การจำแนกและวิเคราะห์หมวดหมู่การใช้งาน สำหรับลูกค้าของแพลตฟอร์ม Text to Speech

ANALYSIS AND CLASSIFICATION OF CUSTOMER TEXT FROM TEXT-TO-SPEECH PLATFORM

วิชยุตม์ แท่นจิรวัฒนา

สหกิจศึกษานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิทยาศาสตร์บัณฑิต (สถิติประยุกต์)
ภาควิชาสถิติ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2566

ANALYSIS AND CLASSIFICATION OF CUSTOMER TEXT FROM TEXT-TO-SPEECH PLATFORM

WICHAYUT THANJIRAWATHANA

A COOPERATIVE EDUCATION SUBMITTED IN PARTIAL
FULFILLMENT OF THE REQUIREMENT FOR
THE DEGREE OF BACHELOR OF SCIENCE (APPLIED STATISTICS)
DEPARTMENT OF STATISTICS, SCHOOL OF SCIENCE
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2023

หัวข้อสหกิจศึกษา	การจำแนกและวิเคราะห์หมวดหมู่การใช้งานสำหรับลูกค้าของแพลตฟอร์ม text to speech
ชื่อนักศึกษา	นายวิชยุตม์ แท่นจิรวัฒนา รหัสนักศึกษา 63050663
ปริญญา	วิทยาศาสตร์บัณฑิต (สถิติประยุกต์)
ภาควิชา	สถิติ
คณะ	วิทยาศาสตร์
มหาวิทยาลัย	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง (สจล.)
ปีการศึกษา	2566
อาจารย์ที่ปรึกษา	ผศ.ดร.ยุวดี กล่อมวิเศษ

คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง (สจล.) อนุมัติให้สหกิจศึกษานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตร์บัณฑิต (สถิติประยุกต์) ประจำปีการศึกษา 2566

คณะกรรมการสอบ	ลายมือชื่อ
ผศ.ดร.สิทธิชัย เจริญเศรษฐศิลป์ ประธานกรรมการ	
คุณสุทธิดา ลือชัย กรรมการ	
ผศ.ดร.ยุวดี กล่อมวิเศษ กรรมการและอาจารย์ที่ปรึกษา	

หัวข้อสหกิจศึกษา	การจำแนกและวิเคราะห์หมวดหมู่การใช้งานสำหรับลูกค้าของแพลตฟอร์ม Text to Speech
ชื่อนักศึกษา	นายวิชยุตม์ แท่นจิรวัฒนา รหัสนักศึกษา 63050663
ปริญญา	วิทยาศาสตร์บัณฑิต (สถิติประยุกต์)
ภาควิชา	สถิติ
คณะ	วิทยาศาสตร์
มหาวิทยาลัย	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง (สจล.)
ปีการศึกษา	2566
อาจารย์ที่ปรึกษา	ผศ.ดร.ยุวดี กล่อมวิเศษ

บทคัดย่อ

ในแพลตฟอร์ม text to speech นั้น การที่ลูกค้าเข้ามาใช้งานแต่ละคนก็มีจุดประสงค์ที่มาใช้งานแตกต่างกัน เช่น ลูกค้าจะนำไปใช้สำหรับรีวิวสินค้า ลูกค้าอาจจะต้องการเสียงบอทที่มีความน่าดึงดูด น่าสนใจ ดังนั้นจึงต้องมีเครื่องมือที่ช่วยในการจำแนกหมวดหมู่การใช้งาน เพื่อนำไปพัฒนาแพลตฟอร์ม text to speech ให้ตอบโจทย์ความต้องการการใช้งานของลูกค้าแต่ละหมวดหมู่ โดยผู้วิจัยได้ใช้ข้อมูลที่เป็นข้อความที่ลูกค้าเข้ามาใช้งานในแพลตฟอร์ม text to speech ตั้งแต่วันที่ 1 มกราคม พ.ศ.2566 ถึง วันที่ 10 มิถุนายน พ.ศ.2566 จำนวน 250,000 ข้อความ และทำการแบ่งข้อมูลออกเป็น 2 ส่วนคือ ชุดข้อมูลที่ใช้ในการเรียนรู้ 80% และข้อมูลที่ใช้ในการทดสอบ 20% จากนั้นใช้วิธีการประมวลผลภาษาธรรมชาติ คือ การแปลงเชิงปริมาณด้วยเทคนิคการคัดแยกคำตามความสำคัญ เพื่อนำข้อมูลมาสร้างแบบจำลอง 4 แบบ ได้แก่ แบบจำลองแรนดอมฟอเรสต์, แบบจำลองการถดถอยแบบโลจิสติก, แบบจำลองซัพพอร์ตเวกเตอร์แมชชีน และแบบจำลองนาอีฟเบย์ ในการเปรียบเทียบประสิทธิภาพของแบบจำลองโดยพิจารณาจากค่า Macro - F1-Score เป็นอันดับแรกในการวัดผล เมื่อพิจารณาจากผลลัพธ์ของแบบจำลองในชุดข้อมูลที่ใช้ในการทดสอบ พบว่าแบบจำลองซัพพอร์ตเวกเตอร์แมชชีน มีค่า F1 – Score สูงที่สุด จึงเป็นแบบจำลองที่เหมาะสมที่สุดสำหรับการจำแนกหมวดหมู่ข้อความของลูกค้าที่เข้ามาใช้งานแพลตฟอร์ม text to speech

คำสำคัญ : การประมวลผลภาษาธรรมชาติ, เทคนิคการคัดแยกคำตามความสำคัญ, แรนดอมฟอเรสต์, การถดถอยโลจิสติก, ซัพพอร์ตเวกเตอร์แมชชีน, นาอีฟเบย์

Title	Analysis And Classification Of Customer Text From Text-To-Speech Platform
Students	Mr. Wichayut Thanjirawathana Student ID 63050663
Degree	Bachelor of science (Applied Statistics)
Department	Statistics
School	Science
University	King Mongkut's Institute of Technology Ladkrabang (KMITL)
Academic Year	2023
Advisor	Asst. Prof. Dr. Yuwadee Klomwises

Abstract

In the text to speech platform, each customer who comes to use it has different purposes. For example, some users may utilize the platform for product reviews, desiring an engaging and attractive bot voice. Therefore, it is necessary to create a tool that is capable of categorizing their usage patterns. The development of a text-to-speech platform that can meet the requirements of each customer group is important. We use text data that customers use on the platform from January 1, 2023 to June 10, 2023, a total of 250,000 messages. We divided the data into 2 parts: 80% for the training dataset and 20% for the testing data. Then, natural language processing method is applied, called TF-IDF for text representation. In addition, four classification models are utilized for this purpose, which are Random Forest model, Logistic Regression model, Support Vector Machine model and Naïve Bayes model. The evaluation of model performance is primarily based on the F1-Score value. As a result, we found that Support Vector Machine model achieves the highest F1 – Score. Thus, it can be concluded that Support Vector Machine is the most suitable model for text classification of customer utilizing the text-to-speech platform.

Keywords : Natural Language Processing; NLP, TF-IDF, Random Forest, Logistic Regression, Support Vector Machine, Naïve Bayes

กิตติกรรมประกาศ

โครงงานสหกิจเล่มนี้สำเร็จไปได้โดยดี เนื่องจากผู้วิจัยได้รับความอนุเคราะห์และความกรุณาจากคณะอาจารย์และบุคคลผู้มีพระคุณหลายท่าน ดังรายนามต่อไปนี้

ขอขอบพระคุณ ผศ.ดร.ยุวดี กล่อมวิเศษ ที่เป็นอาจารย์ที่ปรึกษาสหกิจศึกษาที่ได้ช่วยแนะนำและให้คำปรึกษา รวมถึงเสนอแนะแนวทางการแก้ไขปัญหา ตลอดจนการทำงานวิจัยสำเร็จ รวมทั้งตรวจทานแก้ไขสหกิจศึกษาเล่มนี้ให้สมบูรณ์

ขอขอบพระคุณ ผศ.ดร.สิทธิชัย เจริญเศรษฐศิลป์ ที่กรุณาเป็นกรรมการในการสอบสหกิจศึกษา อีกทั้งยังให้ความรู้ คำแนะนำ และช่วยตรวจสอบแก้ไขให้สหกิจเล่มนี้ออกมาสมบูรณ์

ขอขอบพระคุณทางบริษัทเป็นอย่างสูงที่มอบโอกาสให้เข้ามาฝึกงาน ได้ประสบการณ์ในการทำงาน และอนุญาติให้นำข้อมูลมาใช้ในการทำสหกิจครั้งนี้

ขอขอบพระคุณ นางสาวสุทธิดา ลือชัย ผู้ที่ได้มอบหมายหัวข้อการทำสหกิจศึกษาในครั้งนี้ และพี่ๆ ในทีมที่คอยให้คำแนะนำ ให้ความรู้เกี่ยวกับวิธีการทำงาน ประสบการณ์ในการทำงาน รวมทั้งช่วยเหลือด้านต่าง ๆ และให้คำปรึกษาในการทำสหกิจครั้งนี้จนสามารถสำเร็จลุล่วงไปได้ด้วยดี

สุดท้ายนี้ผู้วิจัยขอขอบคุณครอบครัวของผู้วิจัย รวมทั้งเพื่อน ๆ พี่ ๆ ภาควิชาสถิติ และบุคคลที่ไม่ได้กล่าวถึงมา ณ ที่นี้ที่ให้ความช่วยเหลือ การสนับสนุน และกำลังใจตลอดการทำสหกิจศึกษานี้ให้สำเร็จไปด้วยดี

วิชยุตม์ แท่นจิรวัฒนา

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ก
บทคัดย่อภาษาอังกฤษ	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง
สารบัญตาราง	ช
สารบัญรูป	ฌ
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์ของงานวิจัย	2
1.3 ขอบเขตงานวิจัย	2
1.4 ประโยชน์ที่คาดว่าจะได้รับ	2
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง	3
2.1 การเรียนรู้ของเครื่อง (Machine Learning)	3
2.1.1 การเรียนรู้แบบมีผู้สอน (Supervised Learning)	4
2.1.2 การเรียนรู้โดยไม่มีผู้สอน (Unsupervised Learning)	4
2.1.3 การเรียนรู้แบบเสริมกำลัง (Reinforcement Learning)	4
2.2 การประมวลผลภาษาธรรมชาติ (NLP)	4
2.2.1 วิวัฒนาการของการประมวลผลภาษาธรรมชาติ	5
2.2.2 ความสำคัญของการประมวลผลภาษาธรรมชาติ	5
2.2.3 กระบวนการทำงานของการประมวลผลภาษาธรรมชาติ	6
2.2.3.1 การเตรียมข้อมูล (Data Preprocessing)	6
2.2.3.2 การแปลงเชิงปริมาณ (Text Representation)	6
2.2.3.3 เทคนิคการคัดแยกคำตามความสำคัญ (Term Frequency – Inverse Document Frequency : TF-IDF)	6
2.3 การจำแนกประเภท	7
2.3.1 แบบจำลองแรนดอมฟอเรสต์ (Random Forest)	8
2.3.2 แบบจำลองการถดถอยแบบโลจิสติก (Logistic Regression)	8
2.3.3 แบบจำลองซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machine; SVM)	11

สารบัญ (ต่อ)

	หน้า
2.3.4 แบบจำลองนาอีฟเบย์ (Naïve Bayes)	14
2.4 การวัดประสิทธิภาพของแบบจำลอง (Evaluation)	15
2.4.1 เมทริกซ์ความสับสน (Confusion Matrix)	15
2.4.2 ค่าความแม่นยำ (Accuracy)	16
2.4.3 ค่าความเที่ยง (Precision)	16
2.4.4 ค่าระลึก (Recall)	17
2.4.5 ค่าความถ่วงดุล (F1–Score)	17
2.4.6 ค่าเฉลี่ยมาโคร (Macro Average)	17
2.5 ปัญหาโอเวอร์ฟิตติง (Overfitting)	18
2.6 งานวิจัยที่เกี่ยวข้อง	18
บทที่ 3 วิธีดำเนินงานวิจัย	21
3.1 ขั้นตอนการดำเนินงาน	21
3.2 การจัดเตรียมข้อมูล	21
3.3 การตรวจสอบและทำความสะอาดข้อมูล	22
3.3.1 การทำความสะอาดข้อความ (Text Cleaning)	22
3.3.2 การตัดคำ (Tokenization) และการลบคำฟุ่มเฟือย (Stop Words)	23
3.3.3 กำหนดผลหมวดหมู่ของข้อความ	24
3.3.4 การสกัดคุณลักษณะ (Feature Extraction)	25
3.4 การออกแบบคุณลักษณะและแบบจำลอง	26
3.5 การเปรียบเทียบผลพยากรณ์ของแบบจำลอง	26
3.6 เครื่องมือที่ใช้ในการวิจัย	26
บทที่ 4 ผลการวิจัยและการอภิปรายผล	27
4.1 ผลการทดสอบประสิทธิภาพของแบบจำลองแรนดอมฟอเรสต์ (Random Forest)	27
4.2 ผลการทดสอบประสิทธิภาพของแบบจำลองการถดถอยแบบโลจิสติก (Logistic Regression)	29
4.3 ผลการทดสอบประสิทธิภาพของแบบจำลองซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machine; SVM)	30
4.4 ผลการทดสอบประสิทธิภาพของแบบจำลองนาอีฟเบย์ (Naive Bayes)	32
4.5 ผลการทดสอบประสิทธิภาพของแบบจำลองด้วยข้อมูลชุดเรียนรู้	33
4.6 การเปรียบเทียบประสิทธิภาพแบบจำลองทั้งหมด	38
4.7 การนำแบบจำลองไปใช้งาน	39

สารบัญ (ต่อ)

	หน้า
บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ	41
5.1 สรุปผลการวิจัย	41
5.2 ข้อเสนอแนะ	42
เอกสารอ้างอิง	43
ภาคผนวก	46
ภาคผนวก ก	47

สารบัญตาราง

ตารางที่		หน้า
2.1	เมทริกซ์ความสับสน (Confusion Matrix) กรณีพิจารณาจากหมวดหมู่ A	19
3.1	ตัวอย่างการทำความสะอาดข้อมูล	26
3.2	ตัวอย่างการตัดคำและลบคำฟุ่มเฟือย	28
3.3	ตัวอย่าง Dictionary	28
3.4	ตัวอย่างการกำหนดหมวดหมู่ของข้อความ	29
3.5	ไลบรารี (Library) ที่จำเป็นต่อการวิเคราะห์	31
4.1	เมทริกซ์ความสับสน (Confusion Matrix) ของการแปลงเชิงปริมาณ TF-IDF กับแบบจำลองแรนดอมฟอเรสต์	33
4.2	ประสิทธิภาพในการทำนายของการแปลงเชิงปริมาณ TF-IDF กับแบบจำลองแรน ดอมฟอเรสต์	34
4.3	เมทริกซ์ความสับสน (Confusion Matrix) ของการแปลงเชิงปริมาณ TF-IDF กับแบบจำลองการถดถอยแบบโลจิสติก	35
4.4	ประสิทธิภาพในการทำนายของการแปลงเชิงปริมาณ TF-IDF กับแบบจำลองการถดถอยแบบโลจิสติก	36
4.5	เมทริกซ์ความสับสน (Confusion Matrix) ของการแปลงเชิงปริมาณ TF-IDF กับแบบจำลองซัพพอร์ตเวกเตอร์แมชชีน	37
4.6	ประสิทธิภาพในการทำนายของการแปลงเชิงปริมาณ TF-IDF กับแบบจำลองซัพพอร์ตเวกเตอร์แมชชีน	38
4.7	เมทริกซ์ความสับสน (Confusion Matrix) ของการแปลงเชิงปริมาณ TF-IDF กับแบบจำลองนาอีฟเบย์	39
4.8	ประสิทธิภาพในการทำนายของการแปลงเชิงปริมาณ TF-IDF กับแบบจำลองนาอีฟเบย์	40
4.9	เมทริกซ์ความสับสน (Confusion Matrix) ของการแปลงเชิงปริมาณ TF-IDF กับแบบจำลองแรนดอมฟอเรสต์ด้วยข้อมูลชุดเรียนรู้	41
4.10	ประสิทธิภาพในการทำนายของการแปลงเชิงปริมาณ TF-IDF กับแบบจำลองแรน ดอมฟอเรสต์ด้วยข้อมูลชุดเรียนรู้	41
4.11	เมทริกซ์ความสับสน (Confusion Matrix) ของการแปลงเชิงปริมาณ TF-IDF กับแบบจำลองการถดถอยแบบโลจิสติกด้วยข้อมูลชุดเรียนรู้	42
4.12	ประสิทธิภาพในการทำนายของการแปลงเชิงปริมาณ TF-IDF กับแบบจำลองการถดถอยแบบโลจิสติกด้วยข้อมูลชุดเรียนรู้	42

สารบัญตาราง (ต่อ)

ตารางที่		หน้า
4.13	เมทริกซ์ความสับสน (Confusion Matrix) ของการแปลงเชิงปริมาณ TF-IDF กับแบบจำลองซัพพอร์ตเวกเตอร์แมชชีนด้วยข้อมูลชุดเรียนรู้	43
4.14	ประสิทธิภาพในการทำนายของการแปลงเชิงปริมาณ TF-IDF กับแบบจำลองซัพพอร์ตเวกเตอร์แมชชีนด้วยข้อมูลชุดเรียนรู้	43
4.15	เมทริกซ์ความสับสน (Confusion Matrix) ของการแปลงเชิงปริมาณ TF-IDF กับแบบจำลองนาอีฟเบย์ด้วยข้อมูลชุดเรียนรู้	44
4.16	ประสิทธิภาพในการทำนายของการแปลงเชิงปริมาณ TF-IDF กับแบบจำลองนาอีฟเบย์ด้วยข้อมูลชุดเรียนรู้	44
4.17	เปรียบเทียบประสิทธิภาพของแบบจำลองทั้ง 4 แบบ กับข้อมูลชุดเรียนรู้และข้อมูลชุดทดสอบ ด้วยค่าความถ่วงดุลแบบมาโคร (Macro - F1-Score)	45
4.18	เปรียบเทียบประสิทธิภาพของแบบจำลองทั้ง 4 แบบ กับข้อมูลชุดเรียนรู้และข้อมูลชุดทดสอบ ด้วยค่าความแม่นยำ (Accuracy)	45
4.19	เปรียบเทียบประสิทธิภาพของแบบจำลองทั้ง 4 แบบ ด้วยค่าความถ่วงดุลแบบมาโคร (Macro - F1-Score) และค่าความแม่นยำ (Accuracy)	46
5.1	สรุปผลการเปรียบเทียบประสิทธิภาพของแบบจำลองทั้ง 4 แบบ ด้วยค่าความถ่วงดุลแบบมาโคร (Macro - F1-Score)	48
5.2	สรุปผลการเปรียบเทียบประสิทธิภาพของแบบจำลองทั้ง 4 แบบ ด้วยค่าความแม่นยำ (Accuracy)	48

สารบัญรูป

รูปที่		หน้า
2.1	ตัวอย่างการสร้างคุณลักษณะของข้อความ TF-IDF	8
2.2	กระบวนการทำงานของวิธีแรนดอมฟอเรสต์	9
2.3	ฟังก์ชันโลจิสติก (Logistic Function)	10
2.4	ซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machine; SVM)	13
2.5	ตัวอย่างตาราง Confusion Matrix ขนาด 2x2	17
3.1	ขั้นตอนการดำเนินงาน	25
3.2	แสดง Python Code สำหรับการทำความสะอาดข้อมูล	26
3.3	แสดง Python Code สำหรับการตัดคำและลบคำฟุ่มเฟือย	27
3.4	แสดง Python Code การสกัดคุณลักษณะ	29
4.1	กราฟแสดงความถี่ของหมวดหมู่การใช้งานของลูกค้า	47

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

Text to Speech คือ เทคโนโลยีช่วยเหลือประเภทหนึ่งที่ใช้ในการอ่านออกเสียงข้อความที่ผู้ใช้ป้อนเข้าไปให้มีความเสมือนเสียงมนุษย์มากที่สุด เขียนย่อๆได้ว่า TTS โดยเทคโนโลยี Text To speech จะรับข้อความที่เป็นตัวอักษรจากผู้ใช้มา แล้ววิเคราะห์ว่าประโยคที่ป้อนเข้ามาควรจะอ่านอย่างไร และสร้างเสียงสังเคราะห์ขึ้นมาอ่านประโยคนั้น ปัจจุบันเทคโนโลยี Text to Speech ถูกนำมาใช้กันอย่างกว้างขวางและสามารถนำไปใช้ได้ในหลายรูปแบบ เช่น การนำไปใช้ออกเสียงในโปรแกรมแปลภาษา, การนำไปใช้ในการอ่านข่าวหรืออ่านหนังสือ, การนำไปสร้างสื่อการสอน, การนำไปทำระบบเสียงเพื่อช่วยเหลือผู้พิการทางสายตา เป็นต้น ซึ่งเทคโนโลยีนี้มีมานานแล้ว และเริ่มเป็นที่นิยมเป็นที่ใช้งานในชีวิตประจำวันเรามากขึ้นเรื่อย ๆ โดยที่หลายๆคนอาจจะไม่สังเกต หรือไม่รู้ตัวว่านี่คือเทคโนโลยีแปลงข้อความเป็นเสียง หากยกตัวอย่างการใช้งานที่สามารถพบได้ในชีวิตประจำวันหลายเราก็คงพอจะเข้าใจ และนึกภาพออกทันที เช่น Google Maps, Siri, Google Translate เป็นต้น (Botnoi Group, 2567)

โดยทางบริษัทที่ให้บริการ text to speech ได้มีแพลตฟอร์ม text to speech ที่เป็นที่นิยม เนื่องจากมีผู้ใช้งานแพลตฟอร์ม text to speech เป็นจำนวนมาก ผู้ที่เข้ามาใช้งานแต่ละคนก็มีจุดประสงค์ที่แตกต่างกัน ด้วยเหตุนี้ผู้วิจัยจึงประยุกต์ใช้การจำแนกหมวดหมู่การใช้งานของกลุ่มลูกค้าเพื่อนำข้อมูลไปพัฒนาแพลตฟอร์ม text to speech แต่เนื่องจากข้อมูลที่ผู้วิจัยต้องการศึกษามีจำนวนมาก และการจำแนกหมวดหมู่เองโดยใช้มนุษย์ อาจจะเกิดความล่าช้า และสิ้นเปลืองทรัพยากรแรงงาน และเวลาเป็นอย่างมาก

ในงานวิจัยครั้งนี้ผู้วิจัยนำการเรียนรู้ของเครื่อง (Machine Learning) มาใช้ในการจำแนกข้อความจากลูกค้า เนื่องจากข้อมูลที่ใช้ในการจำแนกเป็นข้อมูลที่มีลักษณะเป็นข้อความ จึงนำการประมวลผลภาษาธรรมชาติ (NLP) เพื่อทำให้คอมพิวเตอร์สามารถทำความ “เข้าใจ” ข้อมูลที่มีลักษณะเป็นข้อความ และยังรวมถึงการรับรู้ถึงความหมายโดยนัย ความรู้สึกของผู้เขียน ความแตกต่างทางบริบทของภาษา รวมถึงสามารถทำการวิเคราะห์ในรูปแบบต่าง ๆ ได้อีกด้วย (Big Data Institute, 2565) และนำแบบจำลองการจำแนกประเภทข้อมูล (Classification Model) มาใช้ ซึ่งแบบจำลองการจำแนกประเภทข้อมูล เป็นหนึ่งในวิธีการวิเคราะห์ข้อมูลที่เป็นการเรียนรู้ของเครื่องประเภท การเรียนรู้แบบมีผู้สอน (Supervised Learning) ซึ่งผลลัพธ์จากการวิเคราะห์ข้อมูลด้วยแบบจำลองการจำแนกประเภทข้อมูล จะเป็นในรูปแบบของการจำแนกข้อมูลเพื่อให้ได้คำตอบที่เป็นตัวเลือก เช่น Yes กับ No, เป็นกับไม่เป็น หรือเป็นกลุ่มคำตอบว่าเป็นกลุ่ม A B หรือ C (สถาบันนวัตกรรมและธรรมาภิบาลข้อมูล, 2565) ผู้วิจัยหวังเป็นอย่างยิ่งว่าการนำการเรียนรู้ของเครื่อง มาใช้ในการจำแนกกลุ่มลูกค้าจะช่วยในองค์กรในการประหยัดทรัพยากรแรงงานและเวลา และได้ผลลัพธ์ซึ่งแม่นยำพอกับการใช้มนุษย์ในการจำแนกกลุ่มลูกค้า

1.2 วัตถุประสงค์ของงานวิจัย

1) เพื่อศึกษาและจัดหมวดหมู่ของข้อความ

2) เพื่อประยุกต์ใช้แบบจำลองการจำแนกประเภทข้อมูล ในการทำนายหมวดหมู่ของข้อความ

3) เพื่อเปรียบเทียบประสิทธิภาพของแบบจำลองการในจำแนก และวิเคราะห์หมวดหมู่การใช้งานสำหรับลูกค้าของแพลตฟอร์ม text to speech

1.3 ขอบเขตงานวิจัย

1) ขอบเขตด้านข้อมูล

การศึกษาครั้งนี้ได้ดำเนินการศึกษากับบริษัทที่ให้บริการ text to speech ทำการศึกษาเกี่ยวกับข้อมูลที่มีลักษณะเป็นข้อความของลูกค้าที่เข้ามาใช้งานในแพลตฟอร์ม text to speech เป็นข้อมูลของปี พ.ศ.2566

2) ขอบเขตด้านเครื่องมือ

2.1 โปรแกรม Google Colab

2.2 โปรแกรม VSCode (Visual Studio Code)

2.3 ภาษาที่ใช้สำหรับการเขียนโปรแกรม Python

3) ขอบเขตด้านระยะเวลา

งานวิจัยฉบันนี้ทำการดำเนินงานระหว่าง วันที่ 1 มกราคม พ.ศ.2566 ถึง วันที่ 10 มิถุนายน พ.ศ.2566

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1) สามารถนำการเรียนรู้ของเครื่อง มาใช้จำแนกหมวดหมู่ของข้อความต่าง ที่ลูกค้าพิมพ์เข้ามาในระบบ

2) เมื่อทราบจุดประสงค์ของการใช้บริการแล้ว สามารถนำไปปรับกลยุทธ์การตลาด และโปรโมชั่นให้เหมาะสมกับแต่ละกลุ่มของลูกค้า

3) สามารถนำข้อมูลไปพัฒนาเสียงของบอท ปรับปรุงให้ตอบสนองต่อพฤติกรรมของลูกค้าแต่ล่ะกลุ่มเพื่อเพิ่มตัวเลือก และปรับปรุงให้เสียงของบอทมีความเหมาะสมในแต่ล่ะกลุ่มลูกค้า

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

การศึกษาวิจัยในครั้งนี้ผู้วิจัยได้รวบรวมแนวคิด ทฤษฎี และหลักการต่าง ๆ จากเอกสาร และงานวิจัยที่เกี่ยวข้อง ดังหัวข้อต่อไปนี้

2.1 การเรียนรู้ของเครื่อง (Machine Learning)
- 2.1.1 การเรียนรู้แบบมีผู้สอน (Supervised Learning)
- 2.1.2 การเรียนรู้โดยไม่มีผู้สอน (Unsupervised Learning)
- 2.1.3 การเรียนรู้แบบเสริมกำลัง (Reinforcement Learning)

2.2 การประมวลผลภาษาธรรมชาติ (NLP)
- 2.2.1 วิวัฒนาการของการประมวลผลภาษาธรรมชาติ
- 2.2.2 ความสำคัญของการประมวลผลภาษาธรรมชาติ
- 2.2.3 กระบวนการทำงานของการประมวลผลภาษาธรรมชาติ
  - 2.2.3.1 การเตรียมข้อมูล (Data Preprocessing)
  - 2.2.3.2 การแปลงเชิงปริมาณ (Text Representation)
  - 2.2.3.3 เทคนิคการคัดแยกคำตามความสำคัญ (Term Frequency – Inverse Document Frequency : TF-IDF)

2.3 จำแนกประเภทข้อมูล (Classification)
- 2.3.1 แบบจำลองแรนดอมฟอเรสต์ (Random Forest)
- 2.3.2 แบบจำลองการถดถอยแบบโลจิสติก (Logistic Regression)
- 2.3.3 แบบจำลองซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machine; SVM)
- 2.3.4 แบบจำลองนาอีฟเบย์ (Naïve Bayes)

2.4 การวัดประสิทธิภาพของแบบจำลอง (Evaluation)
- 2.4.1 เมทริกซ์ความสับสน (Confusion Matrix)
- 2.4.2 ค่าความแม่นยำ (Accuracy)
- 2.4.3 ค่าความเที่ยง (Precision)
- 2.4.4 ค่าระลึก (Recall)
- 2.4.5 ค่าความถ่วงดุล (F1–Score)
- 2.4.6 ค่าเฉลี่ยมาโคร (Macro Average)

2.5 ปัญหาโอเวอร์ฟิตติง (Overfitting)

2.6 งานวิจัยที่เกี่ยวข้อง

2.1 การเรียนรู้ของเครื่อง (Machine Learning)

คือ การเรียนรู้ของเครื่อง (Machine Learning) ซึ่งเป็นส่วนหนึ่งของปัญญาประดิษฐ์ (Artificial Intelligence) ที่ช่วยให้ระบบสารสนเทศรู้จักกับรูปแบบพื้นฐานของอัลกอริทึม และชุดข้อมูลต่าง ๆ เพื่อเป็นการเรียนรู้แบบอัตโนมัติผ่านข้อมูล และประสบการณ์ด้วยตัวเองเพื่อทำการค้นหา แยกแยะ สรุป คาดคะเน และคำนวณความน่าจะเป็น และเพื่อพัฒนากระบวนการแก้ไข

ปัญหาได้อย่างเหมาะสม โดยไม่ต้องมีมนุษย์มาคอยกำกับ หรือเขียนโปรแกรมเพิ่มเติม และไม่ว่าในอนาคตมันจะมีข้อมูลรูปแบบใหม่ ๆ ที่เกิดขึ้นมา มนุษย์ก็ไม่จำเป็นที่จะต้องไปนั่งเขียนโปรแกรมใหม่ เพราะคอมพิวเตอร์สามารถตีความและตอบสนองได้ด้วยตัวมันเอง แน่นอนว่าธุรกิจ หรืออุตสาหกรรมไหนที่ได้นำเทคโนโลยีนี้ไปปรับใช้ได้อย่างถูกวิธี จะทำให้ได้เปรียบในเชิงการแข่งขันของธุรกิจหรืออุตสาหกรรมเป็นอย่างมาก เพราะสามารถลดเวลาในการทำงาน และลดต้นทุนแรงงานได้มากเลยทีเดียว โดยการเรียนรู้ของเครื่อง (Machine Learning) จะแบ่งออกเป็น 3 ประเภทตามรูปแบบการเรียนรู้ ดังนี้

2.1.1 การเรียนรู้แบบมีผู้สอน (Supervised Learning)

คือ การสอนคอมพิวเตอร์จากข้อมูลตัวอย่าง และผลลัพธ์ที่เรากำหนด (Label) เพื่อให้คอมพิวเตอร์สามารถตอบผลลัพธ์ของข้อมูลชุดใหม่จากตัวอย่างที่ให้ไป

2.1.2 การเรียนรู้โดยไม่มีผู้สอน (Unsupervised Learning)

คือ การสอนให้คอมพิวเตอร์นั้นสามารถเรียนรู้ได้ด้วยตนเอง โดยไม่ต้องมีผลลัพธ์ที่เรากำหนด (Label) ของแต่ละข้อมูล ซึ่งวิธีการคือมนุษย์จะเป็นผู้ใส่ข้อมูลต่าง ๆ และกำหนดสิ่งที่ต้องการจากข้อมูลเหล่านั้น ทำให้เครื่องจักรวิเคราะห์จากการจำแนก และสร้างแบบแผนจากข้อมูลที่ได้รับมา

2.1.3 การเรียนรู้แบบเสริมกำลัง (Reinforcement Learning)

คือ วิธีการเรียนรู้แบบหนึ่งที่ใช้การเรียนรู้ที่เกิดมาจากการปฏิสัมพันธ์ (interaction) ระหว่างผู้เรียนรู้ (agent) กับสิ่งแวดล้อม (environment) ภายใต้การเลือกกระทำสิ่งต่าง ๆ ให้ได้ผลลัพธ์ที่มากที่สุดผ่านการลองผิดลองถูกภายใต้สถานการณ์ที่พัฒนาระบบการตัดสินใจให้ดีขึ้นเรื่อย ๆ ยกตัวอย่าง เช่น Alpha Go เงื่อนไขของการเล่นหมากล้อมให้ชนะ คือ ใช้หมากของตนล้อมพื้นที่บนกระดาน ให้ครอบครองดินแดนมากกว่าคู่ต่อสู้ ทีนี้ Alpha Go ก็จะเรียนรู้ว่าหากคู่ต่อสู้เดินหมากนี้ ตัวมันเองจะเดินหมากไหนเพื่อให้บรรลุเงื่อนไขที่กำหนดไว้ให้ นั่นคือการยึดพื้นที่บนกระดานให้ได้มากที่สุด

โดยในงานวิจัยชุดนี้ผู้วิจัยจะใช้การเรียนรู้ของเครื่อง (Machine Learning) ประเภทการเรียนรู้แบบมีผู้สอน (Supervised Learning) เนื่องจากงานวิจัยชุดนี้ต้องการจำแนก และวิเคราะห์หมวดหมู่ โดยมีการกำหนดหมวดหมู่ไว้เป็นผลลัพธ์ เพื่อให้คอมพิวเตอร์สามารถเรียนรู้และตอบผลลัพธ์ของข้อมูลชุดใหม่จากตัวอย่างที่ให้ไป

2.2 การประมวลผลภาษาธรรมชาติ (NLP)

การประมวลผลภาษาธรรมชาติ (NLP) เป็นเทคโนโลยีการเรียนรู้ของเครื่อง (Machine Learning) ที่ช่วยให้คอมพิวเตอร์สามารถตีความ จัดการ และทำความเข้าใจภาษามนุษย์ได้ องค์กรในปัจจุบันมีข้อมูลเสียง และข้อความจำนวนมากจากช่องทางการสื่อสารต่าง ๆ เช่น อีเมล ข้อความ ฟีด ข่าวโซเชียลมีเดีย วิดีโอ เสียง และอื่น ๆ พวกเขาใช้เทคโนโลยีการประมวลผลภาษาธรรมชาติ (NLP) เพื่อให้คอมพิวเตอร์เข้าใจคำพูด หรือข้อความที่เราพูดหรือเขียนจนสามารถตอบกลับมาโดยอัตโนมัติได้อย่างถูกต้องและแม่นยำ (AWS, 2567)

2.2.1 วิวัฒนาการของการประมวลผลภาษาธรรมชาติ

วิทยาการด้านการประมวลผลภาษาธรรมชาตินั้นมิใช่ศาสตร์ที่เพิ่งเกิดขึ้นใหม่ อย่างไรก็ตาม ความก้าวหน้า และนวัตกรรมใหม่ ๆ ก็กำลังเกิดขึ้นในสาขานี้อย่างต่อเนื่อง รวมไปถึงความก้าวหน้าของ big data ตลอดจนความสามารถในการประมวลผล และอัลกอริทึมที่มีความทันสมัย ปัจจุบันนั้นมนุษย์เรานั้นมีภาษาเป็นของตนเอง เช่น ภาษาอังกฤษ ภาษาสเปน หรือภาษาจีน แต่ภาษาที่คอมพิวเตอร์ใช้ในการทำงานต่าง ๆ นั้น แตกต่างออกไปจากภาษาของเรา ซึ่งเป็นภาษาที่เรียกว่า machine code หรือ machine language ซึ่งเป็นภาษาที่มนุษย์ส่วนมากไม่สามารถตีความได้ การทำงานทุกอย่างของอุปกรณ์ของคุณนั้นล้วนแต่ประกอบขึ้นจากกระบวนการในรูปรหัส 0 และ 1 จำนวนนับล้าน ๆ รายการ ที่ถูกตีความ และแปลงผลให้กลายเป็นการตอบสนองที่มีเหตุผล

ในโลกปัจจุบันการออกคำสั่งแก่อุปกรณ์ของคุณเป็นเรื่องที่ง่ายดายอย่างยิ่ง เช่น คุณสามารถบอกอุปกรณ์ของคุณว่า "Alexa ฉันชอบเพลงนี้" แล้วจากนั้นอุปกรณ์ที่สามารถเล่นเพลงในบ้านของคุณจะตอบสนองความต้องการของคุณได้ เช่น มันอาจลดระดับเสียงลง และตอบคุณด้วยคำพูด และน้ำเสียงที่เหมือนมนุษย์ว่า "โอเค บันทึกการจัดอันดับของคุณไว้แล้ว" จากนั้น มันจะปรับอัลกอริทึมในตัวของมันเองเพื่อเล่นเพลง ๆ นั้น และเพลงอื่น ๆ ที่อาจคล้ายคลึงกันในครั้งต่อ ๆ ไปที่คุณฟังเพลงจากช่องที่เล่นดนตรีช่องดังกล่าวอีก

เมื่อเราพิจารณาการมีปฏิสัมพันธ์ระหว่างมนุษย์ และระบบคอมพิวเตอร์ให้ละเอียดยิ่งขึ้นนั้น เราจะเห็นว่าอุปกรณ์ทำงานเมื่อได้ยินเสียงของคุณและถ้อยคำที่คุณพูด และเข้าใจถึงเจตนาในการพูดของคุณแม้ว่าคุณจะไม่ได้พูดถึงเจตนานั้นโดยตรง จากนั้นมันจึงทำงานบางอย่าง และตอบสนองกลับมาแก่คุณเป็นภาษาอังกฤษที่สละสลวย ซึ่งกระบวนการทั้งหมดนี้กินเวลาเพียงประมาณห้าวินาทีเท่านั้น ซึ่งการทำงานของอุปกรณ์ทั้งหมดที่กล่าวมานี้ เกิดขึ้นได้ด้วยการประมวลผลภาษาธรรมชาติ (NLP) (SAS, 2563)

2.2.2 ความสำคัญของการประมวลผลภาษาธรรมชาติ

ระบบที่ทันสมัยในปัจจุบันสามารถวิเคราะห์ข้อมูลในปริมาณมหาศาลเกินกว่าขีดความสามารถของมนุษย์ โดยตัดข้อจำกัดเรื่องความเหน็ดเหนื่อยออกไป และสามารถทำงานด้วยความแม่นยำ คงเส้นคงวา และปราศจากอคติ การทำงานในปัจจุบันมักต้องรับมือกับข้อมูลดิบจำนวนมหาศาล ซึ่งเกิดขึ้นอย่างต่อเนื่องในแต่ละวัน ไม่ว่าจะเป็นการทำงานในด้านประวัติคนไข้ และทางการแพทย์ ไปจนถึงข้อมูลจากโซเชียลมีเดีย ซึ่งการทำงานโดยอัตโนมัติจาก AI จะเป็นกุญแจสำคัญในการวิเคราะห์ข้อมูลเหล่านี้ได้ ไม่ว่าจะเป็นข้อมูลในรูปข้อความหรือคำพูด การใช้การประมวลผลภาษาธรรมชาติ (NLP) มาช่วยในการรับมือกับข้อมูลข้อความที่มีปริมาณมหาศาลจึงเป็นสิ่งที่สำคัญ

การประมวลผลภาษาธรรมชาติ (NLP) ยังมาช่วยในการจัดระเบียบข้อมูลในลักษณะที่ไร้รูปแบบต่าง ๆ เนื่องจากภาษาที่มนุษย์ใช้กันนั้น มีความซับซ้อน และหลากหลายอย่างยิ่ง เพราะมนุษย์มีวิธีการแสดงออกมากมายนับไม่ถ้วน ทั้งในด้านการสื่อสารด้วยคำพูดหรือข้อความที่เกิดขึ้นด้วยการเขียน นอกจากการมีภาษานับร้อย ๆ พัน ๆ ภาษา ซึ่งต่างมีภาษาถิ่นแยกย่อยลงไปอีกนั้น ทุกภาษายังทวีความซับซ้อนยิ่งขึ้นไปอีกด้วยการมีชุดไวยากรณ์ และโครงสร้างทางภาษาเฉพาะตัวของตนเอง รวมถึงคำ กลุ่มคำ และแม้แต่ศัพท์แสลงต่าง ๆ และเมื่อมนุษย์เราใช้ภาษาในการสื่อสารกันนั้น เรายังมักนิยมเขียนข้อความในรูปแบบย่อ ละเครื่องหมายวรรคตอนออกไป หรือแม้แต่การสะกดคำผิด ส่วนการสื่อสารด้วยวาจานั้นก็ยังมีประเด็นท้าทายของภาษาถิ่นและสำเนียงเฉพาะของแต่ละภูมิภาค แม้แต่ในภาษาเดียวกัน รวมถึงการพูดที่ไม่ชัดเจน อ้ำอึ้ง หรือใช้คำทับศัพท์แทรก ด้วยเหตุนี้การ

ประมวลผลภาษาธรรมชาติ (NLP) จึงมีความสำคัญในการลดความสับสนทางการวิเคราะห์ภาษาลง และเพิ่มมิติให้แก่ข้อมูลในรูปของตัวเลข เพื่อการนำไปใช้งานต่าง ๆ ต่อไป

2.2.3 กระบวนการทำงานของการประมวลผลภาษาธรรมชาติ

การประมวลภาษาธรรมชาติ ประกอบด้วยหลากหลายวิธีการประมวลผล และแปลความหมายของภาษาปกติของมนุษย์ โดยมีขั้นตอนดังต่อไปนี้

2.2.3.1 การเตรียมข้อมูล (Data Preprocessing)

1) การทำความสะอาดข้อความ (Text Cleaning)

เป็นขั้นตอนการลบสิ่งที่ไม่ต้องการออกจากข้อความ เช่น เช่น ตัวอักษรพิเศษ, เครื่องหมายวรรคตอน, หรืออักขระที่ไม่ใช่ตัวอักษร หรือตัวเลข และข้อความที่ไม่มีประโยชน์หรือไม่เกี่ยวข้องกับวัตถุประสงค์ที่กำหนดไว้ เช่น ลิงก์, แท็ก HTML, หรือสัญลักษณ์พิเศษ

2) การตัดคำ (Tokenization)

คือ การแบ่งคำออกเป็นคำ ๆ อย่างถูกต้องตามหลักภาษา โดยการตัดคำภาษาไทย ผู้วิจัยจะใช้ library ที่ชื่อว่า PyThaiNLP โดยในการวิจัยครั้งนี้ผู้วิจัยได้ใช้ตัวอัลกอริทึมในการตัดคำภาษาไทยที่มีชื่อว่า newmm

3) การลบคำฟุ่มเฟือย (Stop Words)

เป็นการนำคำที่ไม่มีความหมายหรือคำที่ไม่มีนัยสำคัญ ในเอกสารออกโดยที่ความหมายของคำ หรือข้อความจะไม่เปลี่ยนแปลง หลังจากที่ตัดคำแล้ว จะนำคำเหล่านั้นไปตรวจสอบกับพจนานุกรมภาษาไทยโดยใช้ Library PyThaiNLP หากพบว่าคำนั้นไม่มีความหมายให้ถือว่าเป็น Stop Word รวมทั้งคำที่มีความหมายแต่มีตัวอักขระเพียง 1 ตัว ก็ถือว่าเป็น Stop Word ด้วยเช่นกัน

2.2.3.2 การแปลงเชิงปริมาณ (Text Representation)

คือ การแปลงข้อความ (Text) ให้กลายเป็นตัวเลข (Numerical) ที่อยู่ในรูปแบบของ เวกเตอร์ (Vector) ที่เหมาะกับการนำไปเข้าการวิเคราะห์แบบจำลอง ซึ่งการแปลงเชิงปริมาณที่ผู้วิจัยนำมาใช้ในงานวิจัยนี้ คือ เทคนิคการคัดแยกคำตามความสำคัญ (Term Frequency – Inverse Document Frequency : TF-IDF)

2.2.3.3 เทคนิคการคัดแยกคำตามความสำคัญ (Term Frequency – Inverse Document Frequency : TF-IDF)

เทคนิคการคัดแยกคำตามความสำคัญโดยการให้น้ำหนักคำในแต่ละคำโดยใช้ 2 ปัจจัยคือ TF และ IDF ซึ่งมักใช้เพื่อหาคำเหมือนในเอกสารที่มีจำนวนมาก หรือเป็นวิธีการให้น้ำหนักสำหรับคำที่มีความสำคัญหรือใช้เป็นตัวแทนของเอกสารที่ควรจะปรากฏอยู่เป็นจำนวนมากในเนื้อหาของเอกสารเฉพาะฉบับนั้น โดยการแปลงคำเป็นตัวเลข โดยจะไม่นำลำดับของคำภายในเอกสารมาวิเคราะห์ประกอบด้วย

- **Term Frequency (TF)**

มีแนวคิดว่าหากคำใดถูกกล่าวถึงบ่อย ๆ ในเอกสารนั้น ๆ มีความเป็นไปได้สูงที่จะเกี่ยวข้องกับความสำคัญของเอกสารนั้นมาก ๆ มีสมการ ดังนี้

$$\text{TF} = \frac{\text{จำนวนคำนั้น ๆ ในเอกสาร}}{\text{จำนวนของคำทั้งหมดในเอกสาร}} \quad (1)$$

- **Inverse Document Frequency (IDF)**

เป็นการคำนวณค่าน้ำหนักความสำคัญของแต่ละคำที่พบในเอกสารหากพบคำนั้นบ่อย จะมีค่า IDF ต่ำ โดยจะมีสูตรการคำนวณ ดังนี้

$$IDF(t) = \log\left(\frac{n}{DF_t}\right) \quad (2)$$

เมื่อ t คือ คำ 1 คำ

n คือ จำนวนเอกสารทั้งหมด

DF_t คือ จำนวนเอกสารที่พบคำ t

จะได้การคำนวณเทคนิคการคัดแยกคำตามความสำคัญ (TF-IDF) ดังนี้

$$TFIDF = TF \times IDF \quad (3)$$

	corn	export	fall	futures	hopes	on	profit	rise	taking
corn futures fall on profit taking	0.3347	0.0000	0.4704	0.3347	0.0000	0.3347	0.4704	0.0000	0.4704
corn futures rise on export hopes	0.3347	0.4704	0.0000	0.3347	0.4704	0.3347	0.0000	0.4704	0.0000

รูปที่ 2. 1 ตัวอย่างการสร้างคุณลักษณะของข้อความ TF-IDF
(**ที่มา** : วันธนวัฒน์, 2565)

2.3 การจำแนกประเภท (Classification)

เป็นแบบจำลองประเภทการเรียนรู้แบบมีผู้สอน (Supervised Learning) หมายถึง แบบจำลองที่ต้องมีตัวแปรตามเป็นตัวตั้งต้นให้เรียนรู้ โดยเป้าหมายของการจำแนกประเภทจะมีข้อมูลเชิงกลุ่ม (Categorical) หมายถึง ข้อมูลที่จัดเป็นหมวดหมู่หรือกลุ่มก้อนแยกประเภทชัดเจน (จำนวนข้อมูลที่แน่นอน ไม่ใช่ตัวเลขที่มีความต่อเนื่อง) ซึ่งจะไม่มีผลในทางคณิตศาสตร์ ไม่มีความหมายในการคำนวณ ไม่มีผลเวลาบวกลบคูณหารกับตัวเลข เช่น ใช่หรือไม่ เพศชายหรือหญิง A/B/C เป็นต้น และนำไปใช้คำนวณทางคณิตศาสตร์ไม่ได้

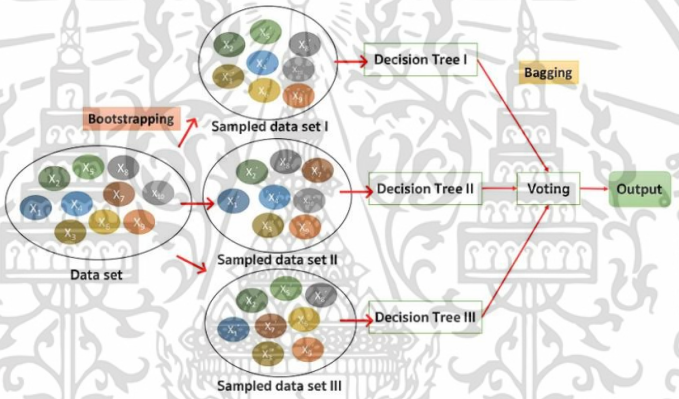
แบบจำลองการจำแนกประเภทข้อมูล (Classification Model) แบ่งออกเป็นหลายประเภท โดยในงานวิจัยชุดนี้ได้ใช้ การจำแนกประเภทแบบหลายคลาส (Multi-Class Classification) เป็นรูปแบบการจำแนกที่มีกระบวนการวิเคราะห์ข้อมูล เพื่อให้ได้ผลลัพธ์ออกมาโดยมีผลลัพธ์มากกว่า 2 ผลลัพธ์ขึ้นไป เช่น นำเข้าข้อมูลที่เป็นรูปภาพ โดยให้แบบจำลองการจำแนกประเภทข้อมูลจำแนกว่ารูปภาพที่เห็นเป็นภาพสัตว์ สิ่งของ หรือไม่ใช่ทั้งสัตว์และสิ่งของ ซึ่งคำตอบของแบบจำลองมีทั้งหมด 3 คำตอบ (สถาบันนวัตกรรมและธรรมาภิบาลข้อมูล, 2565) ซึ่งสามารถประเมินผลที่ได้จากแบบจำลองการจำแนกประเภทข้อมูล โดยการวัดค่าความถ่วงดุล (F1-Score) ค่าความแม่นยำ

(Accuracy) ค่าความเที่ยงตรง (Precision) และค่าระลึก (Recall) จากการใช้เมทริกซ์ความสับสน (Confusion Matrix)

ในงานวิจัยครั้งนี้ผู้วิจัยเปรียบเทียบประสิทธิภาพของแบบจำลองในการจำแนกประเภทข้อมูลทั้งหมด 4 แบบจำลอง ดังนี้ 1.แบบจำลองแรนดอมฟอเรสต์ (Random Forest) 2.แบบจำลองการถดถอยแบบโลจิสติก (Logistic Regression) 3.แบบจำลองซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machine; SVM) 4.แบบจำลองนาอีฟเบย์ (Naive Bayes)

2.3.1 แบบจำลองแรนดอมฟอเรสต์ (Random Forest)

วิธีแรนดอมฟอเรสต์ เป็นแบบจำลองการเรียนรู้แบบมีผู้สอนที่ถูกพัฒนาขึ้นจากวิธีต้นไม้ตัดสินใจ (Decision Tree) โดยที่วิธีแรนดอมฟอเรสต์นั้นเป็นการเพิ่มจำนวนต้นไม้เป็นต้นไม้หลาย ๆ ต้นทำให้ประสิทธิภาพในการทำงานสูงขึ้นแม่นยำมากขึ้น โดยแต่ละแบบจำลองจะได้รับชุดข้อมูลไม่เหมือนกันซึ่งเป็นข้อมูลชุดย่อยของชุดข้อมูลทั้งหมดตอนการฝึกสอนแบบจำลอง ในการพยากรณ์ก็จะให้แต่ละต้นไม้ตัดสินใจและพยากรณ์ผลของตัวเองจากนั้น เมื่อได้ผลแต่ละต้นไม้แล้วจะทำการโหวต (Vote) หรือทำการตัดสินใจเลือกค่าที่ดีที่สุด (จิรวัฒน์และ คณะ, 2565)

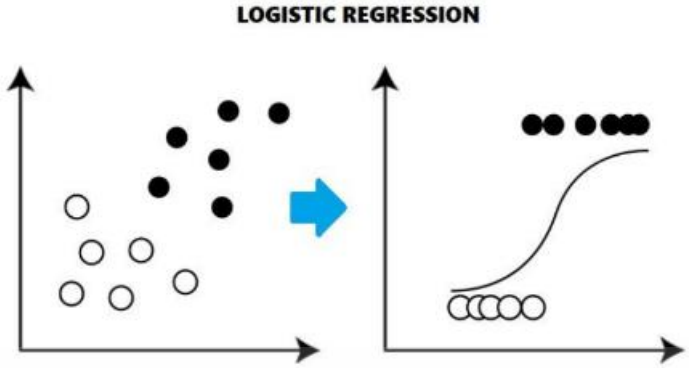


รูปที่ 2.2 กระบวนการทำงานของวิธีแรนดอมฟอเรสต์

จากรูป 2.2 จะแสดงให้เห็นถึงการสร้างต้นไม้และเป็นประเภทข้อมูลแบบ Bagging (Bootstrap Aggregation) โดยต้นไม้แต่ละต้นที่นำมาฝึกสอนในแบบจำลอง จะมีตัวแปรแต่ละตัวเป็นส่วนหนึ่งของคุณลักษณะ (Feature) ซึ่งจะนำมาฝึกสอนในรูปแบบสุ่มและในส่วนขั้นตอนการพยากรณ์ข้อมูล จะกำหนดให้ต้นไม้แต่ละต้นพยากรณ์ในต้นของตัวเองและคัดเลือกผลพยากรณ์สุดท้ายจากค่าพยากรณ์ที่ได้รับการโหวตมากที่สุดเทคนิคดังกล่าวเรียกว่า "การสุ่มตัวอย่างข้อมูล" และเทคนิคการจำแนกประเภทข้อมูลแบบ Bagging (สุภาภรณ์, 2564)

2.3.2 แบบจำลองการถดถอยแบบโลจิสติก (Logistic Regression)

การถดถอยแบบโลจิสติก (Logistic Regression) เป็นเทคนิคที่ใช้ในการพยากรณ์ความน่าจะเป็นที่จะเกิดเหตุการณ์หรือไม่เกิดเหตุการณ์ที่สนใจ ศึกษาความสัมพันธ์ของตัวแปรต้น (Independent Variables) กับตัวแปรตาม (Dependent Variables) ที่เป็นข้อมูลเชิงกลุ่ม (Categorical) มีมาตรวัดแบบนามบัญญัติ (Nominal Scale) แตกต่างจากการถดถอยแบบเดิม (Traditional Regression) ในเบื้องต้น (ยุวดี, 2564) ดังนี้



รูปที่ 2. 3 ฟังก์ชันโลจิสติก (Logistic Function)

1) Model Logistic Regression คือ Logit Model ในรูป

$$log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_j X_j \quad (4)$$

โดย Log เป็น Natural Log หรือเขียน ln หรือเขียนเป็น Model Logistic Regression คือ

$$p = \frac{1}{1+e^{-\left(\beta_0+\beta_1 X_1+\beta_2 X_2+\cdots+\beta_j X_j\right)}} \quad (5)$$

หรือ

$$p = \frac{1}{1+e^{-\left(\beta_0+\sum \beta_j X_j\right)}} \quad (6)$$

เมื่อ ค่า p เป็นค่าความน่าจะเป็น (Probability) ซึ่ง $0 \leq p \leq 1$ โดยพารามิเตอร์ของแบบจำลองการถดถอยโลจิสติกจะไม่สามารถประมาณค่าโดยใช้วิธีกำลังสองน้อยที่สุด (Least Square) ได้

2) การทำนายตัวแปรตาม เมื่อมีค่าใหม่ของตัวแปรต้น ต้องใช้ค่า เป็นเกณฑ์เพื่อจัดเข้ากลุ่มของตัวแปรตาม

การถดถอยแบบโลจิสติกจะแบ่งประเภทตามจำนวนกลุ่ม (Categories) และชนิดของตัวแปรตาม ได้ 2 ประเภทใหญ่ ๆ คือ

1) การถดถอยแบบโลจิสติกทวิ (Binary Logistic Regression) ได้แก่ การถดถอยแบบโลจิสติกที่ตัวแปรตาม มีเพียง 2 กลุ่ม เช่น ซื้อหรือไม่ซื้อ ชนะหรือแพ้ รักษาหายหรือไม่ขาย ควรลงทุนหรือไม่ควรลงทุน เป็นต้น และจำแนกย่อยได้อีก ตามจำนวนของตัวแปรต้น ดังนี้

1.1) การถดถอยแบบโลจิสติกทวิ (Binary Logistic Regression) แบบการถดถอย อย่างง่าย (Simple Regression) หมายถึง การถดถอยแบบโลจิสติกที่ตัวแปรตามมี 2 กลุ่ม และมีตัวแปรต้นตัวเดียว

$$p = \frac{1}{1+e^{-(\beta_0+\beta_1 X)}} \quad (7)$$

1.2) การถดถอยแบบโลจิสติกทวิ (Binary Logistic Regression) แบบการถดถอยพหุคูณ (Multiple Regression) หมายถึง การถดถอยแบบโลจิสติกที่ตัวแปรตามมี 2 กลุ่ม และมีตัวแปรต้นหลายตัวได้

$$p = \frac{1}{1+e^{-\left(\beta_0+\beta_1 X_1+\beta_2 X_2+\cdots+\beta_j X_j\right)}} \quad (8)$$

หรือ

$$p = \frac{1}{1+e^{-\left(\beta_0+\sum \beta_j X_j\right)}} \quad (9)$$

2) การถดถอยแบบโลจิสติกพหุกลุ่ม (Multinomial Logistic Regression) ได้แก่ การถดถอยแบบโลจิสติกที่ตัวแปรตามมีจำนวนมากกว่า 2 กลุ่ม (ตั้งแต่ 3 กลุ่ม ขึ้นไป) ซึ่งจำแนกย่อยไปตามชนิดของตัวแปรต้น

2.1) การถดถอยแบบโลจิสติกแบบนามบัญญัติ (Nominal Logistic Regression) หมายถึง การถดถอยแบบโลจิสติกที่ตัวแปรตามมีจำนวนตั้งแต่ 3 กลุ่ม ขึ้นไป และมีมาตรวัดแบบนามบัญญัติ (Nominal Scale) เช่น กลุ่มคนไข้กรุ๊ปเลือด O, A, B, AB กลุ่ม ผู้รับประทาน อาหารมังสวิรัติ, ไม่ผักเลย, อาหารทั่วไป กลุ่มผู้เรียนคณะวิทยาศาสตร์, คณะศิลปศาสตร์, คณะเกษตรศาสตร์, คณะวิศวกรรมศาสตร์ เป็นต้น

2.2) การถดถอยแบบโลจิสติกแบบเรียงอันดับ (Ordinal Logistic Regression) หมายถึง การถดถอยแบบโลจิสติกที่ตัวแปรตามมีจำนวนตั้งแต่ 3 กลุ่ม ขึ้นไป และมีมาตรวัดแบบเรียงอันดับ (Ordinal Scale) เช่น กลุ่ม อายุมากกว่า 50 ปี, 30-49 ปี, ต่ำกว่า 30 ปี กลุ่มผู้มีความคิดเห็น มาก, ปานกลาง, น้อย เป็นตัน ซึ่งแต่ละกลุ่มเปรียบเทียบคู่กันได้ว่ากลุ่มใดระดับสูงกว่ากัน ทั้งแบบการถดถอยแบบโลจิสติกแบบนามบัญญัติ (Nominal Logistic Regression) และการถดถอยแบบโลจิสติกแบบเรียงอันดับ (Ordinal Logistic Regression) มีสมการแบบ Simple หรือ Multiple ขึ้นกับจำนวนตัวแปรต้นเหมือนกับการถดถอยแบบโลจิสติกทวิ (Binary Logistic Regression)

ความสัมพันธ์ระหว่างตัวแปรต้นกับตัวแปรตามของการถดถอยโลจิสติก ไม่เป็นรูปแบบเชิงเส้น จึงต้องมีการปรับให้อยู่ในรูปของเชิงเส้นในรูปแบบของ ออดส์ (odds) ซึ่งหมายถึง อัตราส่วนระหว่างโอกาสที่จะเกิดเหตุการณ์ที่สนใจกับโอกาสจะไม่เกิดเหตุการณ์ที่สนใจ จะได้ดังสมการที (10)

$$Odds = \frac{p}{1-p} \quad (10)$$

เมื่อ p คือ โอกาสที่จะเกิดเหตุการณ์ที่สนใจ $p(y=1)$

โดยค่าของ Odds จะเป็นการบอกว่าโอกาสที่จะเกิดเหตุการณ์ที่สนใจเป็นกี่เท่าของโอกาสจะไม่เกิดเหตุการณ์ที่สนใจ การเขียนแบบจำลองโลจิสติกจะอยู่ในรูปแบบ Log ของ Odds ซึ่งเรียกว่า Logit หรือ Logistic Response Function โดยจะเขียนอยู่ในรูปดังสมการที่ (11)

$$logit = ln\left(\frac{p}{1-p}\right) \quad (11)$$

หรือ

$$logit = b_0 + b_1x_1 + b_2x_2 + \cdots + b_ix_i \quad (12)$$

เมื่อ b_i คือ สัมประสิทธิ์การถดถอย

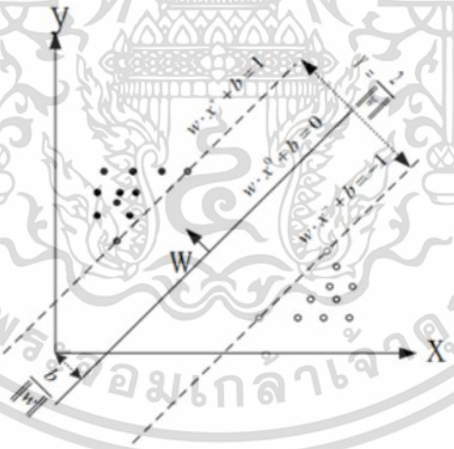
x_i คือ ตัวแปรต้น

เมื่อได้ Logit แล้ว รูปแบบของตัวแปรตามจะสามารถทำนายได้ด้วยแบบจำลองเชิงเส้นตรง และสามารถอธิบายความสัมพันธ์ระหว่างตัวแปรต้นกับตัวแปรตามได้ว่า เมื่อตัวแปร b_i เพิ่มขึ้น 1 หน่วย หาก b เป็นบวก หมายความว่าค่าออดส์ (Odds) จะเพิ่มขึ้น หาก b_i เป็นลบ จะหมายความว่าค่าออดส์ (Odds) จะลดลง และถ้าค่า b_i เป็น 0 หมายความว่าค่าออดส์ (Odds) ไม่เปลี่ยนแปลง ซึ่งสามารถคำนวณค่าออดส์ที่เปลี่ยนแปลงไปได้ดังสมการต่อไปนี้

$$\text{ร้อยละค่าออดส์ที่เปลี่ยนแปลงไป} = (e^{b_i} - 1) \times 100 \quad (13)$$

2.3.3 แบบจำลองซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machine; SVM)

เป็นส่วนหนึ่งของการเรียนรู้ของเครื่อง (Machine Learning) ซึ่งสามารถจำแนกข้อมูลที่มีมิติของข้อมูลสูงออกเป็นสองส่วนขึ้นไป ดังภาพที่ 2.4



รูปที่ 2. 4 ซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machine; SVM)
(**ที่มา:** เทิดศักดิ์ และคณะ 2560)

ซึ่งก่อนที่จะจำแนกข้อมูลจะต้องทำการสอน (Train) ให้เกิดการจดจำข้อมูลของกลุ่มตัวอย่างที่ต้องการจำแนก จากนั้นนำข้อมูลที่ต้องการจำแนกป้อนเข้าสู่ซัพพอร์เวกเตอร์แมชชีน (Support Vector Machine: SVM) เพื่อให้จำแนกกลุ่มข้อมูลออกมา โดยโครงสร้างข้อมูลสำหรับสอน และผลลัพธ์ที่ออกมาจะทำให้ระบบเกิดการจดจำ ดังสมการต่อไปนี้

$$(x_i, y_i), \dots, (x_n, y_n) \text{ เมื่อ } x \in R^m, y \in \{+1, -1\} \quad (14)$$

เมื่อ $(x_i, y_i), \dots, (x_n, y_n)$ เป็นคุณลักษณะสำหรับใช้ในการสอน

n คือ จำนวนข้อมูลตัวอย่าง

m คือ จำนวนมิติของข้อมูล

y คือ ผลลัพธ์มีค่าเป็น +1 หรือ -1

ดังนั้นข้อมูลจะถูกจำแนกออกมาเป็นสองกลุ่ม ดังสมการที่ (15) และ (16)

$$(w \cdot x) + b > 0 \text{ ถ้า } y_i = +1 \quad (15)$$

และ

$$(w \cdot x) + b < 0 \text{ ถ้า } y_i = -1 \quad (16)$$

เมื่อ w คือ ค่าน้ำหนัก

B คือ ค่าความเอนเอียง (bias)

Y คือ ผลลัพธ์มีค่าเป็น +1 หรือ -1

โดยมีเส้นแบ่ง หรือระนาบการตัดสินใจ ซึ่งสามารถคำนวณได้จากสมการที่ (17)

$$(w \cdot x) + b = 0 \quad (17)$$

เวกเตอร์ของข้อมูลที่ป้อนสู่ระบบการสอน เพื่อให้ระบบเรียนรู้ และข้อมูลทั้งสองด้านแบ่งเป็นบวกและลบ ข้อมูลถูกแทนด้วย y ซึ่งประกอบด้วย 2 ค่า คือ y = 1 และ y = -1 แต่ยังตัดสินไม่ได้ว่าเส้นแบ่งใดดีที่สุด ซึ่งวิธีการหาเส้นแบ่งที่ดีที่สุดคือการเพิ่มขอบให้กับเส้นแบ่งทั้งสองด้าน ทำให้ได้เส้นขอบ (Margin) เส้นใหม่ซึ่งถือว่าเป็นขอบของข้อมูลแต่ละด้าน เส้นของทั้งสองเส้นจะถูกแทนด้วยสมการที่ (18) และ (19)

$$(w \cdot x^+) + b \geq y \geq 1 \text{ ถ้าอยู่ด้าน } y = +1 \quad (18)$$

และ

$$(w \cdot x^-) + b \leq y \leq 1 \text{ ถ้าอยู่ด้าน } y = -1 \quad (19)$$

ถ้าเส้นขอบของเส้นแบ่งใด ๆ มีความกว้างมากที่สุด แสดงว่าข้อมูลทั้ง 2 ชุด มีการแบ่งออกกันอย่างชัดเจน จึงบอกได้ว่าเส้นแบ่งนั้นเป็นระนาบการตัดสินใจที่ดีที่สุด ซึ่งสามารถหาความกว้างของเส้นขอบ (Maximization of margin) ได้จากสมการที่ (20) ค่าของ w หาได้จากสมการที่ (21)

$$Maximize\ \gamma = \frac{2}{\|w\|} \quad (20)$$

$$subject\ to\ y_i(w^T x_i + b) \geq 1, \forall i$$

โดย

$$w = \sum_{i=1}^{N} \alpha_i y_i x_i \quad (21)$$

เมื่อ α คือ สัมประสิทธิ์คงที่ $\alpha_i \geq 0; i = 1,2,3, \ldots, N$

เพื่อความสะดวกในการแก้ปัญหามักนิยมหาค่าน้อยที่สุดมากกว่าการหาค่ามากที่สุด ซึ่งสามารถพิจารณาได้จากความสัมพันธ์ต่อไปนี้

$$\gamma = \frac{2}{\|w\|} = \frac{2}{\sqrt{w^T w}} \propto \frac{2}{w^T w} \quad (22)$$

ดังนั้นการหาค่าที่เหมาะสมที่สุดเป็นดังต่อไปนี้

$$Minimize \frac{1}{2} w^T w \quad (23)$$

$$subject\ to\ y_i(w^T x_i + b) \geq 1, \forall i$$

การหาขอบที่กว้างที่สุดระหว่างข้อมูล 2 กลุ่มจะทำได้ก็ต่อเมื่อสามารถหาระนาบที่ สามารถแบ่งข้อมูลทั้ง 2 กลุ่มออกจากกันได้ถูกต้องทั้งหมด เรียกว่า “Hard Margin” แต่ความเป็นจริงข้อมูลอาจไม่เป็นเช่นนั้น จึงทำการเพิ่มตัวแปรที่เรียกว่า “Slack” (ξ) เข้าไปเพื่อเพิ่มประสิทธิภาพ ให้แบบจำลอง และยอมรับค่าสูญเสีย (Loss) ได้ในระดับหนึ่ง ซึ่งเรียกว่า “Soft Margin” ซึ่งสามารถสร้างพจน์เพื่อกำหนดปริมาณความผิดพลาดได้โดยใช้ผลรวมของตัวแปร Slack ดังนี้

$$C \sum_{i=1}^{n} \xi_i \quad (24)$$

เมื่อ C คือ ค่าคงที่ซึ่งเป็นพารามิเตอร์ในการกำหนดปริมาณความผิดพลาด

หากมีค่ามาก หมายถึง ยอมให้ความผิดพลาดเกิดได้น้อย ซึ่งหากมีค่ามากอาจจะเกิด ปัญหาพอดีเกินไป (Overfitting) ของแบบจำลองได้

หากมีค่าน้อย หมายถึง ยอมให้ความผิดพลาดเกิดได้มาก แต่จะลดปัญหา Overfitting ทำให้สามารถใช้งานกับข้อมูลทั่วไปได้มากกว่า แต่หากน้อยเกินย่อมมีค่าผิดพลาดมากเกินที่จะยอมรับได้

ดังนั้นการเลือกค่า C จะมีผลต่อประสิทธิภาพของแบบจำลอง ซึ่งการเลือกค่าที่เหมาะสมนั้นทำได้ยาก ส่วนใหญ่ผู้ใช้มักเป็นผู้กำหนด

เมื่อนำค่าความหย่อน (Slack) มารวมกับปัญหาเดิม จะได้ปัญหาใหม่สำหรับซัพพอร์ตเวเตอร์แมชชีน (SVM) กรณี Soft Margin เป็นดังนี้

$$Minimize \frac{1}{2} w^T w + C \sum_{i=1}^{n} \xi_i \quad (25)$$

$$subject\ to\ y_i(w^T x_i + b) \geq 1, \forall i$$

ฟังก์ชันเคอร์เนล (Kernel Function)

เป็นฟังก์ชันการส่งชนิดหนึ่งที่เกิดจากผลคูณภายในทั้งหมดที่เป็นไปได้ของเวกเตอร์เซ็ตหนึ่ง สามารถเปลี่ยนข้อมูลที่มีมิติต่ำกว่าให้มีมิติสูงขึ้นเพื่อการแบ่งข้อมูล โดยจะอยู่ในรูปดังนี้

$$k(u,v) = \Phi(u)^T \Phi(v) \quad (26)$$

โดยฟังก์ชันเคอร์เนลที่นำมาใช้จะต้องสอดคล้องกับเงื่อนไขของเมอร์เซอร์ (Mercer's Condition) ซึ่งจะมีสมบัติต่อเนื่อง (Continuous), สมมาตร (Symmetric) และกึ่งบวกแน่นอน (Positive Semi-Definite) ซึ่งหมายความว่า เมตริกซ์นี้จะไม่มีค่าลักษณะเฉพาะ (Eigenvalue) ที่เป็นลบ โดยฟังก์ชันเคอร์เนลที่นิยมใช้ มีดังต่อไปนี้

1.เส้นตรง (Linear)

$$K(a,b) = a^T b \quad (27)$$

2.พหุนาม (Polynomial)

$$K(a,b) = (\gamma a^T b + r)^d \quad (28)$$

3.เกาส์เซียน เรเดียลเบสิสฟังก์ชัน (Gaussian RBF)

$$K(a,b) = e^{(-\gamma \|a-b\|^2)} \quad (29)$$

4.ซิกมอยด์ (Sigmoid)

$$K(a,b) = tanh(\gamma a^T b + r) \quad (30)$$

เมื่อ d, γ, a และ b เป็นพารามิเตอร์ของฟังก์ชันเคอร์เนล โดยมีค่าคงที่และขึ้นอยู่กับความเหมาะสม ซึ่งจะนิยมปรับด้วยมือ

2.3.4 แบบจำลองนาอีฟเบย์ (Naive Bayes)

เป็นวิธีที่ให้ผลการจำแนกได้ดีไม่แตกต่างวิธีการอื่นโดยมีขั้นตอนวิธีการทำงานที่ไม่ซับซ้อน การเรียนรู้ของนาอีฟเบย์จะเป็นการเรียนรู้โดยใช้หลักการของความน่าจะเป็น (Probability) ซึ่งมีพื้นฐานมาจากทฤษฎีเบส์ (Bayes Theorem) หรือทฤษฎีว่าด้วยโอกาสที่จะเกิดของเหตุการณ์ต่าง ๆ ซึ่งจะใช้การคำนวณความน่าจะเป็นแบบมีเงื่อนไข (Conditional Probability) ดังสมการที่ (31) (Dietrich et al., 2015)

$$P(h|D) = \frac{P(D|h) \times P(h)}{P(D)} \quad (31)$$

D แทนข้อมูลที่นำมาใช้ในการคำนวณการแจกแจงความน่าจะเป็นภายหลัง (Posteriori Probability) ของการเกิดเหตุการณ์ h คือ $P(h|D)$

โดยที่ $P(h)$ คือ ค่าความน่าจะเป็นก่อน (Prior probability) ของการเกิดเหตุการณ์ h

$P(D)$ คือ ค่าความน่าจะเป็นก่อนของชุดข้อมูลตัวอย่าง D

$P(h|D)$ คือ ค่าความน่าจะเป็นของ h เมื่อรู้ D

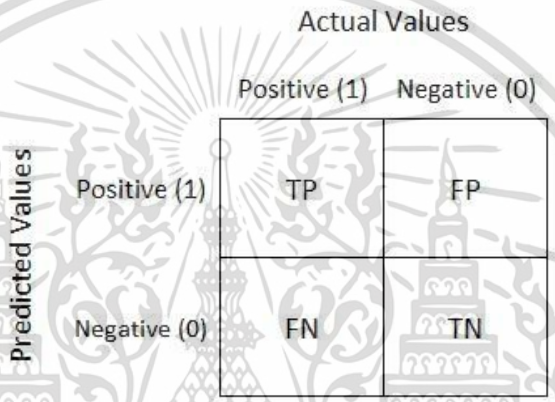
$P(D|h)$ คือ ค่าความน่าจะเป็นของ D เมื่อรู้ h

กำหนดให้ $P(h)$ คือ ความน่าจะเป็นที่จะเกิดเหตุการณ์ h และ $P(h|D)$ คือความน่าจะเป็นที่จะเกิดเหตุการณ์ h เมื่อเกิดเหตุการณ์ D แล้วจากตัวแปรที่กำหนด และแนวคิดของเบส์นั้นเราสามารถพยากรณ์เหตุการณ์ที่พิจราณาได้จากการเกิดของเหตุการณ์ต่าง ๆ ได้ดั่งสมการข้างต้น

2.4 การวัดประสิทธิภาพของแบบจำลอง (Evaluation)

2.4.1 เมทริกซ์ความสับสน (Confusion Matrix)

เป็นเครื่องมือสำคัญในการประเมินผลลัพธ์ของการทำนาย หรือประเมินวัดความแม่นยำและประสิทธิภาพของแบบจำลองที่พยากรณ์กับข้อมูลที่เกิดขึ้นจริงว่ามีสัดส่วนเป็นอย่างไร โดยจะสร้างเป็นตาราง ดังรูปที่ 2.5



รูปที่ 2. 5 ตัวอย่างตาราง Confusion Matrix ขนาด 2x2

ความหมายตัวย่อของรูปที่ 2.5

บวกจริง (True Positive : TP) คือ การทำนายกลุ่มที่สนใจได้อย่างถูกต้อง

ลบจริง (True Negative : TN) คือ การทำนายกลุ่มที่ไม่สนใจได้อย่างถูกต้อง

บวกเท็จ (False Positive :FP) คือ การที่แบบจำลองทำนายออกมาเป็นกลุ่มที่สนใจ แต่ในข้อมูลจริงคือกลุ่มที่ไม่สนใจ

ลบเท็จ (False Negative : FN) คือ การที่แบบจำลองทำนายออกมาเป็นเป็นกลุ่มที่ไม่สนใจ แต่ในข้อมูลจริงคือกลุ่มที่สนใจ

เนื่องจากข้อมูลที่นำมาใช้ในงานวิจัยนี้ตัวแปรตาม (หมวดหมู่การใช้งาน) โดยแบ่งเป็น 9 หมวดหมู่ ได้แก่ A, B, C, D, E, F, G, H และ I จะได้ขนาดของ Confusion Matrix เป็นเมทริกซ์ขนาด 9x9 ซึ่งจะมีจำนวนเซลล์ทั้งหมด 81 เซลล์ โดยแต่ละเซลล์จะแสดงจำนวนของการจำแนกตัวอย่างที่ออกจากแบบจำลองในแต่ละค่า (true positive, false positive, false negative, true negative) โดยแถวจะแสดงค่าจริง (Actual Values) และคอลัมน์จะแสดงค่าที่แบบจำลองทำนายได้ (Predicted Values) โดย จะแบ่งเป็น 9 ตาราง คือ กรณีพิจารณาจากหมวดหมู่ A, กรณีพิจารณาจากหมวดหมู่ B, กรณีพิจารณาจากหมวดหมู่ C, กรณีพิจารณาจากหมวดหมู่ D, กรณีพิจารณาจากหมวดหมู่ E, กรณีพิจารณาจากหมวดหมู่ F, กรณีพิจารณาจากหมวดหมู่ G, กรณีพิจารณาจาก

หมวดหมู่ H และกรณีพิจารณาจากหมวดหมู่ I โดยผู้วิจัยจะยกตัวอย่าง เมทริกซ์ความสับสน (Confusion Matrix) กรณีพิจารณาจากหมวดหมู่ A

โดยสามารถอธิบายความหมายของ ตารางที่ 2.1 เมทริกซ์ความสับสน (Confusion Matrix) กรณีพิจารณาจากหมวดหมู่ A ได้ ดังนี้

บวกจริง (True Positive : TP) คือ การทำนายเป็นหมวดหมู่ A ได้ถูกต้อง

ลบจริง (True Negative : TN) คือ การทำนายเป็นหมวดหมู่ B ถึง I ได้ถูกต้อง

บวกเท็จ (False Positive :FP) คือ การทำนายเป็นหมวดหมู่ A โดยที่ค่าจริง คือ หมวดหมู่ B ถึง I

ลบเท็จ (False Negative : FN) คือ การทำนายเป็นหมวดหมู่ B ถึง I โดยที่ค่าจริง คือ หมวดหมู่ A

ตารางที่ 2. 1 เมทริกซ์ความสับสน (Confusion Matrix) กรณีพิจารณาจากหมวดหมู่ A

		ค่าทำนาย								
		A	B	C	D	E	F	G	H	I
ค่าจริง	A	TP	FN	FN	FN	FN	FN	FN	FN	FN
	B	FP	TN	FN	FN	FN	FN	FN	FN	FN
	C	FP	FN	TN	FN	FN	FN	FN	FN	FN
	D	FP	FN	FN	TN	FN	FN	FN	FN	FN
	E	FP	FN	FN	FN	TN	FN	FN	FN	FN
	F	FP	FN	FN	FN	FN	TN	FN	FN	FN
	G	FP	FN	FN	FN	FN	FN	TN	FN	FN
	H	FP	FN	FN	FN	FN	FN	FN	TN	FN
	I	FP	FN	FN	FN	FN	FN	FN	FN	TN

2.4.2 ค่าความแม่นยำ (Accuracy)

ค่าความแม่นยำ (Accuracy) เป็นการวัดความแม่นยำของแบบจำลองโดยรวม กล่าวคือ แบบจำลองทำนายถูกกี่ครั้งจากจำนวนการทำนายทั้งหมด โดยสามารถคำนวณได้จาก

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (32)$$

2.4.3 ค่าความเที่ยง (Precision)

ค่าความเที่ยง (Precision) คือค่าความแม่นยำในการทำนายในกลุ่มเป้าหมายหรือพิจารณาเฉพาะที่เป็น True Positives (TP) โดยสามารถคำนวณได้จาก

$$Precision = \frac{TP}{TP+FP} \quad (33)$$

2.4.4 ค่าระลึก (Recall)

ค่าระลึก (Recall) คือความสามารถของแบบจำลองในการทำนาย หรือระบุสิ่งที่เราสนใจ ซึ่งคือสัดส่วนของ True Positives (TP) กับข้อมูลที่จริงๆ เป็น Positive ทั้งหมด โดยสามารถคำนวณได้จาก

$$Recall = \frac{TP}{TP+FN} \quad (34)$$

2.4.5 ค่าความถ่วงดุล (F1-Score)

ค่าความถ่วงดุล (F1-Score) คือค่าเฉลี่ยแบบฮาร์โมนิกของค่าความเที่ยง (Precision) และค่าระลึก (Recall) โดยสามารถคำนวณได้จาก

$$F1 - Score = 2 \times \left[\frac{Precision \times Recall}{Precision+Recall}\right] \quad (35)$$

ซึ่งการวัดผลทั้ง 4 ค่าที่กล่าวมาข้างต้น หากมีค่ามาก หมายความว่าแบบจำลองมีประสิทธิภาพที่ดี และสามารถอธิบายเป็นค่าร้อยละได้ (ปณยา, 2566)

2.4.6 ค่าเฉลี่ยมาโคร (Macro Average)

เนื่องจากชุดข้อมูลที่ใช้เป็นชุดข้อมูลที่ไม่สมดุล (Imbalance Data) เพื่อให้ความสำคัญกับทุกคลาสเท่าๆ กันโดยไม่คำนึงว่าแต่ละคลาสจะมีจำนวนข้อมูลมากน้อยเพียงใดจึงเลือกใช้การวัดผลแบบ Macro average ในการคำนวณเพื่อทำให้ค่าเฉลี่ยของน้ำหนักแต่ละคลาสเท่ากัน และให้มั่นใจว่าการวัดผลมีความสมดุล โดยเราจะหา ค่าความเที่ยงแบบมาโคร ค่าระลึกแบบมาโคร และค่าความถ่วงดุลแบบมาโครจากสมการด้านล่าง ดังนี้

1.ค่าความเที่ยงแบบมาโคร (Marco - Precision)

$$Macro - Precision = \frac{Precision_1+Precision_2+\ldots+Precision_i}{i} \quad (36)$$

2.ค่าระลึกแบบมาโคร (Macro - Recall)

$$Macro - Recall = \frac{Recall_1+Recall_2+\ldots+Recall_i}{i} \quad (37)$$

3.ค่าความถ่วงดุลแบบมาโคร (Macro - F1-Score)

$$Macro - F1 - score = 2 \times \frac{Macro - Precision \times Macro - Recall}{Macro - Precision+Macro - Recall} \quad (38)$$

เนื่องจากแบบจำลองการจำแนกประเภทข้อมูล (Classification Model) ในงานวิจัยชุดนี้ได้ใช้การจำแนกประเภทแบบหลายคลาส (Multi-Class Classification) การเปรียบเทียบประสิทธิภาพของแบบจำลองในงานวิจัยชุดนี้ จึงใช้ค่าความแม่นยำมาวัดผลไม่ได้ เพราะถ้าเป็นการจำแนกประเภทที่มากกว่า 2 คลาสขึ้นไปวิธีนี้จะไม่สามารถวัดค่าความแม่นยำบนแต่ละคลาสได้แม่นยำ เช่น ถ้าชุดข้อมูลมีทุเรียนหมอนทองมากกว่าทุเรียนก้านยาว แน่นอนว่ามันก็ต้องทำนายเก่งบนทุเรียนหมอนทอง และส่งผลให้ความแม่นยำสูงตามไปด้วยเพราะเฉลยเองก็มีแต่หมอนทองเป็นส่วนใหญ่ จึงทำให้ไม่ทราบเลยว่ามันสามารถแยกแยะได้จริงหรือไม่

โดยผู้วิจัยจะดูค่า F1-Score เป็นอันดับแรกเพราะ F1-Score เป็นการเฉลี่ยเอาความสมดุลระหว่าง Precision และ Recall โดยจะให้ความสำคัญเท่ากันทั้งสอง ซึ่งช่วยให้เราได้มองการทำนาย

ที่ดีทั้งในกรณีที่ต้องการลด False Positive (สูงขึ้นใน Precision) และในกรณีที่ต้องการลด False Negative (สูงขึ้นใน Recall) และถูกออกแบบมาให้ทำงานได้ดีกับชุดข้อมูลที่ไม่สมดุล (Korstanje, 2021)

2.5 ปัญหาโอเวอร์ฟิตติง (Overfitting)

หมายถึง การนำแบบจำลองไปทำนายข้อมูลที่ถูกสอนหรือถูกฝึกได้อย่างถูกต้อง และแม่นยำมาก แต่พอนำไปทดสอบกับข้อมูลที่ยังไม่เคยพบเห็น หรือนำไปใช้งานจริง กลับพบว่าถูกต้อง และแม่นยำลดลง หรือน้อยมาก (sklsongkiat, 2565) ยังไม่ได้ใส่อ้างอิง

2.6 งานวิจัยที่เกี่ยวข้อง

Akanksha Patro, Mahima Patel, Richa Shukla, & Jagurti Save, (2020) ได้กล่าวถึงการมีแหล่งข้อมูลจำนวนมากบนอินเทอร์เน็ตที่สร้างข่าวรายวันจำนวนมหาศาล จึงมีความจำเป็นที่จะต้องจัดประเภทบทความข่าวเพื่อให้ข้อมูลพร้อมใช้งานแก่ผู้ใช้ได้อย่างรวดเร็ว และมีประสิทธิภาพ ดังนั้นงานวิจัยนี้จึงทำการจัดประเภทข่าว โดยเริ่มต้นด้วยการรวบรวมบทความ ข่าวแบบเรียลไทม์จากเว็บไซต์ข่าวด้วยเทคนิคการทำ Web Scraping แล้วทำการจัดประเภทข่าวโดยอัตโนมัติโดยใช้โมเดลในการจำแนกประเภทต่าง ๆ บทความนี้กล่าวถึงโมเดลในการจัดประเภทข่าวได้แก่ Naïve Bayes, Multinomial Logistic Regression, Support Vector Machine (SVM), Decision tree และ Random Forest สำหรับการจัดหมวดหมู่บทความข่าวโดยอัตโนมัติให้เป็นประเภทต่าง ๆ โดยใช้ชุดข้อมูลจากเว็บไซต์รายงานข่าวนานาชาติของสำนักข่าวบีบีซี (BBC News) ที่มีบทความประเภทข่าวที่แตกต่างกันได้แก่ ธุรกิจ, บันเทิง, การเมือง, กีฬา และเทคโนโลยี โดยบทความนี้จะตรวจสอบผลลัพธ์ของโมเดลการจำแนกประเภท และทำการเปรียบเทียบประสิทธิภาพของโมเดลต่าง ๆ โดยจากผลการทดลองโดยพิจารณาค่าความแม่นยำ, ค่าความเที่ยงตรง, ค่าความถูกต้อง และค่าF1-Score พบว่าโมเดล Multinomial Logistic Regression ให้ค่าที่ดีที่สุดถึง 95.5% โดยเหมือนทำการตรวจสอบผลลัพธ์การจัดประเภทของข่าว พบข่าวประเภทธุรกิจมีการจัดประเภทที่ถูกต้องที่สุดตามด้วยข่าวประเภทกีฬา โดยงานวิจัยที่กล่าวมาข้างต้นมีปัญหาในการจัดประเภทข่าวเพราะข่าวมีจำนวนเยอะมากจึงต้องการจัดประเภทข่าวให้พร้อมใช้งาน และมีประสิทธิภาพซึ่งสอดคล้องกับปัญหาของงานวิจัยชุดนี้ ผู้วิจัยได้ศึกษาเกี่ยวกับการใช้การเรียนรู้ของเครื่องในการจัดประเภทของข่าว และการใช้แบบจำลอง Naïve Bayes, Multinomial Logistic Regression, Support Vector Machine (SVM) และ Random Forest

Velay and Daniel (2018) ได้ศึกษาการใช้การประมวลผลภาษาธรรมชาติเพื่อทำนายแนวโน้มของดัชนี DJIA ใช้การสร้างตัวแทนเชิงความหมายของคำ และข้อความ (Word embedding) ด้วยวิธี Word2Vec และใช้แบบจำลอง Logistic Regression, Linear Discriminant Analysis, K-Nearest Neighbors, Decision Tree Classifier, Support Vector Machine, Random Forrest, Extreme Gradient Boosting, Naive Bayes, Long Short Term Memory (LSTM), Multi-Layer Perceptron (MLP) พบว่าแบบจำลอง Logistic Regression มีความแม่นยำ 57% ซึ่งมากกว่าแบบจำลองประเภทอื่น ๆ โดยงานวิจัยที่กล่าวมาข้างต้นผู้วิจัยได้ศึกษาเกี่ยวกับการประมวลผลภาษาธรรมชาติ และแบบจำลอง Logistic Regression, Support Vector Machine, Random Forrest และNaive Bayes

เอกรัฐ และภราดร (2564) ได้นำเสนอการพัฒนาตัวแบบในการจำแนกประเภทข้อความภาษาไทยที่เป็นการระรานทางไซเบอร์ โดยใช้อัลกอริทึม LSTM (Long Short-Term Memory), GRU (Gated Recurrent Units), Bidirectional LSTM, และ Bidirectional GRU โดยการทำ Word Representation แบบ Word2Vec ที่สร้างขึ้นจากชุดข้อมูล และการใช้ Pretrained Model จาก WangchanBerta นอกจากนี้ยังใช้อัลกอริทึม Naïve Bayes โดยทำ Word Representation แบบ TF-IDF นำมาเปรียบเทียบอัลกอริทึมเพื่อประเมินประสิทธิภาพในการจำแนกข้อความทั้งในด้านความถูกต้อง (Accuracy) ความแม่นยำ (Precision) ความระลึก (Recall) และค่าความถ่วงดุล (F1-Score) ซึ่งพบว่า อัลกอริทึมที่ใช้ Word Representation จาก Word2Vec ที่พัฒนาจากชุดข้อมูล มีประสิทธิภาพมากกว่า การใช้อัลกอริทึม Naïve Bayes ที่ใช้ Word Representation แบบ Tf-Idf และให้ประสิทธิภาพดีกว่าการใช้ Pretrained Model แบบ WangchanBerta โดยงานวิจัยที่กล่าวมาข้างต้นผู้วิจัยได้ศึกษาเกี่ยวกับวิธีการทำ Word Representation ทั้งวิธี Word2Vec และวิธี TD-IDF

กานดา แผ่วัฒนากุล (2566) กล่าวว่าข้อเสนอแนะของผู้บริโภคช่วยบ่งชี้ว่าธุรกิจควรปรับปรุงในทิศทางใด แต่เนื่องจากอินเตอร์เน็ตมีบทวิจารณ์จำนวนมาก ทั้งข้อเท็จจริง ข้อคิดเห็น และข้อเสนอแนะปะปนกัน อีกทั้งโครงสร้างประโยคที่ไม่แน่นอนทำให้ยากต่อการตีความ การจำแนกประเภทข้อมูลจะช่วยให้ประมวลผลได้ดีขึ้น บทความวิจัยนี้จึงนำเสนอกระบวนการแก้ปัญหาดังกล่าว ได้แก่ (1) กระบวนการจำแนกข้อเสนอแนะออกจากบทวิจารณ์ประเภทอื่น โดยเปรียบเทียบผลลัพธ์ของ อัลกอริทึมต้นไม้ตัดสินใจ นาอีฟเบย์ และซัพพอร์ตเวกเตอร์แมชชีน เพื่อหาอัลกอริทึมที่เหมาะสมที่สุด (2) กระบวนการจำแนกประเภทข้อเสนอแนะ ออกเป็น 4 ประเภท ได้แก่ ข้อเสนอแนะทางตรง ข้อเสนอแนะเชิงขอร้อง ข้อเสนอแนะเชิงคำถาม และข้อเสนอแนะเชิงเงื่อนไข การทดลองใช้บทวิจารณ์ทั้งสิ้น 2,561 ประโยค พบว่าอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีน แบบเคอร์เนลโพลิโนเมียล ที่มีอินพุตเวกเตอร์ประกอบด้วยคำ กับการกำกับคำที่เกิดขึ้นร่วมกันบ่อย ได้ผลลัพธ์การจำแนกข้อเสนอแนะดีที่สุด มีค่าความแม่นยำ 85.75% ค่าความระลึก 93.62% และค่าถ่วงดุล 89.51% จากนั้นจำแนกประเภทข้อเสนอแนะ และวัดประสิทธิภาพด้วยค่าเฉลี่ยแบบให้น้ำหนักทุกประเภทเท่ากัน (Micro averaging) ได้ค่าความแม่นยำ 94.94% และความระลึก 94.94% กระบวนการที่นำเสนอถือว่ามีความถูกต้องสูงสำหรับข้อเสนอแนะที่ไม่มีความกำกวม ช่วยลดระยะเวลาการอ่านบทวิจารณ์ และข้อเสนอแนะลงได้ โดยงานวิจัยที่กล่าวมาข้างต้นได้ทำการวัดประสิทธิภาพด้วยวิธี Macro Averaging คือการคำนวณด้วยค่าเฉลี่ยแบบให้น้ำหนักเท่ากันทุกคลาสซึ่งสอดคล้องกับวิธีการวัดประสิทธิภาพของงานวิจัยนี้ ผู้วิจัยได้ศึกษาเกี่ยวกับการประเมินประสิทธิภาพการจำแนกประเภทข้อความ ค่า Micro averaging และค่า Macro averaging

Yildirim et al. (2018) ศึกษาการจำแนก “ข่าวด่วน” เพื่อพยากรณ์การเงินด้วยเทคนิค NLP ใช้ Text Representation ด้วยวิธี Bag of Words และ TF-IDF พบว่าแบบจำลอง Support Vector Machine (SVM) สามารถจำแนก “ข่าวด่วน” เพื่อพยากรณ์การเงินโดยมีความแม่นยำที่ 91.4% ได้ดีกว่าเมื่อเปรียบเทียบกับแบบจำลอง k-Nearest Neighbor (KNN), Logistic Regression, linear kernel and multinomial Naïve Bayes (m-NB) โดยงานวิจัยที่กล่าวมาข้างต้นผู้วิจัยได้ศึกษาเกี่ยวกับการจำแนกประเภทข่าวด้วยเทคนิค NLP โดยใช้ Text Representation ด้วยวิธี Bag of Words และ TF-IDF โดยนำไปใช้กับแบบจำลอง Support Vector Machine (SVM) และแบบจำลอง Logistic Regression

Cai et al. (2019) ได้ทำการศึกษา และพัฒนาแบบจำลองสำหรับระบบการแจ้งเตือนภัยสำหรับโรงงานอุตสาหกรรม โดยในงานวิจัยนี้ได้เน้นการพัฒนาระบบสำหรับใช้ในโรงงานรูปแบบสมัยใหม่ที่เน้นการทำงานในรูปแบบอัตโนมัติ และสามารถเพิ่มประสิทธิภาพในการทำงานในด้านต่าง ๆ โดยทางผู้วิจัยได้ทำการพัฒนาแบบจำลองการแจ้งเตือนอุทกภัยโดยรอบ ซึ่งการแจ้งเตือนที่รวดเร็วจะทำให้ลดความสูญเสียในด้านต่าง ๆ ได้เป็นอย่างมาก การพัฒนาแบบจำลองนั้นจะใช้หลักการของ Deep Learning และ Natural Language Processing ในการพัฒนาแบบจำลอง โดยใช้ Word2Vec ในการสร้าง Feature สำหรับนำไปใช้ในการฝึกสอน และใช้งาน LSTM ในการสร้างแบบจำลองการทำนายผลโดยสามารถทำนายผลได้อย่างแม่นยำสูงถึง 80% โดยงานวิจัยที่กล่าวมาข้างต้นผู้วิจัยได้ศึกษาเกี่ยวกับการใช้ Natural Language Processing หรือการประมวลผลภาษาธรรมชาติในการสร้างแบบจำลอง

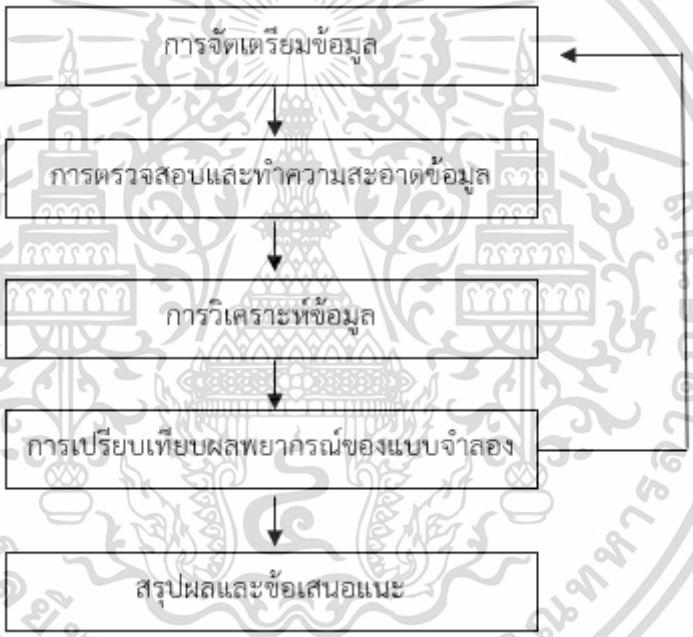
Haddi et al. (2013) ได้ทำการศึกษาวิธีการสกัด Feature จากชุดข้อมูลข้อความสำหรับการสร้างแบบจำลองจากชุดข้อมูลที่มาจาก Social Network ซึ่งเป็นชุดข้อมูลขนาดใหญ่ที่ต้องการสร้างระบบการทำ Opinion Mining หรือการวิเคราะห์ความรู้สึกแบบอัตโนมัติผู้วิจัยได้ทำการศึกษาการสร้าง Feature ด้วยหลักการของ TF-IDF โดยเป็นการวิเคราะห์ในรูปแบบของการตรวจสอบทั้งจำนวนความถี่ของคำที่พบ และความถี่ของคำที่พบในแต่ละเอกสาร และทำการฝึกสอนแบบจำลองโดยใช้อัลกอริทึม Support Vector Machine โดยใช้ชุดข้อมูลการแสดงความคิดเห็นต่อภาพยนตร์เรื่องต่าง ๆ บนระบบอินเทอร์เน็ต จากนั้น ทำการวัดประสิทธิภาพในการสร้างพบว่า การสร้าง Feature ด้วย TF-IDF และสร้างแบบจำลองโดยใช้ Support Vector Machine นั้นให้ประสิทธิภาพในการทำนายด้วยค่าความถูกต้องที่ 93.5%, Precision 94%, Recall 93.06% และ FMeasure 93.53% โดยงานวิจัยที่กล่าวมาข้างต้นผู้วิจัยได้ศึกษาเกี่ยวกับการสกัดคุณลักษณะ ด้วยวิธี TF-IDF และแบบจำลอง Support Vector Machine

บทที่ 3
วิธีดำเนินงานวิจัย

งานวิจัยนี้มีวัตถุประสงค์เพื่อศึกษาเกี่ยวกับการจำแนกและวิเคราะห์หมวดหมู่การใช้งานสำหรับลูกค้าของแพลตฟอร์ม text to speech ผู้วิจัยได้นำทฤษฎี แนวคิด และงานวิจัยที่เกี่ยวข้องมากำหนดขั้นตอนในการศึกษาดังนี้

3.1 ขั้นตอนการดำเนินงาน

การจำแนกและวิเคราะห์หมวดหมู่การใช้งานสำหรับลูกค้าของแพลตฟอร์ม text to speech เพื่อให้เกิดความเข้าใจในข้อมูล และการจัดเตรียมข้อมูลในการวิเคราะห์ สามารถแบ่งออกเป็นขั้นตอนต่าง ๆ ได้ดังนี้



รูปที่ 3. 1 ขั้นตอนการดำเนินงาน

3.2 การจัดเตรียมข้อมูล

งานวิจัยนี้ได้ใช้ชุดข้อมูลของบริษัทที่ให้บริการ text to speech โดยข้อมูลที่ใช้มีลักษณะเป็นข้อความของลูกค้าที่เข้ามาใช้งานในแพลตฟอร์ม text to speech โดยทำการเก็บรวบรวมข้อมูลตั้งแต่วันที่ 1 มกราคม พ.ศ.2566 ถึง วันที่ 10 มิถุนายน พ.ศ.2566

3.3 การตรวจสอบและทำความสะอาดข้อมูล

3.3.1 การทำความสะอาดข้อความ (Text Cleaning)

```
def cleanText(text):
    text = str(text)
    text = re.sub('[^ก-๙]','',text)
    stop_word = list(thai_stopwords())
    sentence = word_tokenize(text)
    result = [word for word in sentence if word not in stop_word and " " not in
word]
    return text
cleaning = [ ]
for txt in df["Mes"]:
    cleaning.append(cleanText(txt))
cleaning[:10]
```

รูปที่ 3. 2 แสดง Python Code สำหรับการทำความสะอาดข้อมูล

จากรูปที่ 3.2 คือ Python Code สำหรับการทำความสะอาดข้อมูล ซึ่งมีขั้นตอนในการแปลงข้อมูลทั้งหมดให้เป็นข้อมูลชนิดข้อความ (String) ทั้งหมด ลบข้อมูลที่เป็นพวก ตัวอักษรพิเศษ, คำซ้ำ, คำผิด, เครื่องหมายวรรคตอน และพวกลิงก์หรือแท็กต่าง ๆ จากคำสั่ง Python Code สำหรับการทำความสะอาดข้อมูลข้างต้นจะได้ผลลัพธ์ ดังตารางที่ 3.1

ตารางที่ 3. 1 ตัวอย่างการทำความสะอาดข้อมูล

Text	Cleaning
สำหรับลูกค้าที่ไม่ต้องการทำความสะอาดบ้านด้วยตัวเองนะครับ วันนี้ทางร้านของเราขอนำเสนอ เครื่องทำความสะอาดอัตโนมัติ ดีไซน์สวยงาม แข็งแรง ถ้าลูกค้าสนใจ กดที่ตะกร้าหน้าโปรไฟล์ได้เลยค้าบ	สำหรับลูกค้าที่ไม่ต้องการทำความสะอาดบ้านด้วยตัวเองนะครับวันนี้ทางร้านของเราขอนำเสนอเครื่องทำความสะอาดอัตโนมัติดีไซน์สวยงามแข็งแรงถ้าลูกค้าสนใจกดที่ตะกร้าหน้าโปรไฟล์ได้เลยค้าบ
วัยรุ่นชเวดากองแง๊ๆๆๆน http:xxx	วัยรุ่นชเวดากองแง๊น

3.3.2 การตัดคำ (Tokenization) และการลบคำฟุ่มเฟือย (Stop Words)

```
def cleanText(text):
    text = str(text)
    text = re.sub('[^ก-๙]','',text)
    stop_word = list(thai_stopwords())
    sentence = word_tokenize(text, engine="newmm")
    result = [word for word in sentence if word not in stop_word and " " not in word]
    return ",".join(result)
def tokenize(d):
    result = d.split(",")
    result = list(filter(None, result))
    return result
Newmm = [ ]
for txt in df['cleaning']:
    Newmm.append(cleanText(txt))


vectorizer = CountVectorizer(tokenizer=tokenize)
transformed_data = vectorizer.fit_transform(Newmm)
count_data = zip(vectorizer.get_feature_names_out(), np.ravel(transformed_data.sum(axis=0)))
keyword_df = pd.DataFrame(columns=['word', 'count'])
keyword_df['word'] = vectorizer.get_feature_names_out()
keyword_df['count'] = np.ravel(transformed_data.sum(axis=0))
keyword_df.sort_values(by=['count'], ascending=False).head(10)
```

รูปที่ 3. 3 แสดง Python Code สำหรับการตัดคำ และลบคำฟุ่มเฟือย

จากรูปที่ 3.3 คือ Python Code สำหรับการตัดคำ และลบคำฟุ่มเฟือย ซึ่งมีขั้นตอนสร้างรายการคำที่เป็น Stop Words ภาษาไทย ทำการตัดคำ โดยใช้อัลกอริทึม newmm ในการตัดคำ และกรองคำที่ไม่ใช่ stop words ออกจากข้อความ แล้วทำการคืนค่าข้อความที่ผ่านการขั้นตอนการตัดคำและลบคำฟุ่มเฟือยโดยเชื่อมด้วยเครื่องหมาย “,” จากคำสั่ง Python Code สำหรับการตัดคำและลบคำฟุ่มเฟือย จะได้ผลลัพธ์ ดังตารางที่ 3.2

ตารางที่ 3. 2 ตัวอย่างการตัดคำ และลบคำฟุ่มเฟือย

Cleaning	Newmm
สำหรับลูกค้าที่ไม่ต้องการทำความสะอาดบ้านด้วยตัวเองนะครับวันนี้ทางร้านของเราขอนำเสนอเครื่องทำความสะอาดอัตโนมัติดีไซน์สวยงามแข็งแรงถ้าลูกค้าสนใจกดที่ตะกร้าหน้าโปรไฟล์ได้เลยค้าบ	['สำหรับ', 'ลูกค้า', 'ที่', 'ไม่ต้องการ', 'ทำความสะอาด', 'บ้าน', 'ด้วยตัวเอง', 'นะ', 'ครับ', 'วันนี้', 'ทาง', 'ร้าน', 'ของ', 'เรา', 'ขอ', 'นำเสนอ', 'เครื่อง', 'ทำความสะอาด', 'อัตโนมัติ', 'ดีไซน์', 'สวยงาม', 'แข็งแรง', 'ถ้า', 'ลูกค้า', 'สนใจ', 'กดที่ตะกร้า', 'หน้า', 'โปรไฟล์', 'ได้', 'เลย', 'ค้าบ']
วัยรุ่นชเวดากองแง๊น	['วัยรุ่น', 'ชเวดากอง', 'แง๊น']

3.3.3 กำหนดผลหมวดหมู่ของข้อความ

หลังจากที่ทำความสะอาดข้อมูล ตัดคำ และลบคำฟุ่มเฟือย ผู้วิจัยจะทำการกำหนดหมวดหมู่ของข้อความจากการดูภาพรวม และคำสำคัญ (Keywords) ของข้อความว่าเกี่ยวข้องกับอะไรมากที่สุด โดยผู้วิจัยจะทำการสร้าง Dictionary ที่เก็บคำสำคัญ (Keywords) ของแต่ละหมวดหมู่ (Category) ขึ้นมาเพื่อให้สอดคล้องกับนิยามสำคัญของประเด็นนั้น ๆ เพื่อให้ทราบได้ว่าลูกค้าที่เข้ามาใช้งานมีจุดประสงค์อะไร ในงานวิจัยนี้ผู้วิจัยได้เลือกหมวดหมู่ (category) ในการจำแนกข้อความไว้ 9 หมวดหมู่

ตารางที่ 3. 3 ตัวอย่าง Dictionary

หมวดหมู่ (Category)	คำสำคัญ (Keywords)
A	คำศัพท์หมวด 18+
B	ข่าว ข่าวกีฬา ข่าวบันเทิง ดราม่า
C	ข้อความที่ไม่มีคำสำคัญ (Keywords) อยู่ใน 8 หมวดหมู่อื่น ๆ
D	สล็อต เว็บพนันออนไลน์ พีเดนนี่ไก่บอลแม่นมาก
E	รีวิว ลดราคา กดที่ตะกร้า โปรโมชั่น กดติดตาม กดที่คอมเม้นได้เลย
F	คำศัพท์เกี่ยวกับการทักทาย แนะนำตัว พูดคุยถามไถ่
G	เปิดบริการทุกวัน เตรียมตัว คำเตือน ติดต่อ
H	นักเรียน คุณครู โรงเรียน ผู้อำนวยการ ห้องปกครอง นายก ราชการ
I	ตัวอย่าง ทดลอง หัวข้อ ขั้นตอน เอกสาร

หมายเหตุ : หมวดหมู่ D มีเนื้อหาที่ไม่สามารถเผยแพร่ได้

ตารางที่ 3. 4 ตัวอย่างการกำหนดหมวดหมู่ของข้อความ

Text	Category
สำหรับลูกค้าที่ไม่ต้องการทำความสะอาดบ้านด้วยตัวเองนะครับ วันนี้ทางร้านของเราขอนำเสนอ เครื่องทำความสะอาดอัตโนมัติ ดีไซน์สวยงาม แข็งแรง ถ้าลูกค้าสนใจ กดที่ตะกร้าหน้าโปรไฟล์ได้เลยค้าบ	E
ข่าวบันเทิงวันนี้ เป็นข่าวของดาราดังที่พึ่งแต่งงานไปได้ไม่นาน แต่ก็มีข่าวฉาวออกมาซะแล้วนะครับคุณผู้ชม	B

3.3.4 การสกัดคุณลักษณะ (Feature Extraction)

```
# Feature Extraction

# แบ่งข้อมูลเป็นชุดฝึกและชุดทดสอบ

X_train, X_test, y_train, y_test = train_test_split(df["Newmm"], df["Category "],
test_size=0.2, random_state=42)


# สร้าง TfidfVectorizer เพื่อแปลงข้อความเป็นเวกเตอร์ TF-IDF

tfidf_vectorizer = TfidfVectorizer(tokenizer=word_tokenize, analyzer='word',
max_features=5000)

X_train_tfidf = tfidf_vectorizer.fit_transform(X_train)

X_test_tfidf = tfidf_vectorizer.transform(X_test)
```

รูปที่ 3.4 แสดง Python Code การสกัดคุณลักษณะ

จากรูปที่ 3.4 คือ Python Code การสกัดคุณลักษณะ ซึ่งมีขั้นตอนแปลงข้อความให้เป็นตัวเลข โดยใช้วิธี TF-IDF โดยผู้วิจัยจะกำหนด ‘max_features=5000’ จะได้เวกเตอร์ TF-IDF ที่มีความยาวไม่เกิน 5000 คำ ซึ่งประกอบด้วยคำที่มีความถี่สูงสุดและมีความสำคัญสูงที่สุดในข้อความ หมายความว่า เราจะเลือกเฉพาะคำที่สำคัญที่สุด 5000 คำในข้อความมาใช้ในการสร้างเวกเตอร์ TF-IDF เท่านั้น คำที่มีความถี่น้อยลงจะไม่ถูกนำมาใช้ในเวกเตอร์ การลดจำนวนคำในเวกเตอร์จะช่วยลดขนาดของข้อมูลและประหยัดทรัพยากรคอมพิวเตอร์ได้

3.4 การออกแบบคุณลักษณะและแบบจำลอง

เนื่องจากข้อมูลทั้งหมดมีจำนวน 3,451,215 แถว ซึ่งผู้วิจัยไม่สามารถจะหาทรัพยากรที่เพียงพอสำหรับการประมวลผลข้อมูลทั้งหมดพร้อมกันได้ ผู้วิจัยจึงทำการสุ่มข้อมูลออกมา 250,000 แถว เพื่อนำมาใช้กับแบบจำลอง โดยในงานวิจัยนี้ผู้วิจัยได้แบ่งข้อมูลทดสอบ (Split Testing) ออกเป็นสองชุด คือ ข้อมูลชุดเรียนรู้ (Training Dataset) 80% และข้อมูลชุดทดสอบ (Testing Dataset) 20 ในงานวิจัยครั้งนี้ ผู้วิจัยได้ใช้ข้อมูลชุดฝึกสอนสำหรับการสร้างแบบจำลองการจำแนกประเภทข้อมูล (Classification Model) จำนวน 4 แบบจำลอง ได้แก่ 1.แบบจำลองแรนดอมฟอเรสต์ (Random Forest) 2.แบบจำลองการถดถอยแบบโลจิสติก (Logistic Regression) 3.แบบจำลองซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machine; SVM) 4.แบบจำลองนาอีฟเบย์ (Naive Bayes)

3.5 การเปรียบเทียบผลพยากรณ์ของแบบจำลอง

การเปรียบเทียบประสิทธิภาพของแบบจำลองจะใช้ค่า F1-score เป็นอันดับแรกในการวัดผล และค่า Accuracy รองลงมา แต่เนื่องจากชุดข้อมูลที่ใช้เป็นชุดข้อมูลที่ไม่สมดุล (Imbalance Data) ผู้วิจัยจึงนำค่า Macro Average มาใช้ในการวัดผลประสิทธิภาพของแบบจำลอง

3.6 เครื่องมือที่ใช้ในการวิจัย

ในการจัดเตรียมชุดข้อมูลเพื่อนำไปใช้ในการสร้างแบบจำลอง และวัดประสิทธิภาพของแบบจำลอง จะดำเนินการด้วยการใช้โปรแกรมภาษาไพธอน (Python 3) บน Colab และ Visual Studio Code (VS Code) และใช้ไลบารี (Library) ที่จำเป็นต่อการวิเคราะห์ดังตารางที่ 3.5

ตารางที่ 3. 5 ไลบรารี (Library) ที่จำเป็นต่อการวิเคราะห์

ไลบรารี (Library)	คำอธิบาย (Description)
Pandas	ใช้สำหรับการจัดการและวิเคราะห์ข้อมูล
Numpy	ใช้สำหรับการคำนวณทางคณิตศาสตร์และสถิติ
PyThaiNLP	ใช้สำหรับการประมวลผลภาษาทางธรรมชาติ
Scikit-Learn	ใช้สำหรับการแปลงเชิงปริมาณ วิธี TF-IDF ใช้สำหรับการสร้างแบบจำลองการเรียนรู้ของเครื่อง เช่น แบบจำลองแรนดอมฟอเรสต์ แบบจำลองการถดถอยแบบโลจิสติก แบบจำลองซัพพอร์ตเวกเตอร์แมชชีน แบบจำลองนาอีฟเบย์ เป็นต้น
Matplotlib	ใช้สำหรับแสดงผลข้อมูล

บทที่ 4
ผลการวิจัยและการอภิปรายผล

ในบทนี้ผู้วิจัยจะกล่าวถึงผลการจำแนกและวิเคราะห์หมวดหมู่การใช้งานสำหรับลูกค้าของแพลตฟอร์ม text to speech โดยใช้การสร้างตัวแปลงเชิงปริมาณให้อยู่ในรูปของคุณลักษณะที่ใช้ในการประมวลผลได้ วิธีเทคนิคการคัดแยกคำตามความสำคัญ (TF-IDF) จากนั้นทำการฝึกสอนโดยใช้แบบจำลองแรนดอมฟอเรสต์ (Random Forest), แบบจำลองการถดถอยแบบโลจิสติก (Logistic Regression), แบบจำลองซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machine; SVM) และแบบจำลองนาอีฟเบย์ (Naive Bayes) สำหรับเนื้อหาในบทนี้จะประกอบไปด้วย ผลการทดสอบประสิทธิภาพของแบบจำลองต่าง ๆ และการอภิปรายผล โดยมีรายละเอียดดังต่อไปนี้

4.1 ผลการทดสอบประสิทธิภาพของแบบจำลองแรนดอมฟอเรสต์ (Random Forest)

4.2 ผลการทดสอบประสิทธิภาพของแบบจำลองการถดถอยแบบโลจิสติก (Logistic Regression)

4.3 ผลการทดสอบประสิทธิภาพของแบบจำลองซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machine; SVM)

4.4 ผลการทดสอบประสิทธิภาพของแบบจำลองนาอีฟเบย์ (Naive Bayes)

4.5 ผลการทดสอบประสิทธิภาพของแบบจำลองด้วยข้อมูลชุดเรียนรู้

4.6 การเปรียบเทียบประสิทธิภาพแบบจำลองทั้งหมด

4.7 การนำแบบจำลองไปใช้งาน

4.1 ผลการทดสอบประสิทธิภาพของแบบจำลองแรนดอมฟอเรสต์ (Random Forest)

การจำแนกและวิเคราะห์หมวดหมู่การใช้งานสำหรับลูกค้าของแพลตฟอร์ม text to speech โดยใช้การสร้างตัวแปลงเชิงปริมาณ วิธีเทคนิคการคัดแยกคำตามความสำคัญ (TF-IDF) กับแบบจำลองแรนดอมฟอเรสต์ (Random Forest) จะสามารถสรุปการทำนายและทดสอบประสิทธิภาพในการทำนาย ดังตารางที่ 4.1 และ 4.2 ตามลำดับ

ตารางที่ 4. 1 เมทริกซ์ความสับสน (Confusion Matrix) ของการแปลงเชิงปริมาณ TF-IDF กับแบบจำลองแรนดอมฟอเรสต์

		แบบจำลองทำนาย									รวม
		A	B	C	D	E	F	G	H	I	
ข้อมูลชุดทดสอบ	A	**3948**	14	311	0	71	40	0	1	0	**4385**
	B	2	**2825**	271	0	164	43	0	8	1	**3314**
	C	30	12	**10330**	0	643	1	1	28	0	**11045**
	D	0	0	17	**268**	8	1	0	0	0	**294**
	E	64	168	1222	4	**8198**	193	82	73	4	**10008**
	F	2	13	359	0	282	**5277**	1	9	0	**5943**
	G	1	6	78	0	42	2	**2881**	2	0	**3012**
	H	9	21	220	0	370	27	10	**6234**	79	**6970**
	I	0	7	84	0	93	13	3	10	**4819**	**5029**
รวม		**4056**	**3066**	**12892**	**272**	**9871**	**5597**	**2978**	**6365**	**4903**	

จากตารางที่ 4.1 พบว่า แบบจำลองทำนายได้ถูกต้องมากที่สุดใน 3 หมวดหมู่นี้ ได้แก่ หมวดหมู่ C, E และ H โดยทำนายได้ถูกต้องเป็นจำนวน 10330, 8198 และ 6234 ข้อความตามลำดับ

ตารางที่ 4. 2 ประสิทธิภาพในการทำนายของการแปลงเชิงปริมาณ TF-IDF กับแบบจำลองแรน ดอมฟอเรสต์

หมวดหมู่	ประสิทธิภาพการทำนายของแบบจำลอง			
	Precision	Recall	F1-Score	Accuracy
A	0.97	0.90	0.94	**0.90**
B	0.92	0.85	0.89	
C	0.80	0.94	0.86	
D	0.99	0.91	0.95	
E	0.83	0.82	0.82	
F	0.94	0.89	0.91	
G	0.97	0.96	0.96	
H	0.98	0.89	0.93	
I	0.98	0.96	0.97	
Macro Average	**0.93**	**0.90**	**0.915**	

จากตารางที่ 4.2 แสดงประสิทธิภาพในการทำนายของการแปลงเชิงปริมาณ TF-IDF กับแบบจำลองแรนดอมฟอเรสต์ จะเห็นได้ว่ามีค่าความถ่วงดุลแบบมาโคร (Macro - F1-Score) เท่ากับ 91.5% และความแม่นยำ (Accuracy) เท่ากับ 90% และความสามารถในการทำนายหมวดหมู่ผู้ใช้งานในแต่ละหมวดหมู่ มีค่า F1-Score อยู่สูงมาก โดยที่ทุกหมวดหมู่มีค่า F1-Score มากกว่า 80% ขึ้นไปทุกตัว ซึ่งถือว่าแบบจำลองแรนดอมฟอเรสต์สามารถทำนายผลการจำแนกหมวดหมู่การใช้งานได้โดยมีความแม่นยำที่สูง

4.2 ผลการทดสอบประสิทธิภาพของแบบจำลองการถดถอยแบบโลจิสติก (Logistic Regression)

การจำแนกและวิเคราะห์หมวดหมู่การใช้งานสำหรับลูกค้าของแพลตฟอร์ม text to speech โดยใช้การสร้างตัวแปลงเชิงปริมาณ วิธีเทคนิคการคัดแยกคำตามความสำคัญ (TF-IDF) กับแบบจำลองการถดถอยแบบโลจิสติก (Logistic Regression) จะสามารถสรุปการทำนายและทดสอบประสิทธิภาพในการทำนาย ดังตารางที่ 4.3 และ 4.4 ตามลำดับ

ตารางที่ 4. 3 เมทริกซ์ความสับสน (Confusion Matrix) ของการแปลงเชิงปริมาณ TF-IDF กับแบบจำลองการถดถอยแบบโลจิสติก

		แบบจำลองทำนาย									รวม
		A	B	C	D	E	F	G	H	I	
ข้อมูลชุดทดสอบ	A	**3823**	40	318	0	124	71	0	9	0	**4385**
	B	4	**2741**	193	0	251	73	2	48	2	**3314**
	C	20	16	**10412**	0	537	0	0	51	9	**11045**
	D	0	4	15	**235**	37	1	0	1	1	**294**
	E	57	221	796	1	**8523**	183	93	107	27	**10008**
	F	0	15	252	0	346	**5284**	0	40	6	**5943**
	G	22	40	70	0	189	14	**2667**	4	6	**3012**
	H	12	54	163	1	551	33	27	**5992**	137	**6970**
	I	5	41	100	3	450	10	10	29	**4381**	**5029**
รวม		**3943**	**3172**	**12319**	**240**	**11008**	**5669**	**2799**	**6281**	**4569**	

จากตารางที่ 4.3 พบว่า แบบจำลองทำนายได้ถูกต้องมากที่สุดใน 3 หมวดหมู่นี้ ได้แก่ หมวดหมู่ C, E และ H โดยทำนายได้ถูกต้องเป็นจำนวน 10412, 8523 และ 5992 ข้อความตามลำดับ

ตารางที่ 4. 4 ประสิทธิภาพในการทำนายของการแปลงเชิงปริมาณ TF-IDF กับแบบจำลองการถดถอยแบบโลจิสติก

หมวดหมู่	ประสิทธิภาพการทำนายของแบบจำลอง			
	Precision	Recall	F1-Score	Accuracy
A	0.97	0.87	0.92	**0.88**
B	0.86	0.83	0.85	
C	0.85	0.94	0.89	
D	0.98	0.80	0.88	
E	0.77	0.85	0.81	
F	0.93	0.89	0.91	
G	0.95	0.89	0.90	
H	0.95	0.86	0.90	
I	0.96	0.87	0.91	
Macro Average	**0.91**	**0.87**	**0.89**	

จากตารางที่ 4.4 แสดงประสิทธิภาพในการทำนายของการแปลงเชิงปริมาณ TF-IDF กับแบบจำลองการถดถอยแบบโลจิสติก จะเห็นได้ว่ามีค่าความถ่วงดุลแบบมาโคร (Macro - F1-Score) เท่ากับ 89% และความแม่นยำ (Accuracy) เท่ากับ 88% และความสามารถในการทำนายหมวดหมู่ผู้ใช้งานในแต่ละหมวดหมู่ มีค่า F1-Score อยู่สูงมาก โดยที่ทุกหมวดหมู่มีค่า F1-Score มากกว่า 80% ขึ้นไปทุกตัว แต่ไม่มีหมวดหมู่ไหนที่ค่า F1-Score มากกว่า 95% ซึ่งถือว่าแบบจำลองการถดถอยแบบโลจิสติกสามารถทำนายผลการจำแนกหมวดหมู่การใช้งานได้โดยมีความแม่นยำที่สูงแต่มีความแม่นยำน้อยกว่าแบบจำลองแรนดอมฟอเรสต์

4.3 ผลการทดสอบประสิทธิภาพของแบบจำลองซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machine; SVM)

การจำแนกและวิเคราะห์หมวดหมู่การใช้งานสำหรับลูกค้าของแพลตฟอร์ม text to speech โดยใช้การสร้างตัวแปลงเชิงปริมาณ วิธีเทคนิคการคัดแยกคำตามความสำคัญ (TF-IDF) กับแบบจำลองซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machine; SVM) จะสามารถสรุปการทำนายและทดสอบประสิทธิภาพในการทำนาย ดังตารางที่ 4.5 และ 4.6 ตามลำดับ

ตารางที่ 4. 5 เมทริกซ์ความสับสน (Confusion Matrix) ของการแปลงเชิงปริมาณ TF-IDF กับแบบจำลองซัพพอร์ตเวกเตอร์แมชชีน

		แบบจำลองทำนาย									รวม
		A	B	C	D	E	F	G	H	I	
ข้อมูลชุดทดสอบ	A	**3921**	38	288	1	81	55	0	1	0	**4385**
	B	7	**2937**	150	0	175	41	0	4	0	**3314**
	C	34	25	**10579**	0	399	0	1	7	0	**11045**
	D	0	0	10	**267**	17	0	0	0	0	**294**
	E	82	235	676	7	**8706**	187	102	12	1	**10008**
	F	3	24	245	0	272	**5394**	1	4	0	**5943**
	G	12	25	56	0	94	6	**2818**	1	0	**3012**
	H	16	58	123	3	418	29	30	**6172**	121	**6970**
	I	8	31	75	5	224	9	13	8	**4656**	**5029**
รวม		**4083**	**3373**	**12202**	**283**	**10386**	**5721**	**2965**	**6209**	**4778**	

จากตารางที่ 4.5 พบว่า แบบจำลองทำนายได้ถูกต้องมากที่สุดใน 3 หมวดหมู่นี้ ได้แก่ หมวดหมู่ C, E และ H โดยทำนายได้ถูกต้องเป็นจำนวน 10579, 8706 และ 6172 ข้อความตามลำดับ

ตารางที่ 4. 6 ประสิทธิภาพในการทำนายของการแปลงเชิงปริมาณ TF-IDF กับแบบจำลองซัพพอร์ตเวกเตอร์แมชชีน

หมวดหมู่	ประสิทธิภาพการทำนายของแบบจำลอง			
	Precision	Recall	F1-Score	Accuracy
A	0.96	0.89	0.93	**0.91**
B	0.87	0.89	0.88	
C	0.87	0.96	0.91	
D	0.94	0.91	0.93	
E	0.84	0.87	0.85	
F	0.94	0.91	0.92	
G	0.95	0.94	0.94	
H	0.99	0.89	0.94	
I	0.97	0.93	0.95	
Macro Average	**0.93**	**0.91**	**0.916**	

จากตารางที่ 4.6 แสดงประสิทธิภาพในการทำนายของการแปลงเชิงปริมาณ TF-IDF กับแบบจำลองซัพพอร์ตเวกเตอร์แมชชีน จะเห็นได้ว่ามีค่าความถ่วงดุลแบบมาโคร (Macro - F1-Score)

เท่ากับ 91.6% และความแม่นยำ (Accuracy) เท่ากับ 91% และความสามารถในการทำนายหมวดหมู่ผู้ใช้งานในแต่ละหมวดหมู่ มีค่า F1-Score อยู่สูงมาก โดยที่ทุกหมวดหมู่มีค่า F1-Score มากกว่า 80% ขึ้นไปทุกตัว ซึ่งถือว่าแบบจำลองซัพพอร์ตเวกเตอร์แมชชีนสามารถทำนายผลการจำแนกหมวดหมู่การใช้งานได้โดยมีความแม่นยำที่สูง

4.4 ผลการทดสอบประสิทธิภาพของแบบจำลองนาอีฟเบย์ (Naive Bayes)

การจำแนกและวิเคราะห์หมวดหมู่การใช้งานสำหรับลูกค้าของแพลตฟอร์ม text to speech โดยใช้การสร้างตัวแปลงเชิงปริมาณ วิธีเทคนิคการคัดแยกคำตามความสำคัญ (TF-IDF) กับแบบจำลองนาอีฟเบย์ (Naive Bayes) จะสามารถสรุปการทำนายและทดสอบประสิทธิภาพในการทำนาย ดังตารางที่ 4.7 และ 4.8 ตามลำดับ

ตารางที่ 4. 7 เมทริกซ์ความสับสน (Confusion Matrix) ของการแปลงเชิงปริมาณ TF-IDF กับแบบจำลองนาอีฟเบย์

		แบบจำลองทำนาย									รวม
		A	B	C	D	E	F	G	H	I	
ข้อมูลชุดทดสอบ	A	**3736**	40	310	0	175	64	4	23	33	**4385**
	B	79	**1739**	376	1	517	106	11	425	60	**3314**
	C	334	105	**7380**	2	1553	342	65	764	500	**11045**
	D	5	4	35	**101**	117	6	0	8	18	**294**
	E	110	266	942	2	**6937**	163	59	782	747	**10008**
	F	144	68	306	0	610	**4460**	33	250	72	**5943**
	G	42	53	224	0	511	55	**1861**	84	182	**3012**
	H	18	55	272	1	634	79	32	**5677**	202	**6970**
	I	2	47	281	1	814	20	22	422	**3420**	**5029**
รวม		**4470**	**2377**	**10126**	**108**	**11868**	**5295**	**2087**	**8435**	**5234**	

จากตารางที่ 4.7 พบว่า แบบจำลองทำนายได้ถูกต้องมากที่สุดใน 3 หมวดหมู่นี้ ได้แก่ หมวดหมู่ C, E และ H โดยทำนายได้ถูกต้องเป็นจำนวน 7380, 6937 และ 5677 ข้อความ ตามลำดับ

ตารางที่ 4. 8 ประสิทธิภาพในการทำนายของการแปลงเชิงปริมาณ TF-IDF กับแบบจำลองนาอีฟเบย์

หมวดหมู่	ประสิทธิภาพการทำนายของแบบจำลอง			
	Precision	Recall	F1-Score	Accuracy
A	0.84	0.85	0.84	
B	0.73	0.52	0.61	
C	0.73	0.67	0.70	
D	0.94	0.34	0.50	
E	0.58	0.69	0.63	**0.71**
F	0.84	0.75	0.79	
G	0.89	0.62	0.73	
H	0.67	0.81	0.74	
I	0.65	0.68	0.67	
Macro Average	**0.76**	**0.66**	**0.69**	

จากตารางที่ 4.8 แสดงประสิทธิภาพในการทำนายของการแปลงเชิงปริมาณ TF-IDF กับแบบจำลองนาอีฟเบย์จะเห็นได้ว่ามีค่าความถ่วงดุลแบบมาโคร (Macro - F1-Score) เท่ากับ 69% และความแม่นยำ (Accuracy) เท่ากับ 71% ซึ่งน้อยมาก ๆ ถ้าเทียบกับแบบจำลองตัวอื่นทั้งหมด และความสามารถในการทำนายหมวดหมู่ผู้ใช้งานในแต่ละหมวดหมู่ มีค่า F1-Score อยู่ค่อนข้างต่ำถ้าเทียบกับแบบจำลองอื่น โดยมีแค่หมวดหมู่เดียวที่มีค่า F1-Score มากกว่า 80% ขึ้นไป ซึ่งถือว่าแบบจำลองนาอีฟเบย์สามารถทำนายผลการจำแนกหมวดหมู่การใช้งานได้โดยมีความแม่นยำที่ต่ำ

4.5 ผลการทดสอบประสิทธิภาพของแบบจำลองด้วยข้อมูลชุดเรียนรู้

เนื่องจากผู้วิจัยต้องการเปรียบเทียบประสิทธิภาพของแบบจำลองทั้งหมด เพื่อหาว่าอาจเกิดปัญหาพอดีเกินไป (Overfitting) ของแบบจำลอง ผู้วิจัยจึงทำการจำแนกและวิเคราะห์หมวดหมู่การใช้งานสำหรับลูกค้าของแพลตฟอร์ม text to speech โดยใช้การสร้างตัวแปลงเชิงปริมาณ วิธีเทคนิคการคัดแยกคำตามความสำคัญ (TF-IDF) กับแบบจำลองแรนดอมฟอเรสต์ (Random Forest) แบบจำลองการถดถอยแบบโลจิสติก (Logistic Regression) แบบจำลองซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machine; SVM) และแบบจำลองนาอีฟเบย์ (Naive Bayes) โดยใช้ข้อมูลชุดเรียนรู้ จะสามารถสรุปการทำนายและทดสอบประสิทธิภาพในการทำนาย ดังตารางที่ 4.9, 4.10, 4.11, 4.12, 4.13, 4.14, 4.15 และ 4.16 ตามลำดับ

ตารางที่ 4. 9 เมทริกซ์ความสับสน (Confusion Matrix) ของการแปลงเชิงปริมาณ TF-IDF กับแบบจำลองแรนดอมฟอเรสต์ด้วยข้อมูลชุดเรียนรู้

		แบบจำลองทำนาย									**รวม**
		A	B	C	D	E	F	G	H	I	
ข้อมูลชุดเรียนรู้	A	16437	1	1016	0	89	72	0	0	0	**17615**
	B	0	12018	1042	0	85	41	0	0	0	**13186**
	C	5	0	43683	0	267	0	0	0	0	**43955**
	D	0	0	66	1137	3	0	0	0	0	**1206**
	E	4	2	3912	0	35839	231	4	0	0	**39992**
	F	0	0	1186	0	180	22691	0	0	0	**24057**
	G	0	0	274	0	25	0	11689	0	0	**11988**
	H	1	0	659	0	131	22	1	27215	1	**28030**
	I	0	0	279	0	36	6	0	0	19650	**19971**
รวม		**16447**	**12021**	**52117**	**1137**	**36655**	**23063**	**11694**	**27215**	**19651**	

ตารางที่ 4. 10 ประสิทธิภาพในการทำนายของการแปลงเชิงปริมาณ TF-IDF กับแบบจำลองแรนดอมฟอเรสต์ด้วยข้อมูลชุดเรียนรู้

หมวดหมู่	**ประสิทธิภาพการทำนายของแบบจำลอง**			
	Precision	Recall	F1-Score	Accuracy
A	1.00	0.93	0.97	**0.95**
B	1.00	0.91	0.95	
C	0.84	0.99	0.91	
D	1.00	0.94	0.97	
E	0.95	0.90	0.94	
F	0.98	0.94	0.96	
G	1.00	0.98	0.99	
H	1.00	0.97	0.99	
I	1.00	0.98	0.99	
Macro Average	**0.98**	**0.95**	**0.96**	

ตารางที่ 4. 11 เมทริกซ์ความสับสน (Confusion Matrix) ของการแปลงเชิงปริมาณ TF-IDF กับแบบจำลองการถดถอยแบบโลจิสติกด้วยข้อมูลชุดเรียนรู้

		แบบจำลองทำนาย									**รวม**
		A	B	C	D	E	F	G	H	I	
ข้อมูลชุดเรียนรู้	A	15548	110	1078	3	583	265	0	28	0	**17615**
	B	4	11242	751	0	842	216	6	113	12	**13186**
	C	100	49	41562	3	2062	0	9	138	32	**43955**
	D	0	20	71	1010	94	8	0	1	2	**1206**
	E	257	797	3020	26	34553	602	334	331	72	**39992**
	F	13	57	918	0	1390	21517	7	133	22	**24057**
	G	49	158	245	0	631	55	10800	33	17	**11988**
	H	56	215	534	3	2137	116	87	24367	515	**28030**
	I	14	157	335	18	1695	51	62	104	17535	**19971**
รวม		**16041**	**12805**	**48514**	**1063**	**43987**	**22830**	**11305**	**25248**	**18207**	

ตารางที่ 4. 12 ประสิทธิภาพในการทำนายของการแปลงเชิงปริมาณ TF-IDF กับแบบจำลองการถดถอยแบบโลจิสติกด้วยข้อมูลชุดเรียนรู้

หมวดหมู่	**ประสิทธิภาพการทำนายของแบบจำลอง**			
	Precision	Recall	F1-Score	Accuracy
A	0.97	0.88	0.92	**0.89**
B	0.88	0.85	0.87	
C	0.86	0.95	0.90	
D	0.95	0.84	0.89	
E	0.79	0.86	0.82	
F	0.94	0.89	0.92	
G	0.96	0.90	0.93	
H	0.97	0.87	0.91	
I	0.96	0.88	0.92	
Macro Average	**0.92**	**0.88**	**0.90**	

ตารางที่ 4. 13 เมทริกซ์ความสับสน (Confusion Matrix) ของการแปลงเชิงปริมาณ TF-IDF กับแบบจำลองซัพพอร์ตเวกเตอร์แมชชีนด้วยข้อมูลชุดเรียนรู้

		ทิศทางที่แบบจำลองทำนาย									รวม
		A	B	C	D	E	F	G	H	I	
ข้อมูลชุดเรียนรู้	A	16056	104	956	5	281	210	0	3	0	**17615**
	B	6	12032	541	0	503	93	0	11	0	**13186**
	C	136	42	42304	0	1455	0	3	13	2	**43955**
	D	0	8	22	1155	21	0	0	0	0	**1206**
	E	269	770	2457	27	35518	545	379	25	2	**39992**
	F	18	66	832	0	918	22203	3	17	0	**24057**
	G	37	78	171	1	241	16	11439	5	0	**11988**
	H	63	195	361	7	1515	85	103	25314	387	**28030**
	I	16	81	243	17	768	26	56	19	18745	**19971**
รวม		**16601**	**13376**	**47887**	**1212**	**41220**	**23178**	**11983**	**25407**	**19136**	

ตารางที่ 4. 14 ประสิทธิภาพในการทำนายของการแปลงเชิงปริมาณ TF-IDF กับแบบจำลองซัพพอร์ตเวกเตอร์แมชชีนด้วยข้อมูลชุดเรียนรู้

หมวดหมู่	ประสิทธิภาพการทำนายของแบบจำลอง			
	Precision	Recall	F1-Score	Accuracy
A	0.97	0.91	0.94	**0.92**
B	0.90	0.91	0.91	
C	0.88	0.96	0.92	
D	0.95	0.96	0.96	
E	0.86	0.89	0.87	
F	0.96	0.92	0.94	
G	0.95	0.95	0.95	
H	1.00	0.90	0.95	
I	0.98	0.94	0.96	
Macro Average	**0.94**	**0.93**	**0.93**	

ตารางที่ 4. 15 เมทริกซ์ความสับสน (Confusion Matrix) ของการแปลงเชิงปริมาณ TF-IDF กับแบบจำลองนาอีฟเบย์ด้วยข้อมูลชุดเรียนรู้

		ทิศทางที่แบบจำลองทำนาย									รวม
		A	B	C	D	E	F	G	H	I	
ข้อมูลชุดเรียนรู้	A	15184	132	1178	0	630	238	9	108	136	**17615**
	B	286	7462	1408	2	1842	399	59	1495	233	**13186**
	C	1258	436	30240	2	5594	1142	317	3000	1966	**43955**
	D	14	22	124	471	411	35	0	44	85	**1206**
	E	441	978	3690	4	28380	596	243	2829	2831	**39992**
	F	557	247	1226	0	2177	18446	150	933	321	**24057**
	G	170	232	907	0	1909	165	7561	445	599	**11988**
	H	82	211	976	4	2447	221	114	23043	932	**28030**
	I	14	183	959	2	3270	52	88	1663	13740	**19971**
รวม		**18006**	**9903**	**40708**	**485**	**46660**	**21294**	**8541**	**33560**	**20843**	

ตารางที่ 4. 16 ประสิทธิภาพในการทำนายของการแปลงเชิงปริมาณ TF-IDF กับแบบจำลองนาอีฟเบย์ด้วยข้อมูลชุดเรียนรู้

หมวดหมู่	ประสิทธิภาพการทำนายของแบบจำลอง			
	Precision	Recall	F1-Score	Accuracy
A	0.84	0.86	0.85	**0.72**
B	0.75	0.57	0.65	
C	0.74	0.69	0.71	
D	0.97	0.39	0.56	
E	0.61	0.71	0.66	
F	0.87	0.77	0.81	
G	0.89	0.63	0.74	
H	0.69	0.82	0.75	
I	0.66	0.69	0.67	
Macro Average	**0.78**	**0.68**	**0.71**	

ตารางที่ 4. 17 เปรียบเทียบประสิทธิภาพของแบบจำลองทั้ง 4 แบบ กับข้อมูลชุดเรียนรู้และข้อมูลชุดทดสอบ ด้วยค่าความถ่วงดุลแบบมาโคร (Macro - F1-Score)

แบบจำลอง	**ค่าความถ่วงดุลแบบมาโคร**	
	ข้อมูลชุดเรียนรู้	ข้อมูลชุดทดสอบ
Random Forest	0.96	0.915
Logistic Regression	0.90	0.89
SVM	0.93	0.916
Naive Bayes	0.71	0.69

ตารางที่ 4. 18 เปรียบเทียบประสิทธิภาพของแบบจำลองทั้ง 4 แบบ กับข้อมูลชุดเรียนรู้และข้อมูลชุดทดสอบ ด้วยค่าความแม่นยำ (Accuracy)

แบบจำลอง	**ค่าความแม่นยำ**	
	ข้อมูลชุดเรียนรู้	ข้อมูลชุดทดสอบ
Random Forest	0.95	0.90
Logistic Regression	0.89	0.88
SVM	0.92	0.91
Naive Bayes	0.72	0.71

จากตารางที่ 4.17 และ 4.18 จะสังเกตได้ว่าชุดข้อมูลการเรียนรู้มีค่าใกล้เคียงกับชุดข้อมูลทดสอบ ซึ่งแสดงได้ว่าแบบจำลองทั้งหมดไม่เกิดปัญหา Overfitting

4.6 การเปรียบเทียบประสิทธิภาพแบบจำลองทั้งหมด

ผู้วิจัยจะทำการเปรียบเทียบประสิทธิภาพของแบบจำลองทั้งหมด โดยการเปรียบเทียบประสิทธิภาพของแต่ละแบบจำลองจะดูจากค่าความถ่วงดุลแบบมาโคร (Macro - F1-Score) เป็นหลักและดูค่าความแม่นยำ (Accuracy) ควบคู่ไปด้วย จะสามารถดูผลการทดสอบประสิทธิภาพของแบบจำลองทั้งหมดดังตารางที่ 4.19

ตารางที่ 4. 19 เปรียบเทียบประสิทธิภาพของแบบจำลองทั้ง 4 แบบ ด้วยค่าความถ่วงดุลแบบมาโคร (Macro - F1-Score) และค่าความแม่นยำ (Accuracy)

แบบจำลอง	ข้อมูลชุดทดสอบ	
	ค่าความถ่วงดุลแบบมาโคร	ค่าความแม่นยำ
Random Forest	0.915	0.90
Logistic Regression	0.89	0.88
SVM	0.916	0.91
Naive Bayes	0.69	0.71

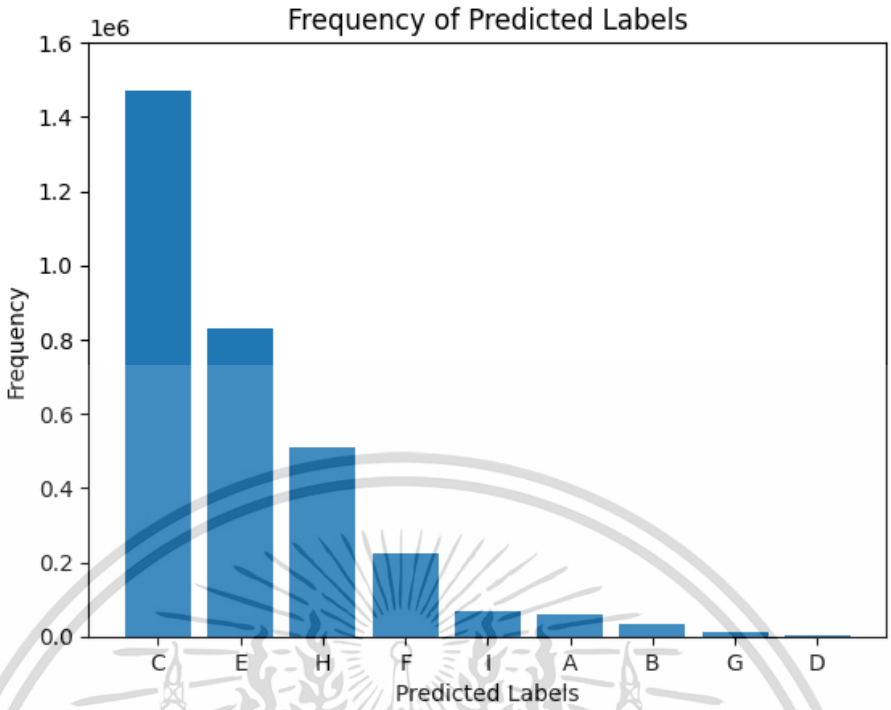
จากการวัดประสิทธิภาพของแบบจำลองทั้งหมด หากพิจารณาจากค่าความถ่วงดุลแบบมาโคร (Macro - F1-Score) พบว่า แบบจำลองซัพพอร์ตเวกเตอร์แมชชีน (SVM) มีค่าความถ่วงดุลแบบมาโครดีที่สุดคือ 91.6% รองลงมา คือ แบบจำลองแรนดอมฟอเรสต์ (Random Forest) และแบบจำลองการถดถอยแบบโลจิสติก (Logistic Regression) มีค่าเท่ากับ 91.5% และ 89% ตามลำดับ โดยที่แบบจำลองนาอีฟเบย์ (Naive Bayes) มีค่าความถ่วงดุลแบบมาโครน้อยที่สุดคือ 69%

และถ้าพิจารณาจากค่าความแม่นยำ (Accuracy) พบว่า แบบจำลองซัพพอร์ตเวกเตอร์แมชชีน (SVM) มีความแม่นยำในการจำแนกหมวดหมู่การใช้งานได้ดีที่สุดคือ 91% รองลงมา คือ แบบจำลองแรนดอมฟอเรสต์ (Random Forest) และแบบจำลองการถดถอยแบบโลจิสติก (Logistic Regression) มีค่าเท่ากับ 90% และ88% ตามลำดับ โดยที่แบบจำลองนาอีฟเบย์ (Naive Bayes) มีความแม่นยำในการจำแนกหมวดหมู่การใช้งานน้อยที่สุดคือ 81%

ดังนั้นแบบจำลองที่ดีที่สุดจากการพิจารณาจากค่าความถ่วงดุลแบบมาโคร (Macro - F1-Score) และค่าความแม่นยำ (Accuracy) ร่วมกัน คือ แบบจำลองซัพพอร์ตเวกเตอร์แมชชีน (SVM) รองลงมา คือ แบบจำลองแรนดอมฟอเรสต์ (Random Forest)

4.7 การนำแบบจำลองไปใช้งาน

นำข้อมูลของลูกค้าที่เข้ามาใช้งานในแพลตฟอร์ม text to speech ตั้งแต่วันที่ 1 มกราคม พ.ศ.2566 ถึง วันที่ 10 มิถุนายน พ.ศ.2566 ที่เหลือจำนวน 3,201,215 ตัวนำไปใช้กับแบบจำลองที่ถูกสร้างขึ้นมา โดยจะได้ผลลัพธ์ดังรูปที่ 4.1



รูปที่ 4. 1 กราฟแสดงความถี่ของหมวดหมู่การใช้งานของลูกค้า (ความถี่ (Frequency) มีหน่วยเป็น ล้านครั้ง)

จากรูปที่ 4.1 สามารถอธิบายได้ว่า ลูกค้าที่เข้ามาใช้งานแพลตฟอร์ม text to speech เข้ามาใช้งานในหมวดหมู่ C เยอะที่สุด ถึง 1,469,347 ครั้ง เนื่องจากแพลตฟอร์ม text to speech ที่ผู้วิจัยใช้นี้ เป็นแพลตฟอร์ม text to speech ที่เปิดให้ทดลองใช้ได้ฟรีตามจำนวนที่ทางบริษัทกำหนด และมีตัวเลือกเพิ่มเติมสำหรับลูกค้าที่ซื้อแพคเก็จเสริมเพิ่มเติม และมีลูกเล่นอีกมากมาย ลูกค้าจึงอาจเข้ามาทดลองใช้ โดยพิมพ์คำที่ไม่มีความหมาย ไม่มีใจความสำคัญอะไร หรือทดลองเสียงของบอทรองลงมา คือ หมวดหมู่ E จำนวน 828,174 ครั้ง แสดงว่าลูกค้าที่เข้ามาใช้งานส่วนใหญ่อาจเป็นลูกค้ากลุ่มที่เป็นพ่อค้าแม่ค้าออนไลน์ มาขายสินค้าในแพลตฟอร์มต่าง ๆ หรืออาจจะเป็นอินฟลูเอนเซอร์ (Influencer) ที่เข้ามาใช้งานเพื่อนำไปสร้างคอนเทนต์ต่าง ๆ เช่น รีวิวสินค้า รีวิวหนัง แนะนำร้านอาหาร เป็นต้น หรืออาจจะเป็นบริษัทที่เริ่มการขายผลิตภัณฑ์สินค้าต่าง ๆ ทางออนไลน์หรือสำหรับการโฆษณา อันดับที่สามที่ลูกค้าเข้ามาใช้งาน คือ หมวดหมู่ H จำนวน 508,308 ครั้ง อาจเป็นลูกค้าประเภท ครู อาจารย์ นักเรียน ที่เข้ามาใช้งานเพื่อทำหนังสือเรียนออนไลน์ ทำคลิปเฉลยการบ้านนักเรียน ทำการบ้านส่งอาจารย์ หรือทำคลิปโปรโมตต่าง ๆ ให้โรงเรียน และหมวดหมู่ที่มีผู้เข้าใช้งานน้อยที่สุดคือ หมวดหมู่ D จำนวน 1245 ครั้ง (หมายเหตุ : หมวดหมู่ D มีเนื้อหาที่ไม่สามารถเผยแพร่ได้)

บทที่ 5
สรุปผลการวิจัยและข้อเสนอแนะ

งานวิจัยนี้ได้พัฒนาแบบจำลองในการจำแนกและวิเคราะห์หมวดหมู่การใช้งานสำหรับลูกค้าของแพลตฟอร์ม text to speech วัดประสิทธิภาพแบบจำลอง และเมื่อทราบจุดประสงค์ของการเข้ามาใช้งานแล้ว สามารถนำไปปรับกลยุทธ์การตลาดและโปรโมชั่นให้เหมาะสมกับแต่ละกลุ่มของลูกค้าได้ จึงสามารถสรุปผลการดำเนินงานและข้อเสนอแนะ ดังนี้

5.1 สรุปผลการวิจัย

ตารางที่ 5. 1 สรุปผลการเปรียบเทียบประสิทธิภาพของแบบจำลองทั้ง 4 แบบ ด้วยค่าความถ่วงดุลแบบมาโคร (Macro - F1-Score)

แบบจำลอง	**ค่าความถ่วงดุลแบบมาโคร**	แบบจำลองที่เหมาะสมที่สุด
Random Forest	0.915	SVM
Logistic Regression	0.89	
SVM	0.916	
Naive Bayes	0.69	

จากตาราง 5.1 พบว่า แบบจำลองที่ดีที่สุดสำหรับการจำแนกและวิเคราะห์หมวดหมู่การใช้งานสำหรับลูกค้าของแพลตฟอร์ม text to speech โดยใช้ค่าความถ่วงดุลแบบมาโคร (Macro - F1-Score) ในการวัดผล คือ แบบจำลองซัพพอร์ตเวกเตอร์แมชชีน (SVM)

ตารางที่ 5. 2 สรุปผลการเปรียบเทียบประสิทธิภาพของแบบจำลองทั้ง 4 แบบ ด้วยค่าความแม่นยำ (Accuracy)

แบบจำลอง	**ค่าความแม่นยำ**	แบบจำลองที่เหมาะสมที่สุด
Random Forest	0.90	SVM
Logistic Regression	0.88	
SVM	0.91	
Naive Bayes	0.71	

จากตาราง 5.2 พบว่า แบบจำลองที่ดีที่สุดสำหรับการจำแนกและวิเคราะห์หมวดหมู่การใช้งานสำหรับลูกค้าของแพลตฟอร์ม text to speech โดยใช้ค่าความแม่นยำ (Accuracy) ในการวัดผล คือ แบบจำลองซัพพอร์ตเวกเตอร์แมชชีน (SVM)

แสดงให้เห็นว่า สำหรับงานวิจัยนี้ แบบจำลองซัพพอร์ตเวกเตอร์แมชชีน (SVM) คือแบบจำลองที่มีประสิทธิภาพดีที่ และเป็นแบบจำลองที่เหมาะสมที่สุดสำหรับการจำแนกและวิเคราะห์หมวดหมู่การใช้งานสำหรับลูกค้าของแพลตฟอร์ม text to speech

ผลจากการนำแบบจำลองไปใช้งานพบว่า ลูกค้าที่เข้ามาใช้งานแพลตฟอร์ม text to speech เยอะที่สุดคือ ลูกค้าหมวดหมู่ C เข้ามาใช้งานจำนวน 1,469,347 ครั้ง และลูกค้าที่เข้ามาใช้งานแพลตฟอร์ม text to speech น้อยที่สุด คือลูกค้าหมวดหมู่ D เข้ามาใช้งานจำนวน 1,245 ครั้ง

5.2 ข้อเสนอแนะ

จากงานวิจัยครั้งนี้ผู้ที่สนใจสามารถนำไปศึกษาต่อในเรื่องต่อไปนี้

5.2.1 งานวิจัยชิ้นนี้ใช้การแปลงเชิงปริมาณแค่ 1 วิธี คือ วิธี TF-IDF ซึ่งยังมีวิธีการแปลงเชิงปริมาณที่น่าสนใจอีกจำนวนมาก เช่น Bag of Word, Word2Vec, Glove และ Bert เป็นต้น ในงานวิจัยถัดไปจึงสามารถนำการแปลงเชิงปริมาณข้างต้นมาใช้กับการจำแนกและวิเคราะห์หมวดหมู่การใช้งานสำหรับลูกค้าของแพลตฟอร์ม text to speech ได้

5.2.2 เนื่องจากข้อมูลที่ใช้ในงานวิจัยนี้มีขนาดใหญ่มาก การใช้แบบจำลองซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machine; SVM) กับงานวิจัยนี้ทำให้ใช้เวลาในการฝึกสอนแบบจำลองเป็นเวลานานมาก ถ้าเทียบกับแบบจำลองอีก 3 ตัว ในงานวิจัย ในงานวิจัยถัดไปในอนาคตถ้าอยากให้แบบจำลองใช้เวลาในการฝึกสอนน้อยลงและมีความแม่นยำมากขึ้น อาจจะต้องศึกษาการเรียนรู้ของเครื่อง (Machine Learning) เพิ่มเติมมากขึ้น

เอกสารอ้างอิง

กานดา แผ่วัฒนากุล. 2566. **การวิเคราะห์เหมืองข้อเสนอแนะจากบทวิจารณ์รายการโทรทัศน์.** วิทยาศาสตรมหาบัณฑิต สาขาวิชาสถิติประยุกต์. สถาบันบัณฑิตพัฒนบริหารศาสตร์

จิรวัฒน์ จันทองพูน, พัณณิตา ไล่สาม, สัณห์ฤทัย แซ่หว่อง, และ พรนรายณ์ บุญราศรี. **การศึกษาการขยายตัวของเมืองด้วยเทคนิควิธี Random Forest กรณีศึกษา อำเภอเมืองสงขลา จังหวัดสงขลา.** 24-26 ในการประชุมวิชาการวิศวกรรมโยธาแห่งชาติ ครั้งที่ 27. สงขลา.

จุฑาทิพย์ ทิพย์พูล และนิเวศ จิระวิชิตชัย. 2559. **“การจำแนกจดหมายอิเล็กทรอนิกส์ที่เป็นสแปมโดยใช้เทคนิคเหมืองข้อมูล.”** วารสารวิทยาศาสตร์และเทคโนโลยี มทร.ธัญบุรี 6, 1: 102-109.

ชิตพงษ์ กิตตินราดร. 2562. **Categorical Encoding.** [ออนไลน์]. เข้าถึงได้จาก: https://guopai.github.io/ml-blog05.html

เทิดศักดิ์ เงินมูล, พิเชษฐ เหมยคำ, วิโรจน์ ปงลังกา และวิวัฒน์ ทิพจร. 2560. **การคัดแยกความสุกสตรอเบอรี่ด้วยซัพพอร์ตเวกเตอร์แมชชีน.** Naresuan University Engineering Journal. 12(2) : 55-62

ธนาภัทร ภัทรวินิจ. 2563. **การทำนายความผิดพลาดระยะต้นของเครื่องวิเคราะห์อินทรีย์คาร์บอนโดยการเรียนรู้เชิงลึก.** วิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์. จุฬาลงกรณ์มหาวิทยาลัย.

ปณยา สุดตา. 2566. **นักการตลาดกับ AI: Confusion Matrix และหลักการประเมินประสิทธิภาพ ML Model.** [ออนไลน์]. เข้าถึงได้จาก: https://www.everydaymarketing.co/business-and-marketing-case-study/ai/confusion-matrix-ml-evaluation/

พศสรัล อภิปวินท์วงศา. 2565. **เทคนิคการเรียนรู้เชิงลึกสำหรับการรู้จำภาพวัสดุกระเป๋าถือแบรนด์เนมปลอม.** วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์. จุฬาลงกรณ์มหาวิทยาลัย

ยุวดี เปรมวิชัย. 2564. **Data Analytics: Prediction with Logistic Regression.** [ออนไลน์] เข้าถึงได้จาก: https://www.mebmarket.com/ebook-153641-Data-Analytics-Prediction-with-Logistic-Regression

วัชรีวรรณ จิตต์สกุล. 2560. **การวิเคราะห์การจำแนกข้อความด้วยการเปรียบเทียบความเสถียรของอัลกอริทึม.** คณะเทคโนโลยีสารสนเทศ. มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

สุภาภรณ์ พัฒนวงศ์ปราการ. 2563. **การวิเคราะห์เทคนิคการจำแนกประเภทข้อมูลกรณีศึกษาการพยากรณ์ระดับชั้นผู้รับเหมาก่อสร้างสำหรับโครงการก่อสร้างของภาครัฐ.** วิทยาศาสตรมหาบัณฑิต สาขาวิชาระบบสารสนเทศเพื่อการจัดการ. มหาวิทยาลัยธรรมมศาสตร์

สถาบันนวัตกรรมและธรรมาภิบาลข้อมูล. 2565. **เข้าใจใน 5 นาที! Classification Model คืออะไร.** [ออนไลน์]. เข้าถึงได้จาก: https://digi.data.go.th/blog/what-is-classification-model/

เอกสารอ้างอิง (ต่อ)

เอกรัฐ และภราดร. 2564. **การจำแนกประเภทข้อความความคิดเห็นที่เป็นการระรานทางไซเบอร์ในสื่อสังคมออนไลน์**. สถิติประยุกต์. สถาบันบัณฑิตพัฒนบริหารศาสตร์

AWS. 2567. **การประมวลผลภาษาธรรมชาติคืออะไร - อธิบาย NLP – AWS.** [ออนไลน์]. เข้าถึงได้จาก: https://aws.amazon.com/th/what-is/nlp/

BDI - Big Data Institute. 2565. **สร้าง AI เข้าใจภาษามนุษย์ด้วย Natural Language Processing. [ออนไลน์].** เข้าถึงได้จาก: https://bdi.or.th/big-data-101/what-is-natural-language-processing/

Botnoi Group. 2567. **Text-To-Speech (TTS) คืออะไร?** [ออนไลน์]. เข้าถึงได้จาก: https://botnoigroup.com/th/blog/about-text-to-speech

SAS. 2563. **การประมวลภาษาธรรมชาติ.** [ออนไลน์]. เข้าถึงได้จาก https://www.sas.com/th_th/insights/analytics/what-is-natural-language-processing-nlp.html

sklsongkiat. 2565. **Overfitting Underfitting วิธีหลีกเลี่ยงและการป้องกันทั้ง 7.** [ออนไลน์]. เข้าถึงได้จาก: https://www.sklsongkiat.com/articles/detail/overfitting-underfitting

Akanksha Patro, Mahima Patel, Richa Shukla, & Jagurti Save. 2020. **Real Time News Classification Using Machine Learning.** International Journal of Advanced Science and Technology, 29, 620-630.

Cai, S., Palazoglu, A., Zhang, L., & Hu, J. (2019). **Process alarm prediction using deep learning and word embedding methods.** ISA Transactions, 85, 274-283.

Dietrich, D. , Heller, B. and Yang, B. 2015. **Data science & big data analytics discovering, analyzing, visualizing and presenting data.** Indiana. john wiley & sons,Inc.

Haddi, E., Liu, X., & Shi, Y. (2013). **The role of text pre-processing in sentiment analysis.** Department of Information System and Computing. Brunel University.

Korstanje, J. (2021). **The F1 score.** [Online] Available: https://towardsdatascience.com/the-f1-score-bec2bbc38aa6

Kevin Markham. 2014. **Simple guide to confusion matrix terminology.** [Online] Available :https://www.dataschool.io/simple-guide-to-confusion-matrix-terminology/

M.Khavitha & Dr.P.Prabhavathy. 2021. **A review on machine learning techniques for text classification.** Department of Computer Science and Engineering. SRM Institute of Science and Technology.

Velay, M and Daniel, F. 2018. **Using NLP on news headlines to predict index trends.** [Online]. Available https://arxiv.org/abs/1806.09533

เอกสารอ้างอิง (ต่อ)

Yildirim, S, Jothimani, D, Kavakioglu, C. and Basar, A. 2018. **Classification of "Hot News" for Financial Forecast Using NLP Techniques.** In 2018 IEEE International Conference on Big Data (Big Data). 4719-4722

ภาคผนวก

ภาคผนวก ก

การจำแนกและวิเคราะห์หมวดหมู่การใช้งานสำหรับลูกค้าของแพลตฟอร์ม text to speech โดยใช้ภาษาไพธอน (Python) มีคำสั่งดังต่อไปนี้

1. การจัดเตรียมข้อมูล

1.1 นำเข้าข้อมูล

```
# dev version
!pip install https://github.com/PyThaiNLP/pythainlp/archive/dev.zip

!pip install pythainlp
!pip install epitran
!pip install sklearn_crfsuite
!pip install tensorflow deepcut
!pip install attacut
!pip install emoji

!pip install gdown
import pandas as pd
from sklearn.feature_extraction.text import CountVectorizer
import numpy as np
import os
import gdown
import pickle
from google.colab import drive
drive.mount('/content/drive')
```



```
# โหลดข้อมูล
```

```
df = pd.read_pickle('/content/drive/MyDrive/BV/Mes.pickle')
df
```

1.2 การทำความสะอาดข้อความ

```
import tweepy
import pandas as pd
from sklearn.feature_extraction.text import CountVectorizer
import numpy as np
import emoji
from pythainlp.tokenize import word_tokenize
from pythainlp.corpus import thai_stopwords
import re
from wordcloud import WordCloud
import matplotlib.pyplot as plt


def cleanText(text):
    text = str(text)
    text = re.sub('[^ก-๙]','',text)
    stop_word = list(thai_stopwords())
    sentence = word_tokenize(text)
    result = [word for word in sentence if word not in stop_word and " " not in word]
    return text
cleaning = []
for txt in df["Mes"]:
    cleaning.append(cleanText(txt))
cleaning[:10]
```

```
df['cleaning'] = cleaning
df
```

1.3 การตัดคำ

```
def cleanText(text):
    text = str(text)
    text = re.sub('[^ก-๙]','',text)
    stop_word = list(thai_stopwords())
    sentence = word_tokenize(text, engine="newmm")
    result = [word for word in sentence if word not in stop_word and " " not in word]
    return ",".join(result)

def tokenize(d):
    result = d.split(",")
    result = list(filter(None, result))
    return result

Newmm = []
for txt in df['cleaning']:
    Newmm.append(cleanText(txt))

vectorizer = CountVectorizer(tokenizer=tokenize)
transformed_data = vectorizer.fit_transform(Newmm)
count_data = zip(vectorizer.get_feature_names_out(),
np.ravel(transformed_data.sum(axis=0)))  # ใช้ get_feature_names_out
keyword_df = pd.DataFrame(columns=['word', 'count'])
keyword_df['word'] = vectorizer.get_feature_names_out()
```

```
keyword_df['count'] = np.ravel(transformed_data.sum(axis=0))
keyword_df.sort_values(by=['count'], ascending=False).head(10)


df['Newmm'] = Newmm
df
```

1.4 กำหนดผลหมวดหมู่ของข้อความ

```
import json


with open('Dictionary.json', encoding='utf-8') as files :
    dictcatsubcat =json.load(files)
dictcatsubcat


#ไว้หาlabel
def getcat(text,lookup= dictcatsubcat ):
  for cat in lookup :
      for keyword in lookup[cat]:
        if keyword in text:
          return {"Category":cat}
  return {"Category":"กลุ่มคำ"}


df["Category"]= df['Newmm'].apply(lambda text:getcat(text)["Category"])
```

2. การสกัดคุณลักษณะ (Feature Extraction)

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from pythainlp.tokenize import word_tokenize
```

```
#Feature Extraction

# แบ่งข้อมูลเป็นชุดฝึกและชุดทดสอบ

X_train, X_test, y_train, y_test = train_test_split(df["Newmm"], df["Category"],
test_size=0.2, random_state=42)


# สร้าง TfidfVectorizer เพื่อแปลงข้อความเป็นเวกเตอร์ TF-IDF

tfidf_vectorizer = TfidfVectorizer(tokenizer=word_tokenize, analyzer='word',
max_features=5000)  # ปรับ max_features ตามต้องการ

X_train_tfidf = tfidf_vectorizer.fit_transform(X_train)

X_test_tfidf = tfidf_vectorizer.transform(X_test)
```

3. การสร้างแบบจำลอง

```
import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns


from sklearn.preprocessing import StandardScaler

from sklearn.discriminant_analysis import LinearDiscriminantAnalysis as LDA

from sklearn.model_selection import train_test_split, GridSearchCV, cross_val_score,
StratifiedKFold

from sklearn.metrics import confusion_matrix, accuracy_score


from sklearn.linear_model import LogisticRegression

from sklearn.svm import SVC

from sklearn.ensemble import RandomForestClassifier

from pythainlp.tokenize import word_tokenize
```

```
from sklearn.feature_extraction.text import TfidfVectorizer

from sklearn.naive_bayes import MultinomialNB
```

3.1 การสร้างแบบจำลอง Random Forest

```
# สร้างและฝึกโมเดล Random Forest

random_forest = RandomForestClassifier(n_estimators=100, max_depth=120, random_state=42)  # ปรับพารามิเตอร์ตามต้องการ

random_forest.fit(X_train_tfidf, y_train)


from sklearn.metrics import f1_score, classification_report, confusion_matrix, ConfusionMatrixDisplay


def evaluation(name, predictions, actuals):
    print(name)
    print(classification_report(actuals, predictions))

    # คำนวณค่า F1-score และแสดงผล
    f1 = f1_score(actuals, predictions, average='micro')
    print("F1-score:", f1)

    # สร้างและแสดง Confusion Matrix
    cm = confusion_matrix(actuals, predictions)
    disp = ConfusionMatrixDisplay(confusion_matrix=cm)
    disp.plot()
    plt.show()

    return f1
```

```
# ประเมินโมเดลในชุดข้อมูลทดสอบ
y_pred_rd = random_forest.predict(X_test_tfidf)

# ใช้ฟังก์ชัน evaluation เพื่อประเมินผลลัพธ์ของโมเดล Logistic Regression
evaluation("Random Forest", y_pred_rd, y_test)
```

3.2 การสร้างแบบจำลอง Logistic Regression

```
# สร้างและฝึกโมเดล Logistic Regression
logistic_regression = LogisticRegression()
logistic_regression.fit(X_train_tfidf, y_train)

from sklearn.metrics import f1_score, classification_report, confusion_matrix,
ConfusionMatrixDisplay

def evaluation(name, predictions, actuals):
    print(name)
    print(classification_report(actuals, predictions))

    # คำนวณค่า F1-score และแสดงผล
    f1 = f1_score(actuals, predictions, average='micro')
    print("F1-score:", f1)

    # สร้างและแสดง Confusion Matrix
    cm = confusion_matrix(actuals, predictions)
    disp = ConfusionMatrixDisplay(confusion_matrix=cm)
    disp.plot()
    plt.show()
```

```
    return f1


# ประเมินโมเดลในชุดข้อมูลทดสอบ
y_pred_test_lr = logistic_regression.predict(X_test_tfidf)


# ใช้ฟังก์ชัน evaluation เพื่อประเมินผลลัพธ์ของโมเดล Logistic Regression
evaluation("Logistic Regression", y_pred_test_lr, y_test)
```

3.3 การสร้างแบบจำลอง SVM

```
# สร้างและฝึกโมเดล SVM
svm_classifier = SVC(kernel='linear')  # เลือก kernel เป็น 'linear' เนื่องจากสำหรับข้อมูลขนาดใหญ่และข้อความมักให้ผลลัพธ์ที่ดี
svm_classifier.fit(X_train_tfidf, y_train)


from sklearn.metrics import f1_score, classification_report, confusion_matrix, ConfusionMatrixDisplay


def evaluation(name, predictions, actuals):
    print(name)
    print(classification_report(actuals, predictions))


    # คำนวณค่า F1-score และแสดงผล
    f1 = f1_score(actuals, predictions, average='micro')
    print("F1-score:", f1)


    # สร้างและแสดง Confusion Matrix
    cm = confusion_matrix(actuals, predictions)
```

```
    disp = ConfusionMatrixDisplay(confusion_matrix=cm)
    disp.plot()
    plt.show()

    return f1

# ประเมินโมเดล
y_pred_svc = svm_classifier.predict(X_test_tfidf)

# ใช้ฟังก์ชัน evaluation เพื่อประเมินผลลัพธ์ของโมเดล Naive Bayes
evaluation("SVC", y_pred_svc, y_test)
```

3.4 การสร้างแบบจำลอง Naive Bayes

```
# สร้างและฝึกโมเดล
# ใช้ Multinomial Naive Bayes ในการจำแนกประเภทข้อความ
nb_classifier = MultinomialNB()
nb_classifier.fit(X_train_tfidf, y_train)

from sklearn.metrics import f1_score, classification_report, confusion_matrix,
ConfusionMatrixDisplay

def evaluation(name, predictions, actuals):
    print(name)
    print(classification_report(actuals, predictions))
```



```
    # คำนวณค่า F1-score และแสดงผล
```

```
    f1 = f1_score(actuals, predictions, average='micro')
    print("F1-score:", f1)

    # สร้างและแสดง Confusion Matrix
    cm = confusion_matrix(actuals, predictions)
    disp = ConfusionMatrixDisplay(confusion_matrix=cm)
    disp.plot()
    plt.show()

    return f1

# ใช้โมเดล Naive Bayes เพื่อทำนายผลลัพธ์
y_pred_nb = nb_classifier.predict(X_test_tfidf)

# ใช้ฟังก์ชัน evaluation เพื่อประเมินผลลัพธ์ของโมเดล Naive Bayes
evaluation("Naive Bayes", y_pred_nb, y_test)
```

3.5 ทดสอบประสิทธิภาพของแบบจำลองด้วยข้อมูลชุดเรียนรู้

3.5.1 แบบจำลอง Random Forest

```
# ทำนายผลลัพธ์ข้อมูล train
y_pred_rd_train = random_forest.predict(X_train_tfidf)

# ใช้ฟังก์ชัน evaluation เพื่อประเมินผลลัพธ์ของโมเดล Random Forest ด้วยข้อมูล train
evaluation("Random Forest", y_pred_rd_train, y_train)
```

3.5.2 แบบจำลอง Logistic Regression


```
# ทำนายผลลัพธ์ข้อมูล train
```

```
y_pred_lr_train = logistic_regression.predict(X_train_tfidf)

# ใช้ฟังก์ชัน evaluation เพื่อประเมินผลลัพธ์ของโมเดล Logistic Regression ด้วยข้อมูล train
evaluation("Logistic Regression", y_pred_lr_train, y_train)
```

3.5.3 แบบจำลอง SVM

```
# ทำนายผลลัพธ์ข้อมูล train
y_pred_svc_train = svm_classifier.predict(X_train_tfidf)

# ใช้ฟังก์ชัน evaluation เพื่อประเมินผลลัพธ์ของโมเดล SVM ด้วยข้อมูล train
evaluation("SVC", y_pred_svc_train,y_train  )
```

3.5.4 แบบจำลอง Naive Bayes

```
# ทำนายผลลัพธ์ข้อมูล train
y_pred_nb_train = nb_classifier.predict(X_train_tfidf)

# ใช้ฟังก์ชัน evaluation เพื่อประเมินผลลัพธ์ของโมเดล Naive Bayes ด้วยข้อมูล train
evaluation("Naive Bayes", y_pred_nb_train, y_train)
```

งานทะเบียนคณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

คำรับรองเล่มสหกิจศึกษา

วันที่ 8 เดือน พฤษภาคม พ.ศ 2567

ข้าพเจ้า นายวิชยุตม์ แท่นจิรวัฒนา รหัสนักศึกษา 63050663

นักศึกษาหลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชา สถิติประยุกต์ ภาควิชา สถิติ

ขอรับรองว่าสหกิจศึกษา เรื่อง

การจำแนกและวิเคราะห์หมวดหมู่การใช้งาน

สำหรับลูกค้าของแพลตฟอร์ม text to speech

ANALYSIS AND CLASSIFICATION OF CUSTOMER

TEXT FROM TEXT-TO-SPEECH PLATFORM

ปีการศึกษา 2566

เป็นผลงานวิจัยที่มิได้คัดลอกหรือละเมิดลิขสิทธิ์ของผู้อื่นและได้ผ่านการตรวจสอบความซ้ำซ้อน เรียบร้อยแล้ว และได้แนบเอกสารการตรวจสอบการลอกเลียนงานวรรณกรรมที่ตรวจสอบจากเล่มสหกิจศึกษาฉบับสมบูรณ์แล้ว

โปรแกรมอักขราวิสุทธิ์ 2.12%

ลงชื่อ.....วิชยุตม์ แท่นจิรวัฒนา.....

(นายวิชยุตม์ แท่นจิรวัฒนา)

นักศึกษา

ข้าพเจ้า ผศ.ดร.ยุวดี กล่อมวิเศษ อาจารย์ที่ปรึกษาสหกิจศึกษา ได้ตรวจสอบสหกิจศึกษาของนักศึกษาข้างต้นแล้ว ขอรับรองว่าเป็นผลงานวิจัยของนักศึกษาจริงและมีเนื้อหาสมบูรณ์ จึงลงชื่อไว้เป็นหลักฐาน

ลงชื่อ.....

(ผศ.ดร.ยุวดี กล่อมวิเศษ)

อาจารย์ที่ปรึกษา