป็นการให้เงินกู้ยืมแก่รัฐบาลเพื่อแลกกับดอกเบี้ยที่คงที่ และจ่ายตามระยะเวลา พันธบัตรรัฐบาลมีความเสี่ยงต่ำเนื่องจากรับประกันโดยรัฐบาล แต่ผลตอบแทนอาจไม่สูงมากนัก

3) ทองคำ

เป็นทางเลือกการลงทุนที่มีความเสี่ยงปานกลาง โดยทองคำเป็นสินทรัพย์ที่มีมูลค่าและถือว่าเป็น "สินทรัพย์ปลอดภัย" โดยเฉพาะในช่วงที่เศรษฐกิจไม่แน่นอน ราคาทองคำมีแนวโน้มที่จะเพิ่มขึ้น การลงทุนในทองคำสามารถทำได้ทั้งในรูปแบบทองคำแท่ง ทองรูปพรรณ หรือกองทุนทองคำ

4) กองทุนรวม

เป็นการลงทุนที่รวมเงินทุนจากหลายๆ แห่งมาลงทุนในสินทรัพย์หลายประเภท เช่น หุ้น พันธบัตร และอสังหาริมทรัพย์ โดยมีผู้จัดการกองทุนคอยดูแลและบริหารการลงทุน กองทุนรวมมีความเสี่ยงในระดับปานกลาง ขึ้นอยู่กับนโยบายการลงทุนของกองทุน

5) อสังหาริมทรัพย์

การลงทุนในอสังหาริมทรัพย์ เช่น บ้าน คอนโด หรือที่ดิน มีความเสี่ยงสูง เพราะเกี่ยวข้องกับตลาดอสังหาริมทรัพย์และสภาพเศรษฐกิจ แต่ก็มีโอกาสสร้างผลตอบแทนจากค่าเช่า และการเพิ่มมูลค่าของทรัพย์สินเมื่อเวลาผ่านไป

6) หุ้น

เป็นการลงทุนโดยการซื้อส่วนแบ่งของบริษัท และมีส่วนร่วมในผลกำไรของบริษัทนั้นๆ หุ้นมีความเสี่ยงสูงเนื่องจากราคาหุ้นอาจเปลี่ยนแปลงตามสภาพเศรษฐกิจและผลประกอบการของบริษัท แต่ก็มีโอกาสได้รับผลตอบแทนที่สูงกว่าด้วยเช่นกัน

7) ของสะสม

การลงทุนในของสะสม เช่น เหรียญที่ระลึก งานศิลปะ หรือของหายาก มีความเสี่ยงสูง เพราะมูลค่าของสะสมขึ้นอยู่กับความนิยมและความหายากในตลาด การประเมินมูลค่าของสะสมสามารถเปลี่ยนแปลงได้อย่างรวดเร็ว แต่ก็มีโอกาสสร้างผลตอบแทนที่สูงมากหากของสะสมนั้นมีมูลค่าเพิ่มขึ้นตามเวลา วิธีนี้จึงเป็นการลงทุนที่เหมาะสมสำหรับผู้ที่มีความรู้เฉพาะทางและมีความสนใจในด้านนั้นๆ อย่างแท้จริง

การใช้ SWOT กับการลงทุน

1) Strengths (จุดแข็ง)

ปัจจัยภายในที่เป็นจุดเด่นของการลงทุน : การเพิ่มโอกาสในการสร้างรายได้, ตัวเลือกที่หลากหลาย, ศักยภาพในการสร้างความมั่นคงทางการเงิน, การใช้ประโยชน์จากทรัพยากรที่มีอยู่

2) Weaknesses (จุดอ่อน)

ปัจจัยภายในที่เป็นข้อด้อยของการลงทุน : ความเสี่ยงที่จะขาดทุน, ขาดความรู้หรือความเชี่ยวชาญ, สภาพคล่องต่ำในบางประเภทการลงทุน, ต้องการเวลาและความพยายามในการติดตาม

3) Opportunities (โอกาส)

ปัจจัยภายนอกที่เอื้ออำนวยต่อการลงทุน : การเติบโตของเศรษฐกิจ, นวัตกรรมและเทคโนโลยีใหม่ ๆ, การเปิดตลาดใหม่ ๆ, อัตราดอกเบี้ยต่ำ, การสนับสนุนจากรัฐบาล

4) Threats (อุปสรรค)

ปัจจัยภายนอกที่เป็นอุปสรรคต่อการลงทุน : ความผันผวนของตลาดโลก, อัตราเงินเฟ้อ, การแข่งขันที่รุนแรง, ข้อกำหนดและกฎระเบียบที่เปลี่ยนแปลง, ภัยคุกคามทางเทคโนโลยี

2.3 ภาษาไพทอน (Python)

ไพทอนเป็นภาษาโปรแกรมที่มีความหลากหลายและยืดหยุ่นสูง ทำให้เป็นที่นิยมสำหรับการพัฒนาซอฟต์แวร์ในหลากหลายรูปแบบ ตั้งแต่แอปพลิเคชันขนาดเล็กไปจนถึงโปรเจกต์ขนาดใหญ่ ทั้งนักพัฒนามืออาชีพและผู้เริ่มต้นต่างพบว่า Python เป็นเครื่องมือที่ทรงพลังและใช้งานง่าย ด้วยไวยากรณ์ที่ถูกออกแบบมาให้อ่านง่ายและไม่ซับซ้อน ผู้เริ่มต้นสามารถเรียนรู้และเริ่มต้นพัฒนาโปรแกรมได้อย่างรวดเร็ว

นอกจากนี้ Python ยังมีชุมชนผู้ใช้ที่กว้างขวางซึ่งช่วยสนับสนุนการพัฒนาอย่างต่อเนื่อง มีแหล่งข้อมูลออนไลน์ การประชุม และโครงการโอเพนซอร์สที่เปิดโอกาสให้ผู้ใช้งานแบ่งปันความรู้และพัฒนาตนเอง การเรียนรู้ Python ไม่ได้ช่วยเพียงพัฒนาทักษะการเขียนโปรแกรม แต่ยังเปิดโอกาสในหลากหลายด้าน เช่น วิทยาการข้อมูล การพัฒนาเว็บไซต์และแอปพลิเคชัน ไปจนถึงการพัฒนาเทคโนโลยีปัญญาประดิษฐ์

ดังนั้น Python จึงถือเป็นเครื่องมือสำคัญที่ตอบโจทย์นักพัฒนาในยุคที่เทคโนโลยีก้าวหน้าอย่างรวดเร็ว และช่วยให้การสร้างสรรค์โปรเจกต์ใหม่ ๆ เป็นไปได้อย่างง่ายดายและมีประสิทธิภาพ



รูปภาพที่ 2.4 ภาษาไพทอน (Python)

2.4 โปรแกรม Visual Studio Code

Visual Studio Code (VS Code) คือ โปรแกรม Code Editor ของ Microsoft ที่ใช้สำหรับเขียนแก้ไขและตรวจสอบความผิดปกติของโค้ด มีประสิทธิภาพสูง มีส่วนขยาย (Extension) เพื่อเพิ่มขีดความสามารถและสามารถเขียนโค้ดได้หลากหลายภาษา



รูปภาพที่ 2.5 แสดงรูปโปรแกรม Visual Studio Code

ส่วนประกอบหลักของ Visual Studio Code

1) Activity Bar แถบด้านข้างที่รวมการเข้าถึงฟีเจอร์ต่าง ๆ ของ VSCode เช่น

Explorer จัดการไฟล์และโฟลเดอร์ในโปรเจกต์

Search ค้นหาและแทนที่คำในโปรเจกต์

Source Control ใช้จัดการเวอร์ชันของโค้ด (เชื่อมต่อกับ Git ได้โดยตรง)

Run and Debug ใช้สำหรับการดีบั๊กโค้ด

Extensions ติดตั้งและจัดการส่วนขยายที่ช่วยเพิ่มฟังก์ชันการทำงานของ VSCode

2) Editor Area

พื้นที่หลักที่ใช้ในการเขียนและแก้ไขโค้ด สามารถเปิดไฟล์หลาย ๆ ไฟล์ได้พร้อมกัน และยังสามารถแบ่งหน้าจอเพื่อเปิดไฟล์หลายไฟล์ขนานกันได้

3) Editor Area

แถบด้านข้างที่ช่วยให้สามารถเปิดไฟล์ โฟลเดอร์ และเครื่องมือต่าง ๆ ได้ง่ายขึ้น

4) Status Bar

แถบที่อยู่ด้านล่างของหน้าจอ แสดงข้อมูลต่าง ๆ เช่น เวอร์ชันของโค้ด ภาษาโปรแกรมที่กำลังใช้งาน และการเชื่อมต่อกับ Git

5) Command Palette

เมนูคำสั่งที่สามารถเข้าถึงได้โดยการกด Ctrl+Shift+P (หรือ Cmd+Shift+P บน macOS) เพื่อค้นหาฟังก์ชันต่าง ๆ ที่ต้องการใช้งานอย่างรวดเร็ว

6) Integrated Terminal

เทอร์มินัลภายในที่อยู่ด้านล่างของหน้าต่าง VSCode สามารถใช้รันคำสั่งต่าง ๆ ได้โดยไม่ต้องออกจากโปรแกรม

วิธีการใช้งานเบื้องต้นใน Visual Studio Code

- 1) เริ่มต้นโปรเจกต์ สร้างโฟลเดอร์สำหรับโปรเจกต์ จากนั้นเปิด VSCode และเลือก File > Open Folder เพื่อเปิดโฟลเดอร์ดังกล่าวใน VSCode ไฟล์ทั้งหมดภายในโฟลเดอร์นี้จะแสดงในแถบ Explorer ทำให้ง่ายต่อการจัดการไฟล์ในโปรเจกต์
- 2) เขียนและแก้ไขโค้ด เปิดไฟล์ที่ต้องการแก้ไข โค้ดจะปรากฏใน Editor Area ซึ่งมีฟังก์ชันเต็มโค้ดอัตโนมัติและแสดงข้อผิดพลาดแบบเรียลไทม์ สามารถใช้คำสั่ง Ctrl+/ เพื่อคอมเมนต์หรือยกเลิกคอมเมนต์โค้ด
- 3) ดีบั๊กโค้ด ไปที่ Run and Debug ใน Activity Bar เลือกการตั้งค่าการดีบั๊กที่เหมาะสม เช่น Python และกดปุ่ม Start เพื่อเริ่มดีบั๊ก VSCode จะช่วยหยุดการทำงานที่ breakpoints และแสดงค่าตัวแปรต่าง ๆ เพื่อให้สามารถวิเคราะห์และแก้ไขข้อผิดพลาดได้ง่ายขึ้น
- 4) ใช้ Git ในการจัดการเวอร์ชัน โดยใน Source Control สามารถดูประวัติการเปลี่ยนแปลงไฟล์และบันทึกการเปลี่ยนแปลงได้ และสามารถเชื่อมต่อกับ GitHub หรือ Git repository อื่น ๆ เพื่อบริหารเวอร์ชันได้สะดวก
- 5) ติดตั้ง Extensions โดยการเข้าไปที่ Extensions ใน Activity Bar และค้นหาส่วนขยายที่ต้องการ เช่น Python, Prettier, ESLint ฯลฯ เมื่อติดตั้งแล้วจะเพิ่มฟังก์ชันที่ช่วยให้ทำงานได้ง่ายขึ้น

2.5 ภาษาตกแต่งเว็บไซต์ CSS (Cascading Style Sheets: CSS)

CSS (Cascading Style Sheets) เป็นภาษาที่ใช้ในการกำหนดรูปแบบและการตกแต่งของเอกสาร HTML ทำให้หน้าเว็บสวยงามและดูเป็นระเบียบมากขึ้น CSS ช่วยกำหนดลักษณะขององค์ประกอบ เช่น สี, ขนาด, ฟอนต์ และการจัดวางในตำแหน่งต่าง ๆ



รูปภาพที่ 2.6 ภาษาตกแต่งเว็บไซต์ CSS (Cascading Style Sheets: CSS)

ส่วนประกอบหลักของ CSS

- 1) Selectors (ตัวเลือก) ใช้เลือกองค์ประกอบ HTML ที่ต้องการกำหนดสไตล์ มีหลายประเภท เช่น

Element Selector (h1, p): เลือกตามชื่อแท็ก HTML

Class Selector (.className) เลือกตามชื่อคลาส โดยใช้นำหน้าด้วยจุด เช่น .container

ID Selector (#idName) เลือกตาม ID ขององค์ประกอบ โดยใช้นำหน้าด้วยเครื่องหมาย #

เช่น #header

Attribute Selector ([type="text"]): เลือกองค์ประกอบตามแอตทริบิวต์ เช่น [type="text"]

Pseudo-class (:hover, :active): เลือกสถานะเฉพาะขององค์ประกอบ เช่น เมื่อผู้ใช้เลื่อนเมาส์ผ่าน

Pseudo-element (::before, ::after) ใช้สร้างองค์ประกอบเสมือนที่ไม่มีใน HTML เช่น ::before ใช้ใส่เนื้อหาก่อนองค์ประกอบหลัก

- 2) Properties (คุณสมบัติ) คือคุณสมบัติที่ใช้กำหนดสไตล์ขององค์ประกอบ เช่น color, font-size, background-color, width, height ฯลฯ แต่ละคุณสมบัติมีค่าที่สามารถตั้งได้หลากหลายเพื่อปรับแต่งลักษณะต่าง ๆ ขององค์ประกอบ
- 3) Values (ค่า) เป็นค่าที่ถูกกำหนดให้กับ properties เช่น

color: red; กำหนดสีเป็นสีแดง

font-size: 16px; กำหนดขนาดฟอนต์เป็น 16 พิกเซล

ค่าของ properties สามารถเป็นหน่วยต่าง ๆ ได้ เช่น px (พิกเซล), % (เปอร์เซ็นต์), em, rem หรือ vh/vw (view height/view width)

- 4) Declarations (คำสั่งการกำหนด) แต่ละคำสั่งจะประกอบด้วย property: value; กลุ่มของ declarations จะอยู่ภายใน { } และถูกเรียกใช้กับองค์ประกอบ HTML ผ่าน selector
- 5) CSS Rule (กฎ CSS) การกำหนดสไตล์ใน CSS หนึ่งชุดเรียกว่า rule โดย rule ประกอบด้วย selector และ declarations หลาย ๆ ตัวที่อยู่ภายใน { }

2.6 ไลบรารี Streamlit

Streamlit คือ framework หรือไลบรารีสำหรับการพัฒนาเว็บแอปพลิเคชัน ด้วยภาษา Python โดยเฉพาะออกแบบมาให้ใช้งานง่าย เหมาะสำหรับนักวิทยาศาสตร์ข้อมูล (Data Scientists) หรือผู้ที่ต้องการสร้างเครื่องมือโต้ตอบ (Interactive Tools) สำหรับการวิเคราะห์ข้อมูล, การแสดงผลข้อมูล, หรือการพัฒนาโมเดลการเรียนรู้ของเครื่อง (Machine Learning) โดยไม่ต้องมีความรู้เชิงลึกซึ่งในด้านการพัฒนาเว็บแอปพลิเคชัน

ส่วนประกอบหลักของ Streamlit

- 1) การแสดงข้อความ ใช้สำหรับแสดงข้อความในรูปแบบต่าง ๆ เช่น หัวข้อ, ย่อหน้า, หรือคำอธิบาย โดยฟังก์ชันที่ใช้ คือ st.title(), st.header(), st.subheader(), st.text(), st.markdown()
- 2) รับข้อมูลจากผู้ใช้ (Inputs) ใช้วิดเจ็ต (Widgets) เพื่อรับข้อมูล เช่น

st.text_input() : ช่องกรอกข้อความ

st.number_input() : กรอกตัวเลข

st.selectbox() : เลือกจากตัวเลือก

st.slider() : สไลด์บาร์สำหรับเลือกค่า

- 3) แสดงข้อมูล เช่น DataFrame, กราฟ, หรือภาพ

st.dataframe() : แสดงตารางข้อมูล

st.line_chart(), st.bar_chart() : สร้างกราฟ

`st.image()` : แสดงภาพ

- 4) การโต้ตอบ (Interactivity) รองรับการอัปเดตแบบเรียลไทม์ด้วยฟังก์ชัน

เช่น `st.button()`, `st.checkbox()`, และ `st.radio()`

- 5) การวาง Layout หรือ การจัดรูปแบบและตำแหน่งองค์ประกอบ

เช่น `st.sidebar` สำหรับเมนูด้านข้าง หรือการใช้คอลัมน์ (`st.columns()`)

- 6) การแสดงผลพล็อตเชิงภาพ รองรับไลบรารีกราฟ

เช่น Matplotlib, Plotly, Altair, และอื่น ๆ ผ่านฟังก์ชัน เช่น `st.pyplot()`, `st.plotly_chart()`



รูปภาพที่ 2.7 ไลบรารี Streamlit

2.7 ไลบรารี Pandas

Pandas เป็นไลบรารีสำหรับภาษา Python ที่ถูกออกแบบมาเพื่อการวิเคราะห์และจัดการข้อมูล (Data Analysis & Data Manipulation) โดยเฉพาะอย่างยิ่งข้อมูลในรูปแบบตาราง เช่น ข้อมูลจากไฟล์ CSV, Excel, SQL Database หรือข้อมูลที่ดึงมาจาก APIs และการ Web Scraping

ประโยชน์ของ Pandas

- 1) การจัดการข้อมูล

Pandas ช่วยให้สามารถเพิ่ม ลบ หรือแก้ไขข้อมูลในรูปแบบแถวและคอลัมน์ได้อย่างมีประสิทธิภาพ รวมถึงการกรองข้อมูล (Filtering) และการรวมข้อมูลจากหลายแหล่ง (Merging, Joining)

- 2) การอ่านและเขียนข้อมูล

Pandas รองรับการอ่านและเขียนข้อมูลจากหลากหลายรูปแบบ เช่น CSV, Excel, JSON, และ SQL ทำให้สะดวกในการนำเข้าหรือส่งออกข้อมูลสำหรับการวิเคราะห์

3) การคำนวณและสรุปผลข้อมูล

สามารถใช้ Pandas ในการคำนวณทางคณิตศาสตร์ เช่น การหาผลรวม ค่าเฉลี่ย และการนับข้อมูล รวมถึงการสรุปผลข้อมูลด้วย Grouping หรือ Pivot Table

4) การจัดการข้อมูลขาดหาย (Missing Data)

Pandas มีฟังก์ชันสำหรับจัดการข้อมูลที่ขาดหาย เช่น การเติมค่า (Imputation) หรือ การลบข้อมูลที่ไม่สมบูรณ์

5) การจัดการข้อมูลเวลา (Time Series Data)

Pandas รองรับการจัดการข้อมูลที่เกี่ยวข้องกับวันเวลา เช่น การรวมกลุ่มข้อมูลตามเดือนหรือปี ซึ่งเหมาะสำหรับการวิเคราะห์ข้อมูลแบบ Time Series

โครงสร้างข้อมูลใน Pandas

1) Series

โครงสร้างข้อมูลแบบ 1 มิติ คล้ายกับ List หรือ Array ใน Python ใช้สำหรับเก็บข้อมูลที่มีดัชนี (Index) กำกับ

2) DataFrame

โครงสร้างข้อมูลแบบ 2 มิติที่ประกอบด้วยแถวและคอลัมน์ คล้ายกับตารางใน Excel ซึ่งเป็นโครงสร้างข้อมูลที่สำคัญที่สุดใน Pandas

ส่วนประกอบของ Pandas

Pandas มีโครงสร้างข้อมูลหลักที่สำคัญ 2 ประเภท ซึ่งเป็นพื้นฐานในการจัดการข้อมูลอย่างมีประสิทธิภาพ

2.1) Series (ข้อมูลแบบ 1 มิติ)

Series เป็นโครงสร้างข้อมูลแบบ 1 มิติ คล้ายกับ List หรือ Array ใน Python แต่มี Index หรือป้ายกำกับระบุค่าข้อมูลแต่ละตัว ทำให้สามารถอ้างอิงข้อมูลได้โดยตรง

2.2) DataFrame (ข้อมูลแบบ 2 มิติ)

DataFrame เป็นโครงสร้างข้อมูลแบบ 2 มิติ ประกอบด้วยแถว (Row) และคอลัมน์ (Column) ซึ่งสามารถเก็บข้อมูลหลากหลายประเภทไว้ในแต่ละคอลัมน์ได้ คล้ายกับตารางใน Excel



รูปภาพที่ 2.8 โลโก้ Pandas

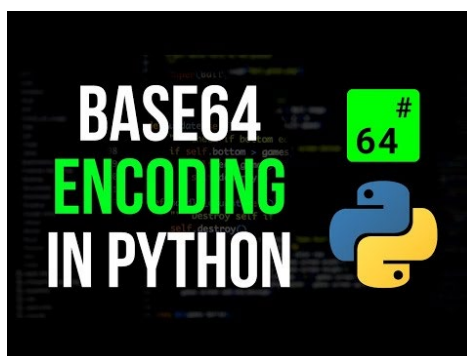
2.8 โลโก้ Base64

Base64 เป็นวิธีการเข้ารหัสข้อมูลที่ช่วยแปลงข้อมูลต้นฉบับ เช่น ไฟล์, ข้อความ, หรือรูปภาพ ให้อยู่ในรูปแบบข้อความ ASCII ที่สามารถส่งผ่านระบบที่รองรับเฉพาะอักขระ ASCII ได้อย่างปลอดภัย เช่น ระบบอีเมล, JSON, หรือ HTML โดย Base64 ใช้อักขระชุด a-z, A-Z, ตัวเลข 0-9 และเครื่องหมายพิเศษ +, / และ = (สำหรับ Padding)

หลักการทำงาน

- 1) ข้อมูลต้นฉบับจะถูกแปลงเป็นไบนารี (binary) และแบ่งออกเป็นชุดข้อมูลขนาด 6 บิต
 - 2) แต่ละชุดข้อมูล 6 บิตจะถูกแปลงไปยังอักขระ Base64 ตามตารางเข้ารหัสมาตรฐาน
 - 3) หากข้อมูลไม่ครบจำนวนบิตที่กำหนด ระบบจะเติม Padding (=) เพื่อให้การเข้ารหัสสมบูรณ์
- ประโยชน์

- 1) การส่งข้อมูลผ่านโปรโตคอลที่รองรับเฉพาะข้อความ
ใช้ส่งข้อมูลไบนารี เช่น รูปภาพหรือไฟล์ ผ่านระบบที่รองรับเฉพาะข้อความ ASCII เช่น JSON หรือ XML
- 2) การฝังข้อมูลในเอกสาร HTML หรือ CSS
ช่วยฝังรูปภาพหรือไฟล์ในรูปแบบ data: scheme เพื่อให้สามารถใช้งานได้ทันทีในเอกสาร
- 3) การส่งไฟล์ในอีเมล
ใช้ใน MIME (Multipurpose Internet Mail Extensions) เพื่อเข้ารหัสไฟล์แนบในอีเมล ให้อยู่ในรูปแบบที่สามารถส่งผ่านโปรโตคอลได้



รูปภาพที่ 2.9 การเข้ารหัสข้อมูล Base64

2.9 ไลบรารี Matplotlib

Matplotlib เป็นไลบรารีมาตรฐานสำหรับการสร้างกราฟในภาษาไพทอน โดยมีความสามารถในการสร้างกราฟ 2 มิติ (2D) และกราฟ 3 มิติ (3D) หลากหลายรูปแบบ เช่น กราฟเส้น (Line Plot), กราฟแท่ง (Bar Chart), ฮิสโตแกรม (Histogram), กราฟพาย (Pie Chart) และกราฟการกระจายตัว (Scatter Plot) ไลบรารีนี้เหมาะสำหรับการวิเคราะห์และแสดงผลข้อมูลในรูปแบบกราฟ เพื่อให้สามารถอธิบายและสื่อสารข้อมูลได้ง่ายขึ้น

คุณสมบัติเด่นของ Matplotlib

1) สร้างกราฟได้หลากหลาย

รองรับการสร้างกราฟหลากหลายรูปแบบ เช่น กราฟเส้น กราฟแท่ง ฮิสโตแกรม และกราฟพาย

2) ปรับแต่งกราฟได้อย่างยืดหยุ่น

สามารถปรับแต่งลักษณะของกราฟ เช่น สี เส้น ตัวอักษร และคำอธิบายต่าง ๆ ได้ตามต้องการ

3) รองรับการทำงานร่วมกับ pandas

สามารถใช้ข้อมูลจาก DataFrame ของไลบรารี pandas ได้โดยตรง

4) รองรับกราฟ 3 มิติ

ช่วยเพิ่มความสามารถในการวิเคราะห์ข้อมูลเชิงลึก

ฟังก์ชันสำคัญของ Matplotlib

Matplotlib ใช้โมดูลย่อย matplotlib.pyplot ในการสร้างและจัดการกราฟ โดยมีฟังก์ชันที่สำคัญดังนี้

1) กราฟเส้น (Line Plot)

แสดงความสัมพันธ์ระหว่างตัวแปรอิสระและตัวแปรตาม

คำสั่ง : `plt.plot(x, y)`

ตัวเลือกเพิ่มเติม :

`color` : กำหนดสีของเส้น

`linewidth` : กำหนดความหนาของเส้น

`linestyle` : กำหนดรูปแบบเส้น เช่น '-' (เส้นตรง), '--' (เส้นประ)

2) กราฟแท่ง (Bar Chart)

ใช้แสดงการเปรียบเทียบค่าของตัวแปร

คำสั่ง: `plt.bar(x, y)`

ตัวเลือกเพิ่มเติม :

`color` : กำหนดสีของแท่ง

`width` : กำหนดความกว้างของแท่ง

3) ฮิสโตแกรม (Histogram)

ใช้แสดงการกระจายตัวของข้อมูล

คำสั่ง : `plt.hist(data, bins=n)`

ตัวเลือกเพิ่มเติม :

`bins` : กำหนดจำนวนช่วงข้อมูล

`color` : กำหนดสีของกราฟ

`edgecolor` : กำหนดสีของขอบแท่ง

4) กราฟการกระจายตัว (Scatter Plot)

ใช้แสดงความสัมพันธ์ระหว่างข้อมูลสองชุดในรูปแบบจุด

คำสั่ง : `plt.scatter(x, y)`

ตัวเลือกเพิ่มเติม :

`color` : กำหนดสีของจุด

`s` : กำหนดขนาดของจุด

`alpha` : กำหนดความโปร่งใส

5) กราฟพาย (Pie Chart)

ใช้แสดงข้อมูลในรูปแบบสัดส่วนของกราฟวงกลม

คำสั่ง : `plt.pie(sizes, labels=labels)`

ตัวเลือกเพิ่มเติม :

`autopct` : แสดงเปอร์เซ็นต์ในกราฟ

`startangle` : หมุนจุดเริ่มต้นของวงกลม

6) กราฟ 3 มิติ (3D Plot)

ใช้แสดงข้อมูลในรูปแบบสามมิติ

คำสั่ง : ใช้โมดูล `mpl_toolkits.mplot3d`

7) การปรับแต่งกราฟ (Customization)

เพิ่มคำอธิบาย :

`plt.title("Title")`

`plt.xlabel("X-axis Label")`

`plt.ylabel("Y-axis Label")`

เพิ่มตารางแกน (Grid) :

`plt.grid(True)`



รูปภาพที่ 2.10 โลโก้ Matplotlib

2.10 โลโก้ Seaborn

Seaborn เป็นไลบรารีในภาษาไพทอนที่พัฒนาขึ้นเพื่อช่วยในการสร้างกราฟและการแสดงผลข้อมูลในรูปแบบที่สวยงาม โดยไลบรารีนี้ถูกสร้างขึ้นบนพื้นฐานของ Matplotlib แต่เพิ่มความสามารถในการจัดการข้อมูลและสร้างกราฟเชิงสถิติที่ซับซ้อนได้อย่างง่ายดาย

จุดเด่นของ Seaborn

- 1) การจัดการข้อมูลง่ายและยืดหยุ่น

รองรับการทำงานกับข้อมูลในรูปแบบ pandas DataFrame โดยตรง

- 2) การแสดงผลข้อมูลเชิงสถิติ

มีฟังก์ชันที่สามารถแสดงผลข้อมูลเชิงสถิติได้หลากหลาย เช่น การแจกแจงข้อมูล (Distribution), การเปรียบเทียบกลุ่มข้อมูล, และการวิเคราะห์ความสัมพันธ์

- 3) กราฟที่สวยงามและชัดเจน

กราฟที่ได้จาก Seaborn มีรูปแบบที่ดูทันสมัย อ่านง่าย และสามารถปรับแต่งได้ตามความต้องการ

- 4) เครื่องมือวิเคราะห์ข้อมูลแบบครบวงจร

รองรับการสร้างกราฟหลายรูปแบบ เช่น Heatmap, Pair Plot และ Violin Plot

ฟังก์ชันหลักของ Seaborn

- 1) การวิเคราะห์การแจกแจงข้อมูล (Distribution)

`sns.histplot()` : แสดงฮิสโตแกรม (Histogram) เพื่อดูการกระจายตัวของข้อมูล

`sns.kdeplot()` : แสดงกราฟ Kernel Density Estimation (KDE)

`sns.boxplot()` : แสดง Box Plot เพื่อดูค่ากลาง ค่าต่ำสุด และค่าสูงสุดของข้อมูล

2) การวิเคราะห์ความสัมพันธ์ (Correlation)

`sns.scatterplot()` : แสดงกราฟ Scatter Plot เพื่อวิเคราะห์ความสัมพันธ์ระหว่างตัวแปร

`sns.heatmap()` : แสดงความสัมพันธ์ระหว่างตัวแปรในรูปแบบ Heatmap

3) การเปรียบเทียบข้อมูลหมวดหมู่ (Categorical Data)

`sns.barplot()` : แสดงค่ากลางของข้อมูลแต่ละกลุ่มในรูปแบบ Bar Plot

`sns.countplot()` : แสดงการนับจำนวนข้อมูลแต่ละหมวดหมู่

`sns.violinplot()` : แสดง Violin Plot เพื่อดูการกระจายตัวของข้อมูลในแต่ละกลุ่ม

4) การสร้าง Pair Plot

`sns.pairplot()` : แสดงกราฟความสัมพันธ์ของตัวแปรหลายตัวใน DataFrame

5) การปรับแต่งธีมและสีของกราฟ

Seaborn รองรับการปรับแต่งธีมและสีของกราฟให้เหมาะสมกับการใช้งาน

`sns.set_theme()` : ตั้งค่าธีมของกราฟ เช่น "darkgrid", "whitegrid", "dark", "white", "ticks"

`sns.color_palette()` : ปรับชุดสีสำหรับกราฟ

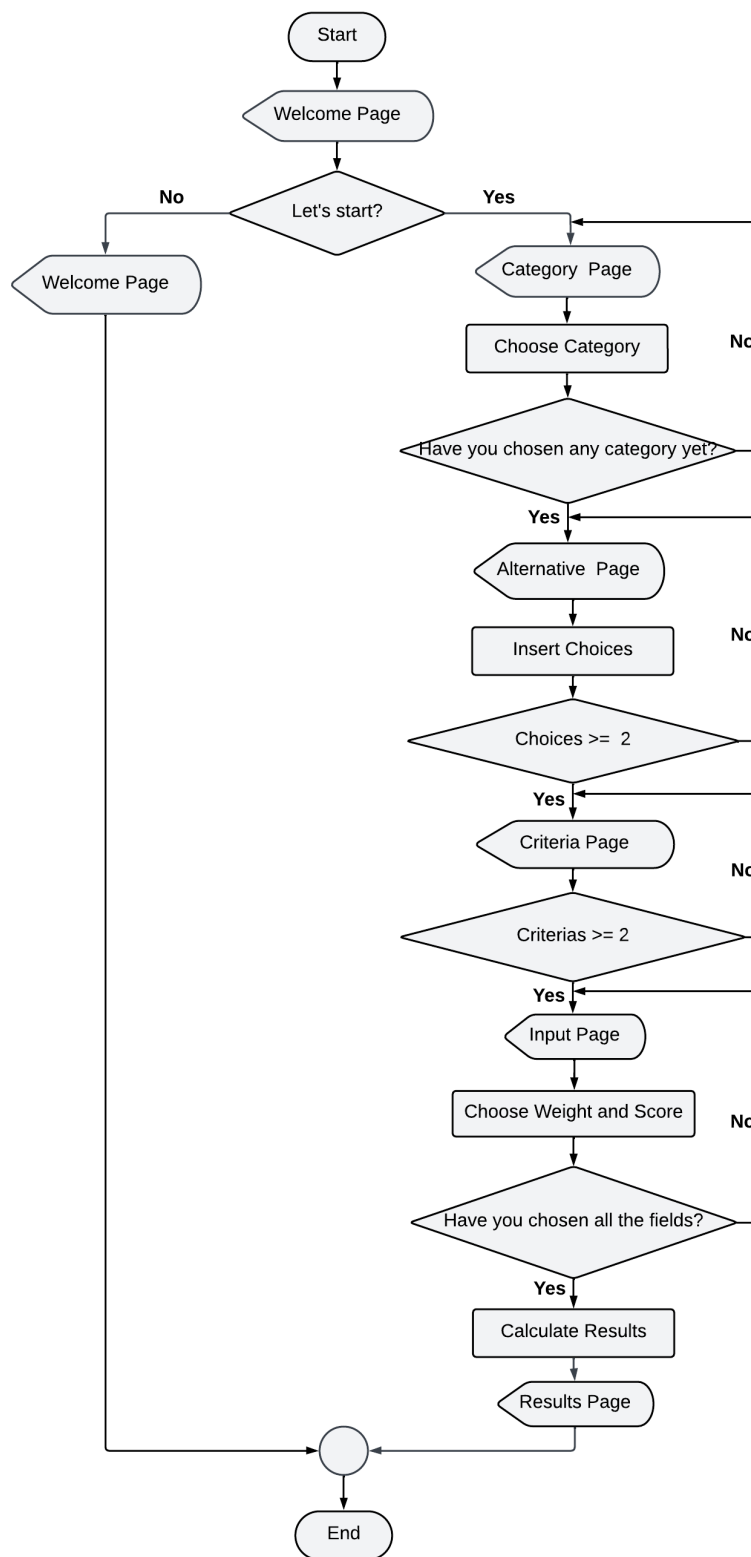


รูปภาพที่ 2.11 โลโก้ Seaborn

บทที่ 3

วิธีการดำเนินการ

3.1 การออกแบบและวางแผนโครงสร้างโปรแกรม



รูปภาพที่ 3.1 แผนภาพการทำงานภาพรวมของทั้งระบบ

3.2 User Interface

3.2.1 หน้าจอเริ่มต้น



รูปภาพที่ 3.2 หน้าจอเริ่มต้น

หน้าจอเริ่มต้น ผู้ใช้ทำการกดปุ่ม “ล้งเล” เพื่อเริ่มต้นการใช้งานและไปยังหน้าจอการเลือกหมวดหมู่

3.2.2 หน้าจอการเลือกหมวดหมู่



รูปภาพที่ 3.3 หน้าจอการเลือกหมวดหมู่

หน้าจอการเลือกหมวดหมู่ ผู้ใช้ต้องทำการกดปุ่มหมวดหมู่ที่ผู้ใช้สนใจ หากผู้ใช้สนใจหมวดหมู่อื่นนอกเหนือจากที่มีในระบบ ให้ทำการกดปุ่ม “หมวดอื่น”

3.2.3 หน้าจอการเพิ่มตัวเลือกที่จะพิจารณา

รูปภาพที่ 3.4 หน้าจอการเพิ่มตัวเลือกที่จะพิจารณา

หน้าจอสำหรับการเพิ่มตัวเลือกที่จะพิจารณา ผู้ใช้สามารถเพิ่มตัวเลือกที่อยู่ในระหว่างการตัดสินใจ โดยระบบจะมีฟอร์มที่แสดงช่องกรอกข้อมูลตัวเลือกสำหรับระบุแต่ละตัวเลือกที่ต้องการเปรียบเทียบหรือพิจารณา ผู้ใช้จำเป็นต้องกรอกตัวเลือกอย่างน้อย 2 ตัวเลือก เพื่อให้สามารถไปยังหน้าจอการเพิ่มเกณฑ์ที่ใช้ในการพิจารณาตัวเลือกได้

3.2.4 หน้าจอการเพิ่มเกณฑ์ที่ใช้ในการพิจารณาตัวเลือก

กลับ

คุณเลือกหมวดหมู่ : กิจกรรมในวันหยุด

เกณฑ์ที่เกี่ยวข้องกับหมวดหมู่

เกณฑ์ที่ 1 (สามารถแก้ไขได้) :

ความผ่อนคลาย

ลบเกณฑ์ที่ 1

เกณฑ์ที่ 2 (สามารถแก้ไขได้) :

ค่าใช้จ่าย

ลบเกณฑ์ที่ 2

เกณฑ์ที่ 3 (สามารถแก้ไขได้) :

ความสะดวกในการเดินทาง

ลบเกณฑ์ที่ 3

เพิ่มเกณฑ์อื่น ๆ

เกณฑ์อื่น ๆ :

เพิ่ม

เกณฑ์ทั้งหมด

	เกณฑ์
0	ความผ่อนคลาย
1	ค่าใช้จ่าย
2	ความสะดวกในการเดินทาง

ถัดไป

รูปภาพที่ 3.5 หน้าจอการเพิ่มเกณฑ์ที่ใช้ในการพิจารณาตัวเลือก

หน้าจอเพิ่มเกณฑ์ที่ใช้ในการพิจารณาตัวเลือก หากผู้ใช้ทำการเลือกหมวดหมู่ที่มีอยู่ ระบบจะกำหนดเกณฑ์เพื่อเป็นแนวทางให้ผู้ใช้ 3 เกณฑ์ หากผู้ใช้มีเกณฑ์อื่นจะใช้ในการพิจารณาตัวเลือก ผู้ใช้สามารถเลือกลบเกณฑ์หรือเพิ่มเกณฑ์ได้ โดยผู้ใช้งานจำเป็นต้องระบุเกณฑ์ที่ใช้ในการพิจารณาอย่างน้อย 2 เกณฑ์เพื่อให้สามารถไปยังหน้ากำหนดน้ำหนักและคะแนนได้

3.2.5 หน้าจอการกำหนดน้ำหนักและคะแนน

กลับ

คุณเลือกหมวดหมู่ : กิจกรรมในวันหยุด

คุณกำลังเลือกระหว่าง 3 ตัวเลือก :
เที่ยวธรรมชาติ , เดินเล่นตลาดนัด ,
พักผ่อนอยู่บ้าน

กำหนดน้ำหนักให้กับแต่ละเกณฑ์ตามที่คุณให้ความสำคัญ (100%)

น้ำหนักสำหรับเกณฑ์ 'ความผ่อนคลาย':

30 - +

น้ำหนักสำหรับเกณฑ์ 'ค่าใช้จ่าย':

35 - +

น้ำหนักสำหรับเกณฑ์ 'ความสะดวกในการเดินทาง':

35 - +

น้ำหนักรวมถูกต้อง

กำหนดคะแนนให้แต่ละตัวเลือก (1-10)

ความผ่อนคลาย (Weight: 30%)

ความผ่อนคลาย : เที่ยวธรรมชาติ

8

1 10

ความผ่อนคลาย : เดินเล่นในตลาดนัด

3

1 10

ความผ่อนคลาย : พักผ่อนที่บ้าน

10

1 10

รูปภาพที่ 3.6 หน้าจอการกำหนดน้ำหนักและคะแนน

ค่าใช้จ่าย (Weight: 35%)

ค่าใช้จ่าย : เกี่ยวกับธรรมชาติ



ค่าใช้จ่าย : เดินเล่นในตลาดนัด



ค่าใช้จ่าย : พักผ่อนที่บ้าน



ความสะดวกในการเดินทาง (Weight: 35%)

ความสะดวกในการเดินทาง : เกี่ยวกับธรรมชาติ



ความสะดวกในการเดินทาง : เดินเล่นในตลาดนัด



ความสะดวกในการเดินทาง : พักผ่อนที่บ้าน



สรุปข้อมูล

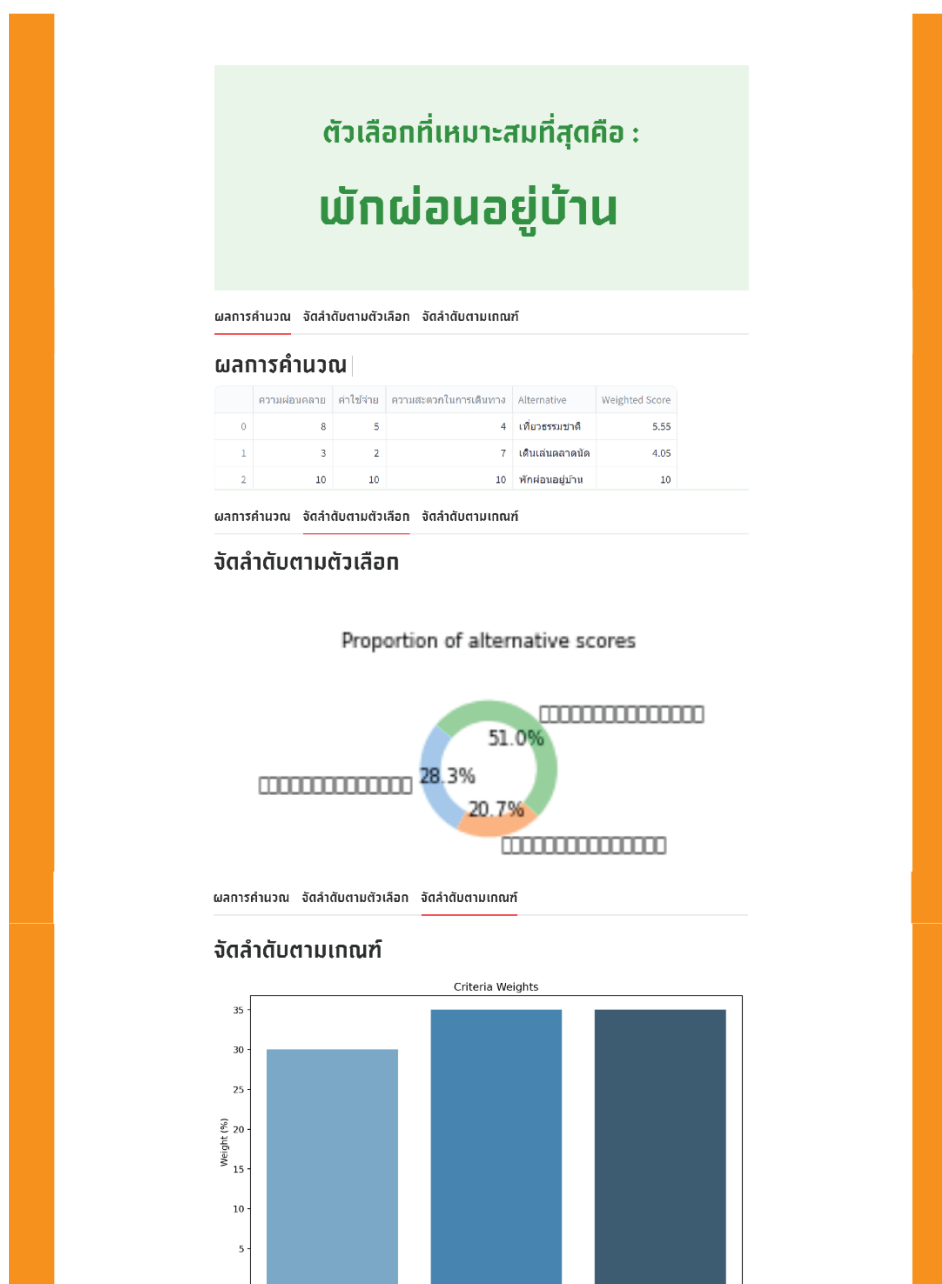
	ความผ่อนคลาย	ค่าใช้จ่าย	ความสะดวกในการเดินทาง	Alternative
0	8	5	4	เกี่ยวกับธรรมชาติ
1	3	2	7	เดินเล่นในตลาดนัด
2	10	10	10	พักผ่อนที่บ้าน

ถัดไป

รูปภาพที่ 3.7 หน้าจอการกำหนดน้ำหนักและคะแนน (2)

หน้าจอการกำหนดน้ำหนักและคะแนน ผู้ใช้สามารถระบุค่าน้ำหนักความสำคัญของแต่ละเกณฑ์ ซึ่งระบบจะตรวจสอบให้ค่าน้ำหนักรวมทั้งหมดเท่ากับ 100% และผู้ใช้อย่างยังสามารถให้คะแนนตัวเลือกในแต่ละเกณฑ์ โดยใช้แถบเลื่อน (Slider) ในช่วงคะแนน 1 ถึง 10 สำหรับตัวเลือกที่ผู้ใช้กำลังพิจารณา

3.2.6 หน้าจอแสดงผลลัพธ์



รูปภาพที่ 3.8 หน้าจอแสดงผลลัพธ์

หน้าจอแสดงผลลัพธ์ แสดงผลลัพธ์ของตัวเลือกที่เหมาะสมที่สุดโดยคำนวณจากคะแนนถ่วงน้ำหนัก (Weighted Scores) พร้อมทั้งแสดงรายละเอียดผลการคำนวณในรูปแบบตาราง ซึ่งรวมถึงคะแนนในแต่ละเกณฑ์และคะแนนถ่วงน้ำหนักของแต่ละตัวเลือก นอกจากนี้ ยังมีการแสดงผลการเปรียบเทียบตัวเลือกในรูปแบบแผนภูมิโดนัทเพื่อแสดงสัดส่วนของคะแนน และแผนภูมิแท่งที่แสดงน้ำหนักของเกณฑ์แต่ละข้อ เพื่อช่วยให้ผู้ใช้เข้าใจผลลัพธ์และปัจจัยที่ส่งผลต่อการคำนวณได้อย่างชัดเจน

บทที่ 4

การพัฒนาโปรแกรม

4.1 ชุดคำสั่งโปรแกรม

4.2.1 การนำเข้าไลบรารี

```
import streamlit as st
import pandas as pd
import base64
import matplotlib.pyplot as plt
import seaborn as sns
```

4.2.2 กำหนดฟังก์ชันคำนวณคะแนนถ่วงน้ำหนัก

```
def calculate_scores(matrix, weights):
    matrix["Weighted Score"] = matrix.iloc[:, :-1].dot(weights)
    return matrix
```

4.2.3 กำหนดเงื่อนไขการเปลี่ยนหน้า

```
# Page Navigation
if "page" not in st.session_state:
    st.session_state["page"] = "welcome"
if st.session_state["page"] == "welcome":
    welcome_page()
elif st.session_state["page"] == "category":
    category_page()
elif st.session_state["page"] == "alternative":
    alternative_page()
elif st.session_state["page"] == "criteria":
    add_criteria_page()
elif st.session_state["page"] == "input":
    input_page()
elif st.session_state["page"] == "results":
```

```
results_page()
```

4.2.4 กำหนดค่าเริ่มต้นของหมวดหมู่และเกณฑ์

```
categories = {
    "อาหาร": ["ราคา", "ความอึด", "ประโยชน์ต่อสุขภาพ"],
    "การแต่งกาย": ["โอกาส", "สไตล์", "ความสบายของเนื้อผ้า"],
    "กิจกรรมในวันหยุด": ["ความผ่อนคลาย", "ค่าใช้จ่าย", "ความสะดวกในการเดินทาง"],
    "การจัดการ": ["ต้นทุน", "คุณภาพ", "เวลา"],
    "ประกันภัย": ["ราคา", "ความคุ้มครอง", "บริการลูกค้า"],
    "การลงทุน": ["ระดับความเสี่ยง", "ผลตอบแทนที่คาดหวัง", "สภาพคล่อง"],
    "การตลาด": ["งบประมาณ", "คุณภาพเนื้อหา", "การมีส่วนร่วมของลูกค้า"],
    "หมวดอื่น": ["(กำหนดเอง)", "(กำหนดเอง)", "(กำหนดเอง)"],
}

criteria = []

# เก็บภาพพื้นหลังปุ่มสำหรับแต่ละหมวดหมู่
category_images = {
    "อาหาร": r"C:\Users\punyawee\OneDrive\เอกสาร\project_python\อาหาร.png",
    "การแต่งกาย": r"C:\Users\punyawee\OneDrive\เอกสาร\project_python\การแต่งกาย.png",
    "กิจกรรมในวันหยุด": r"C:\Users\punyawee\OneDrive\เอกสาร\project_python\กิจกรรมในวันหยุด.png",
    "การจัดการ": r"C:\Users\punyawee\OneDrive\เอกสาร\project_python\การจัดการ.png",
    "ประกันภัย": r"C:\Users\punyawee\OneDrive\เอกสาร\project_python\ประกันภัย.png",
    "การลงทุน": r"C:\Users\punyawee\OneDrive\เอกสาร\project_python\การลงทุน.png",
    "การตลาด": r"C:\Users\punyawee\OneDrive\เอกสาร\project_python\การตลาด.png",
    "หมวดอื่น": r"C:\Users\punyawee\OneDrive\เอกสาร\project_python\หมวดอื่น.png"
}
```

4.2.5 หน้าเริ่มต้น

```
# กดที่ปุ่ม “ลึกลง” ไปหน้าการเลือกหมวดหมู่
```

```
if st.button("ลั้งเล"):
    st.session_state["page"] = "category"
```

4.2.6 หน้าการเลือกหมวดหมู่

```
# กดที่ปุ่ม “กลับ” ไปหน้าเริ่มต้น
if st.button("กลับ"):
    st.session_state["page"] = "welcome"
# แสดงปุ่มสำหรับแต่ละหมวดหมู่
for category, sub_criteria in categories.items():
    # แปลงรูปภาพเป็น Base64
    image_path = category_images[category]
    image_base64 = image_to_base64(image_path)
    if image_base64:
        st.markdown("""
<style>
    button[data-testid="stBaseButton-secondary"]
        div[data-testid="stMarkdownContainer"] p {
            font-family: 'SaoChingCha';
            font-size: 18px;
            color: #4a4a4a;
        }
</style>
""", unsafe_allow_html=True)

    button_html = f"""
<div style="text-align: center; margin-bottom: 20px;">
    <button onclick="document.getElementById('{category}').click();" style="
        background-image: url('{image_base64}');
        background-size: cover;
        background-position: center;
        width: 800px;
        height: 250px;
```

```

        border: none;
        color: transparent;
        font-size: 18px;
        font-family: 'SaoChingCha', sans-serif;
        font-weight: bold;
        cursor: pointer;
        border-radius: 0px;
    ">
        {category}
    </button>
</div>
"""

st.markdown(button_html, unsafe_allow_html=True)
# แสดงปุ่มหมวดหมู่และเก็บค่า
if st.button(category, key=category):
    st.session_state["selected_category"] = category
    st.session_state["criteria"] = [
        {"name": crit, "weight": 0 } for crit in sub_criteria]
    st.session_state["page"] = "alternative"

```

4.2.7 หน้าเพิ่มตัวเลือกที่จะพิจารณา

```

# กดที่ปุ่ม “กลับ” แล้วไปหน้าการเลือกหมวดหมู่
if st.button("กลับ", key="back_alternative"):
    st.session_state["page"] = "category"
# ตรวจสอบว่ามีการเลือกหมวดหมู่
if "selected_category" in st.session_state:
    selected_category = st.session_state["selected_category"]
    st.write(f"<p style='font-family: \"SaoChingCha\"; color: #2D2D2D;'>คุณเลือกหมวดหมู่ :
        <strong>{selected_category}</strong></p>", unsafe_allow_html=True)
else:
    st.warning("<h3 style='font-family: \"SaoChingCha\"; color: #2D2D2D;'>ยังไม่มีทางเลือก
        หมวดหมู่</h3>", unsafe_allow_html=True)

```

```

# ตรวจสอบการกรอกตัวเลือก
if num_alternatives:
    alternatives = []
    for i in range(num_alternatives): # สร้างช่องให้ผู้กรอกตามจำนวนตัวเลือก
        st.markdown(f'<div class="stTextInput-label">ตัวเลือก {i+1} :</div>',
                    unsafe_allow_html=True)
        user_input = st.text_input("", key=f"input_{i}")
        st.markdown("")
        alternatives.append(user_input) # เพิ่มค่าเข้าlist
    st.session_state["alternatives"] = alternatives # เก็บตัวเลือกใน session_state

# ตรวจสอบข้อมูล และกดที่ปุ่ม “ถัดไป” ไปหน้าเพิ่มเกณฑ์
col1, col2, col3, col4, col5 = st.columns(5)
with col5:
    if st.button("ถัดไป", key="next_alternative"):
        # ตรวจสอบว่าชื่อตัวเลือกไม่ใช่ชื่อที่เป็นค่าเริ่มต้น และมีอย่างน้อย 2 ตัวเลือก
        valid_alternatives = [
            alt
            for alt in st.session_state["alternatives"]
            if alt.strip() and alt != f"Alternative {st.session_state['alternatives'].index(alt) + 1}"
        ]
        if len(valid_alternatives) >= 2:
            st.session_state["page"] = "criteria"
        else:
            st.markdown(f'<h6 style="font-family: \'SaoChingCha\'; color: #D32F2F;
                        background-color: #FFEBEE; padding: 10px;">คุณต้องกรอก
                        อย่างน้อย 2 ตัวเลือก</h6>', unsafe_allow_html=True)

```

4.2.8 หน้าการเพิ่มเกณฑ์ที่ใช้ในการพิจารณาตัวเลือก

```

# กดที่ปุ่ม “กลับ” แล้วไปหน้าเพิ่มตัวเลือก
if st.button("กลับ", key="back_criteria"):

```

```

st.session_state["page"] = "alternative"

# ตรวจสอบการกรอกเกณฑ์

# ปุ่มลบเกณฑ์
col1, col2, col3, col4, col5 = st.columns(5)
with col5:
    if st.button(f"ลบเกณฑ์ที่ {i + 1}", key=f"delete_{i}"):
        st.session_state["criteria"] = [c for j, c in enumerate(st.session_state["criteria"]) if
                                         j != i]

# ปุ่มเพิ่มเกณฑ์
col1, col2, col3, col4, col5 = st.columns(5)
with col3:
    if st.button("เพิ่ม"):
        # ตรวจสอบการกรอกเกณฑ์และมีแล้วหรือไม่
        if (new_criteria_name and new_criteria_name not in [c["name"]
                                                             for c in st.session_state["criteria"]]) :
            st.markdown(
                f'<h6 style="font-family: \'SaoChingCha\'; color: #388E3C;
                    background-color: #E8F5E9; padding: 10px; white-space:
                    nowrap; display: inline-block; >เพิ่มเกณฑ์
                    <strong>{new_criteria_name}</strong> เรียบร้อยแล้ว</h6>',
                unsafe_allow_html=True)
            st.session_state["criteria"].append({"name": new_criteria_name})

        elif new_criteria_name in [c["name"] for c in st.session_state["criteria"]]:
            st.markdown(
                f'<h6 style="font-family: \'SaoChingCha\'; color: #F57C00; background-
                    color: #FFF3E0; padding: 10px; white-space: nowrap; display:
                    inline-block;">เกณฑ์ \'{new_criteria_name}\'' มีอยู่แล้ว</h6>',
                unsafe_allow_html=True,)

```


else:

```
st.markdown(f'<h6 style="font-family: \'SaoChingCha\'; color: #D32F2F;
background-color: #FFEBEE; padding: 10px; display: inline-
block ; white-space:nowrap;">กรุณารอกชื่อเกณฑ์</h6>',
unsafe_allow_html=True)
```

ตรวจสอบข้อมูล และกดที่ปุ่ม “ถัดไป” ไปหน้ากำหนดน้ำหนักและคะแนน

```
bottom_col1, bottom_col2, bottom_col3, bottom_col4, bottom_col5 = st.columns(5)
with bottom_col5:
```

```
if st.button("ถัดไป", key="next_criteria"):
```

```
if len(st.session_state["criteria"]) >= 2:
```

```
st.session_state["page"] = "input"
```

else:

```
st.markdown(
```

```
f'<h6 style="font-family: \'SaoChingCha\'; color: #D32F2F; background-color:
#FFEBEE; padding: 10px; display: inline-block ; white-space:nowrap;">คุณ
ต้องกรอกเกณฑ์อย่างน้อย 2 เกณฑ์</h6>',unsafe_allow_html=True)
```

4.2.9 หน้ากำหนดน้ำหนักและคะแนน

กดที่ปุ่ม “กลับ” แล้วไปหน้ากำหนดเกณฑ์

```
if st.button("กลับ", key="back_input"):
```

```
st.session_state["page"] = "criteria"
```

ลูป for รับค่าน้ำหนักของแต่ละเกณฑ์และคะแนนสำหรับแต่ละตัวเลือก

```
for criterion in st.session_state["criteria"]: #แสดงชื่อเกณฑ์ทั้งหมด
```

```
st.markdown(f'<div class="stTextInput-label"> น้ำหนักสำหรับเกณฑ์
\{criterion["name"]}\:</div>', unsafe_allow_html=True)
```

number_input รับค่าน้ำหนักของเกณฑ์

```
criterion["weight"] = st.number_input( "", min_value=0,
```

```
max_value=100, step=1, value=criterion.get("weight", 0),
```

```
key=f'weight_{criterion["name"]}') # ใช้ key เฉพาะสำหรับแต่ละเกณฑ์)
```

```
total_weight += criterion["weight"]
# ตรวจสอบน้ำหนักกรวมต้องเท่ากับ100%
if (total_weight != 100):
    if (total_weight < 100): #ผลต่างจาก100 เพื่อนำไปแสดงว่าต้องเพิ่มหรือลด
        diff = 100 - total_weight
        message = f"น้ำหนักกรวมคือ {total_weight}%&nbsp;&nbsp;&nbsp;(เพิ่มอีก {diff}%)\"
    elif (total_weight > 100):
        diff = total_weight - 100
        message = f"น้ำหนักกรวมคือ {total_weight}%&nbsp;&nbsp;&nbsp;(ลดอีก {diff}%)\"
    st.markdown(
        f<h6 style=\"font-family: \'SaoChingCha\'; color: #D32F2F; background-color:
            #FFEBEE; padding: 10px; \">> {message} </h6>',
        unsafe_allow_html=True)
else:
    st.markdown(f<h6 style=\"font-family: \'SaoChingCha\'; color: #388E3C; background-
        color: #E8F5E9; padding: 10px;\">>น้ำหนักกรวมถูกต้อง</h6>',
        unsafe_allow_html=True)

#อุปแสดงทุกเกณฑ์สำหรับแต่ละตัวเลือก
for i, criterion in enumerate(st.session_state[\"criteria\"]):
    st.markdown(f<h5 style='font-size: 22px; font-family: \'SaoChingCha\\\", sans-serif;
        color: #2D2D2D;;'>{criterion['name']} (Weight: {criterion['weight']})%</h5>',
        unsafe_allow_html=True)
    for alt in alternatives: #ทีละตัวเลือก
        st.markdown(f<div class= \"stTextInput-label\">{criterion[\"name\"]} : {alt}</div>',
            unsafe_allow_html=True)
    #แถบเลื่อนคะแนน
    df.loc[df[\"Alternative\"] == alt, criterion[\"name\"]] = st.slider(
        \"\", min_value=1, max_value=10, value=3, step=1,
        format= \" %d\", help= \"1 = พอใช้, 10 = ดีมาก\",
        key=f\"{alt}_{criterion['name']}\"
    )
```

```
# ตรวจสอบข้อมูล และกดที่ปุ่ม “ถัดไป” แล้วไปหน้าผลลัพธ์
col1, col2, col3, col4, col5= st.columns(5)
with col5:
    if st.button("ถัดไป", key="next_col2"):
        if total_weight == 100:
            st.session_state["df"] = df
            st.session_state["weights"] = [criterion["weight"] / 100 for criterion in
                                            st.session_state["criteria"]]
            st.session_state["page"] = "results"
        else:
            st.markdown(
                f'<h6 style="font-family: \'SaoChingCha\'; color: #D32F2F; background-color:
                #FFEBEE; padding: 10px; "> กรุณากรอกน้ำหนักให้ถูกต้อง </h6>',
                unsafe_allow_html=True )
```

4.2.10 หน้าผลลัพธ์

```
#ฟังก์ชันที่คำนวณคะแนนถ่วงน้ำหนักใน dataframe
df = calculate_scores(df, weights)
#หาตำแหน่งของแถวในคอลัมน์ Weighted Scoreที่มีค่ามากที่สุด
best_alternative = df.loc[df["Weighted Score"].idxmax()]["Alternative"]
#แสดง tab ของผลการคำนวณ : df, donut chart, bar chart
tab1, tab2, tab3 = st.tabs(["ผลการคำนวณ", "จัดลำดับตามตัวเลือก", "จัดลำดับตามเกณฑ์"])
```

บทที่ 5

สรุปผล

ในการดำเนินการทำโครงการเรื่องระบบช่วยตัดสินใจโดยใช้วิธีเมทริกซ์ตัดสินใจนี้มีวัตถุประสงค์เพื่อออกแบบระบบที่ช่วยในการตัดสินใจโดยใช้ Decision Matrix และ SWOT Analysis เพื่อช่วยให้ผู้ใช้งานสามารถเปรียบเทียบตัวเลือกต่าง ๆ อย่างมีระบบ โดยพัฒนาระบบที่สามารถใช้วิเคราะห์และประเมินตัวเลือกต่าง ๆ จากเกณฑ์ที่เกี่ยวข้องได้อย่างมีประสิทธิภาพ โดยทางคณะผู้จัดทำได้ใช้ภาษาไพทอน (Python) และไลบรารี Streamlit ในการพัฒนาระบบที่รองรับการคำนวณและแสดงผลตามการใช้ Decision Matrix และ SWOT Analysis โดยอาศัยความรู้จากทฤษฎีและการประยุกต์ใช้งานในชีวิตจริง

5.1 สรุปผล

ผลการทำงานพบว่าระบบที่พัฒนาขึ้นช่วยให้ผู้ใช้งานสามารถตัดสินใจได้ดีขึ้น โดยการมองเห็นภาพรวมของปัจจัยต่าง ๆ ที่ส่งผลต่อการตัดสินใจ รวมถึงช่วยลดความลังเลและเพิ่มความมั่นใจในการเลือกตัวเลือกที่เหมาะสม โดยผู้ใช้งานสามารถนำข้อมูลที่ได้ไปใช้ในสถานการณ์จริงได้อย่างมีประสิทธิภาพ

5.2 อภิปรายผล

ผลการทำงานพบว่าระบบที่พัฒนาขึ้นสามารถช่วยให้ผู้ใช้งานเปรียบเทียบตัวเลือกต่าง ๆ ได้อย่างมีระบบ โดยการใช้ Decision Matrix ช่วยให้ผู้ใช้งานสามารถพิจารณาปัจจัยต่าง ๆ ได้อย่างชัดเจนและมีเหตุผล โดยตรงกับวัตถุประสงค์ที่ต้องการให้ผู้ใช้งานสามารถตัดสินใจได้ดีขึ้นและลดความลังเลในการเลือก นอกจากนี้ การนำ SWOT Analysis มาใช้ในการประเมินปัจจัยภายในและภายนอกที่มีผลต่อการตัดสินใจ ช่วยเสริมการวิเคราะห์และให้มุมมองที่ครอบคลุมมากขึ้น ซึ่งเป็นไปตามหลักการของการวิเคราะห์สิ่งแวดล้อมในการตัดสินใจ

เมื่อเปรียบเทียบกับงานวิจัยที่เกี่ยวข้อง เช่น การใช้ Decision Matrix ในการตัดสินใจในองค์กร พบว่าผลลัพธ์จากการพัฒนาโปรแกรมครั้งนี้สอดคล้องกับแนวทางที่หลายงานวิจัยเคยศึกษา โดยเฉพาะในเรื่องของการช่วยให้ผู้ใช้งานสามารถตัดสินใจได้อย่างมีประสิทธิภาพและมีความมั่นใจมากขึ้น

สรุปได้ว่าระบบที่พัฒนาขึ้นในโครงการนี้สามารถตอบโจทย์วัตถุประสงค์ที่ตั้งไว้ได้ดี โดยสามารถช่วยให้ผู้ใช้งานตัดสินใจได้ดีขึ้นและประยุกต์ใช้ในชีวิตจริงได้อย่างมีประสิทธิภาพ

5.3 ข้อเสนอแนะ

จากการทำโครงการในครั้งนี้ คณะผู้จัดทำพบปัญหาและข้อเสนอแนะที่เป็นประโยชน์ดังต่อไปนี้

5.3.1 ข้อเสนอแนะในการนำไปใช้จริง

ควรนำระบบที่พัฒนาขึ้นไปใช้ในองค์กรเพื่อช่วยในการตัดสินใจทางธุรกิจ เช่น การเลือกกลยุทธ์ทางการตลาดหรือการลงทุน โดยใช้ Decision Matrix และ SWOT Analysis เพื่อมองเห็นภาพรวมของปัจจัยต่าง ๆ ที่ส่งผลต่อการตัดสินใจ นอกจากนี้ผู้ใช้งานทั่วไปสามารถใช้ระบบในการช่วยตัดสินใจในชีวิตประจำวัน เช่น การเลือกซื้อสินค้าหรือการวางแผนกิจกรรมต่าง ๆ

5.3.2 ข้อเสนอแนะในการพัฒนาในอนาคต

ควรพัฒนาระบบโดยการนำ AI และ Machine Learning มาช่วยในการปรับแต่งเกณฑ์การตัดสินใจให้เหมาะสมและยืดหยุ่นมากขึ้น โดย AI สามารถเรียนรู้จากข้อมูลในอดีตและปรับการตัดสินใจให้แม่นยำและมีประสิทธิภาพยิ่งขึ้น นอกจากนี้ยังสามารถใช้ AI ในการประเมินผลลัพธ์จากการจำลองสถานการณ์ และคาดการณ์ผลในอนาคตที่อาจเกิดขึ้น

ควรเพิ่มฟังก์ชันการใช้งานให้มุ่งเน้นไปทางธุรกิจมากขึ้นโดยใช้ทฤษฎีการตัดสินใจเข้ามาช่วย เช่น Analytic Hierarchy Process (AHP) หรือ Multi-Criteria Decision Analysis (MCDA) มาใช้ร่วมกับการจำลองสถานการณ์ และ BCG Matrix เพื่อเพิ่มความหลากหลายและความเที่ยงตรงในการตัดสินใจ ซึ่งจะช่วยให้ผลลัพธ์ที่ได้มีความครอบคลุมและถูกต้องมากยิ่งขึ้น

บรรณานุกรม

- กษิดิศ สดางค์มงคล. (2565). *Streamlit เริ่มต้นสร้าง Web App ง่ายๆด้วยภาษา Python ได้แค่ 3 บรรทัด*. สืบค้น 25 ธันวาคม 2567, จาก <https://datarockie.com/blog/python-streamlit-web-app/>
- จักรชัย อินลายศ. (2566). *ความหมายของอาหารและโภชนาการ*. สืบค้น 5 มกราคม 2568, จาก https://elchm.ssru.ac.th/sarita_pu/pluginfile.php/63/mod_resource/content/1/%E0%B
- นิรุบล สุขวณิช. (2565). *พัฒนาทักษะการตัดสินใจ (Decision-making Skill) เพื่อสุขภาพจิตที่ดีขึ้น*. สืบค้น 26 ธันวาคม 2567, จาก <https://www.istrong.co/single-post/decision-makingskill?srsltid=AfmBOornSngyAMc4wmVGbfsphvllB1KcbAjajQoskOj7C2HAZW1N4ip>
- ปานชาติ มิตรกุล. (2564). *Pandas – Library อันดับ 1 สำหรับ Data Analysis*. สืบค้น 31 ธันวาคม 2567, จาก <https://blog.skooldio.com/what-is-pandas/>
- สถานเอกอัครราชทูต ณ กรุงเฮลซิงกิ. (2562). *การวิเคราะห์ SWOT (SWOT Analysis)*. สืบค้น 30 ธันวาคม 2567, จาก <https://helsinki.thaiembassy.org/th/page/65202-%E0%B8%81%E0%B8%B2%E0%B8%A3%E0%B8%A7%E0%B8%B4%E0%B9%80%E0%>
- Amonnut, T. (2023). *Matplotlib คืออะไร? ใช้พล็อตอย่างไรใน Python?*. Retrieved 9 January 2025, from <https://medium.com/@athammatadatrakoon/matplotlib-%E0%B8%84>
- DATA TH. (2015). *เครื่องมือทำ Data Visualization แบบเขียนโปรแกรม ใช้ตัวไหนดี (RShiny, Ggplot, Seaborn, Matplotlib, D3.js)*. Retrieved 9 January 2025, from <https://blog.datath.com/data-visualization-library/#:~:text=Seaborn%20%E0%B9%80>
- FWD. (2022). *ทำไมเราควรวางแผนประกันภัย เลือกแบบไหนถึงตอบโจทย์?*. Retrieved 1 January 2025, from <https://www.fwd.co.th/th/article/health/choosing-the-right-insurance-plan/>
- HREX.asia. (2019). *Project Management คืออะไร สามารถประยุกต์ใช้กับองค์กรได้อย่างไรบ้าง*. Retrieved 30 December 2024, from <https://th.hrnote.asia/orgdevelopment/th-190215whatismangement/>
- KTC. (2024). *การลงทุนหมายถึงอะไร มีกี่ประเภท มีอะไรบ้าง สำคัญอย่างไร*. Retrieved 7 January 2025, from <https://www.ktc.co.th/article/knowledge/stocks/what-is-investment>
- Normthing Consult. (2020). *ตัดสินใจด้วย เมตริกซ์การตัดสินใจ decision matrix*. Retrieved 31 December 2024, from <https://www.normthing.com/2020/12/13/decision-making-matrix/>

Phyblas. (2016). *numpy & matplotlib เบื้องต้น บทที่ ๕: การวาดกราฟเบื้องต้น*. Retrieved 9 January 2025, from <https://phyblas.hinaboshi.com/numa05>

Rocket. (2022). *Digital Marketing คืออะไร? และทำไมถึงสำคัญกับธุรกิจของคุณมากกว่าเดิม?*. Retrieved 8 January 2025, from <https://www.rocket.in.th/blog/what-is-digital-marketing/#:~:text=Digital%20Marketing>

Service Design Thailand. (2017). *Pugh Matrix*. Retrieved 8 January 2025, from <https://tcdc.groov.asia/method/pugh-matrix-full.html>

Stuart, B. (2016). *The Systems Thinking Tool Box*. Retrieved 8 January 2025, from <https://www.burgehugheswalsh.co.uk/Uploaded/1/Documents/Decision-Matrix-Tool-v2.pdf>

Wannaphong, P. (2015). *เชื่อมต่อกับระบบปฏิบัติการด้วยไลบรารี os ใน Python*. Retrieved 8 January 2025, from <https://python3.wannaphong.com/2015/04/os-python.html?m=1>