การวิเคราะห์ความรู้สึกต่อโครงการบ้านเพื่อคนไทยจากความคิดเห็นบนยูทูป

SENTIMENT ANALYSIS OF BANPHEUKONTHAI PROJECT BASED ON YOUTUBE COMMENTS

เขมรัตน ทับแว่ว

KHAMARAT TUBWAEW

การค้นคว้าอิสระนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการข้อมูลและการวิเคราะห์
ศูนย์วิเคราะห์ข้อมูลดิจิทัลอัจฉริยะพระจอมเกล้าเจ้าลาดกระบัง คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ. 2568

KMITL-2025-SC-M-017-055

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SENTIMENT ANALYSIS OF BANPHEUKONTHAI PROJECT BASED ON YOUTUBE COMMENTS

KHAMARAT TUBWAEW

AN INDEPENDENT STUDY SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIRENMENT FOR THE DEGREE OF MASTER OF SCIENCE
IN DATA SCIENCE AND ANALYTICS
KMITL DIGITAL ANALTICS AND INTELLIGENCE CENTER SCHOOL OF SCIENCE
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
2025

KMITL-2025-SC-M-017-055

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

COPYRIGHT 2025

SCHOOL OF SCIENCE

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อการค้นคว้าอิสระ	การวิเคราะห์ความรู้สึกต่อโครงการบ้านเพื่อคนไทย จากความคิดเห็นบนยูทูป
ชื่อนักศึกษา	นางสาวเขมรัตน ทับแว่ว
รหัสประจำตัว	63605065
ปริญญา	วิทยาศาสตรมหาบัณฑิต (วิทยาการข้อมูลและการวิเคราะห์) ศูนย์วิเคราะห์ข้อมูลดิจิทัลอัจฉริยะพระจอมเกล้าลาดกระบัง
พ.ศ.	2568
อาจารย์ที่ปรึกษาการค้นคว้าอิสระ	ผู้ช่วยศาสตราจารย์ ดร.กนกกรรณ์ ลี้โรจนาประภา

บทคัดย่อ

งานวิจัยฉบับนี้มีวัตถุประสงค์เพื่อศึกษาความคิดเห็นของประชาชนที่มีต่อโครงการ “บ้านเพื่อคนไทย” ซึ่งเป็นนโยบายด้านที่อยู่อาศัยของภาครัฐ โดยใช้ความคิดเห็นจากเว็บไซต์ยูทูปเป็นแหล่งข้อมูลหลัก ผ่านกระบวนการทำเหมืองข้อความและการวิเคราะห์ข้อความโดยใช้เทคนิคการประมวลผลภาษาธรรมชาติ (Natural Language Processing: NLP) โดยจำแนกความคิดเห็นออกเป็น 3 ประเภท ได้แก่ เชิงบวก เชิงกลาง และเชิงลบ การทดลองประกอบด้วยการเปรียบเทียบประสิทธิภาพของเครื่องมือตัดคำภาษาไทยระหว่าง PyThaiNLP และ DeepCut และการประเมินแบบจำลองการเรียนรู้จำนวน 13 แบบ ทั้งในกลุ่ม Traditional Machine Learning และ Deep Learning โดยใช้ค่าความถูกต้อง ค่าความแม่นยำ ค่าความระลึก และค่าความถ่วงดุล เป็นเกณฑ์วัดผล ผลการวิจัยพบว่าแบบจำลองซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machine) โดยใช้ตัวตัดคำ PyThaiNLP ให้ผลลัพธ์การจำแนกที่ดีที่สุด โดยมีค่าความถูกต้องร้อยละ 60.06 ค่าความแม่นยำร้อยละ 60.40 ค่าความระลึกร้อยละ 60.06 และค่าความถ่วงดุลร้อยละ 59.94

คำสำคัญ : การตัดคำ การทำเหมืองข้อความ การวิเคราะห์ความรู้สึก โครงการบ้านเพื่อคนไทย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Independent Study Title	Sentiment Analysis of Banpheukonthai Project Based on YouTube Comments
Student Name	Khamarat Tubwaew
Student ID	63605065
Degree	Master of Science (Data Science and Analytics) KMITL Digital Analytics and Intelligence Center
Year	2025
Independent study Advisor	Assist. Prof. Dr. Kanogkan Leerojanaprapa

Abstract

This paper investigates public sentiment toward the “Banpheukonthai Project” project, a governmental housing policy, by analyzing user comments on YouTube. The study applies text mining and Natural Language Processing (NLP) techniques to categorize comments into three sentiment classes: positive, neutral, and negative. The methodology involves a comparative evaluation of two Thai word segmentation tools, PyThaiNLP and DeepCut, as well as 13 classification models from both traditional Machine Learning and Deep Learning approaches. Performance is measured using accuracy, precision, recall, and F1-score. Experimental results show that the Support Vector Machine (SVM) model, combined with the PyThaiNLP tokenizer, achieved the best classification performance with an accuracy of 60.06%, precision of 60.40%, recall of 60.06%, and an F1-score of 59.94%.

Keywords : Tokenization, Text Mining, Sentiment Analysis, Banpheukonthai Project

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

การวิจัยฉบับนี้เป็นส่วนหนึ่งของการศึกษาหลักสูตร ปริญญาวิทยาศาสตร์มหาบัณฑิต สาขาวิชาวิทยาการข้อมูลและการวิเคราะห์ ศูนย์วิเคราะห์ข้อมูลดิจิทัลอัจฉริยะพระจอมเกล้าเจ้าลาดกระบัง คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังข้าพเจ้าขอกราบขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร.กนกกรรณ์ ลิ้โรจนาประภา อาจารย์ที่ปรึกษาหลัก ซึ่งได้กรุณาให้คำแนะนำ ตรวจสอบ ตลอดจนให้กำลังใจ และเสนอแนะแนวทางในการทำวิจัยอย่างต่อเนื่องด้วยความอดทน เอาใจใส่ และเมตตา ส่งผลให้งานวิจัยฉบับนี้สามารถดำเนินไปได้อย่างราบรื่นและสำเร็จลุล่วงตามเป้าหมายที่วางไว้

ข้าพเจ้าขอแสดงความขอบพระคุณอย่างสูงต่อ คณะกรรมการสอบวิทยานิพนธ์ ทุกท่าน ที่ได้กรุณาให้คำแนะนำที่มีคุณค่า และช่วยชี้แนะแนวทางที่เหมาะสม เพื่อพัฒนางานวิจัยให้มีความสมบูรณ์มากยิ่งขึ้น

ขอขอบพระคุณ อาจารย์ผู้สอนทุกท่าน ที่ได้ถ่ายทอดความรู้และประสบการณ์อันทรงคุณค่าตลอดระยะเวลาการศึกษาในระดับบัณฑิตศึกษา

นอกจากนี้ ข้าพเจ้าขอขอบคุณ เพื่อนร่วมรุ่นและผู้ให้ข้อมูลทุกท่าน ที่ให้ความร่วมมือและให้กำลังใจในการดำเนินการเก็บรวบรวมข้อมูล รวมถึงช่วยเหลือในด้านต่าง ๆ ด้วยมิตรไมตรีและความจริงใจ

ท้ายที่สุด ข้าพเจ้าขอกราบขอบพระคุณ บิดา มารดา และครอบครัว ที่เป็นแรงสนับสนุนสำคัญทั้งทางด้านจิตใจและทรัพยากรในทุกช่วงเวลา และเป็นกำลังใจสำคัญที่ผลักดันให้ข้าพเจ้าก้าวผ่านอุปสรรคต่าง ๆ มาจนถึงวันนี้

ข้าพเจ้าขอน้อมรับความเมตตาและความช่วยเหลือจากทุกท่านด้วยความซาบซึ้งใจอย่างยิ่ง

นางสาวเขมรัตน ทับแว่ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ	ก
Abstract	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง
สารบัญตาราง	ฌ
สารบัญรูป	ฎ
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์ของงานวิจัย	2
1.3 ขอบเขตของงานวิจัย	3
1.4 ประโยชน์ที่คาดว่าจะได้รับ	3
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง	4
2.1 การทำเหมืองข้อมูล (Data Mining)	4
2.2 การเตรียมข้อมูล (Data Preparation)	4
2.2.1 การติดฉลากข้อมูล (Data Labeling)	5
2.2.2 การนอร์มอลไลซ์ตัวอักษร (Character Normalization)	5
2.2.3 การลบเครื่องหมายวรรคตอนและสัญลักษณ์ (Punctuation Removal)	5
2.2.4 การตัดคำ (Tokenization)	6
2.2.5 การกำจัดคำหยุด (Stop Word Removal)	7
2.2.6 การแปลงคำเป็นเวกเตอร์ตัวเลข (Word Vectorize)	7
2.3 การวิเคราะห์ความรู้สึก (Sentiment analysis)	8
2.3.1 การวิเคราะห์การถดถอยโลจิสติก (Logistic Regression)	8
2.3.2 นาอีฟเบย์ (Naïve Bayes)	9
2.3.3 การหาค่าด้วยเพื่อนบ้านใกล้ที่สุด k จุด (K-Nearest Neighbors – KNN)	10
2.3.4 ซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machine – SVM)	10
2.3.5 ป่าสุ่ม (Random Forest)	11
2.3.6 ต้นไม้ตัดสินใจ (Decision Tree)	12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้า

2.3.7 ตัวจำแนกแบบโครงข่ายประสาทเทียมหลายชั้น (Multi-Layer Perceptron) 12
2.3.8 เพอร์เซปตรอน (Perceptron) 13
2.3.9 สโตแคสติกการ์เดียนดิเซนท์ (Stochastic Gradient Descent) 13
2.3.10 พาสซีฟ อากัสซีฟ (Passive Aggressive) 14
2.3.11 โครงข่ายประสาทหน่วยความจำระยะสั้นแบบยาว (LSTM) 15
2.3.12 โครงข่ายประสาทหน่วยความจำระยะสั้นแบบยาวแบบสองทิศทาง (Bidirectional LSTM) 15
2.3.13 โครงข่ายประสาทเทียมแบบคอนโวลูชัน (CNN) 16
2.4 การสร้างแท็กคลาวด์ (Tag Cloud) 17
2.5 การประเมินผล (Evaluation) 18
2.5.1 เมทริกซ์ความสับสน (Confusion Matrix) 18
2.5.2 ค่าความถูกต้อง (Accuracy) 19
2.5.3 ค่าความแม่นยำ (Precision) 19
2.5.4 ค่าความระลึก (Recall) 19
2.5.5 ค่าความถ่วงดุล (F1-score) 19
2.6 งานวิจัยที่เกี่ยวข้อง 19
บทที่ 3 วิธีการดำเนินงานวิจัย 22
3.1 การศึกษาทฤษฎีและงานวิจัย 23
3.2 การรวบรวมข้อมูล (Data Collection) 23
3.3 การเตรียมข้อมูล (Data Preparation) 25
3.3.1 การติดฉลากข้อมูล (Data Labeling) 26
3.3.2 การนอร์มอลไลซ์ตัวอักษร (Character Normalization) 28
3.3.3 การลบเครื่องหมายวรรคตอนและสัญลักษณ์ (Punctuation Removal) 29
3.3.4 การตัดคำ (Tokenization) 30
3.3.5 การกำจัดคำหยุด (Stop Word Removal) 30
3.3.6 การแปลงคำเป็นเวกเตอร์ตัวเลข (Word Vectorize) 31
3.4 การวิเคราะห์ความรู้สึกของคำในแต่ละกลุ่ม 32
3.5 การจำแนกความรู้สึกจากความคิดเห็น 32
3.5.1 การถดถอยเชิงเส้นโลจิสติค (Logistic Regression) 33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้า

3.5.2 นาอีฟเบย์ (Naïve Bayes) 33
3.5.3 การหาค่าด้วยเพื่อนบ้านใกล้ที่สุด k จุด (K-Nearest Neighbors) 33
3.5.4 ซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machine) 34
3.5.5 ป่าสุ่ม (Random Forest) 34
3.5.6 ต้นไม้ตัดสินใจ (Decision Tree) 34
3.5.7 โครงข่ายประสาทเทียมหลายชั้น (Multi-Layer Perceptron) 34
3.5.8 เพอร์เซปตรอน (Perceptron) 35
3.5.9 สโตแคสติกการ์เดียนดิเซนท์ (Stochastic Gradient Descent) 35
3.5.10 พาสซีฟ อากัสซีฟ (Passive Aggressive) 35
3.5.11 โครงข่ายประสาทหน่วยความจำระยะสั้นแบบยาว (LSTM) 36
3.5.12 โครงข่ายประสาทหน่วยความจำระยะสั้นแบบยาวแบบสองทิศทาง (Bidirectional LSTM) 36
3.5.13 โครงข่ายประสาทเทียมแบบคอนโวลูชัน (CNN) 37
3.6 การปรับค่าพารามิเตอร์ 37
3.7 การวัดประสิทธิภาพ 42
บทที่ 4 ผลการวิจัยและอภิปรายผล 44
4.1 ข้อมูลที่ใช้ในการทดลอง 44
4.2 การเปรียบเทียบการวิเคราะห์คำในแต่ละประเภทความรู้สึกของแต่ละวิธีการตัดคำ 44
4.2.1 ผลวิเคราะห์คำในแต่ละประเภทความรู้สึกของการใช้ตัวตัดคำ PyThaiNLP 45
4.2.2 ผลวิเคราะห์คำในแต่ละประเภทความรู้สึกของการใช้ตัวตัดคำ DeepCut 47
4.3 ผลการวิเคราะห์คำในแต่ละประเภทความรู้สึก 50
4.3.1 ผลวิเคราะห์คำในแต่ละประเภทความรู้สึกเชิงบวก 51
4.3.2 ผลวิเคราะห์คำในแต่ละประเภทความรู้สึกเชิงกลาง 51
4.3.3 ผลวิเคราะห์คำในแต่ละประเภทความรู้สึกเชิงลบ 51
4.4 การเปรียบเทียบผลการจำแนกความคิดเห็นโครงการบ้านเพื่อคนไทยตามตัวตัดคำ 52
4.5 ผลการจำแนกความคิดเห็นต่อโครงการบ้านเพื่อคนไทยจากแบบจำลอง 54
4.5.1 ผลจากแบบจำลองการถดถอยเชิงโลจิสติค (Logistic Regression) 55

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้า

4.5.2 ผลจากแบบจำลองนาอีฟเบย์ (Naïve Bayes) 56
4.5.3 ผลจากแบบจำลองการหาค่าด้วยเพื่อนบ้านใกล้ที่สุด k จุด (K-Nearest Neighbors) 58
4.5.4 ผลจากแบบจำลองซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machine) 60
4.5.5 ผลจากแบบจำลองป่าสุ่ม (Random Forest) 62
4.5.6 ผลจากแบบจำลองต้นไม้ตัดสินใจ (Decision Tree) 64
4.5.7 ผลจากแบบจำลองโครงข่ายประสาทเทียมหลายชั้น (Multi-Layer Perceptron) 66
4.5.8 ผลจากแบบจำลองเพอร์เซปตรอน (Perceptron) 68
4.5.9 ผลจากแบบจำลองสโตแคสติกการ์เดียนดิเซนท์ (Stochastic Gradient Descent) 70
4.5.10 ผลจากแบบจำลองพาสซีฟ อากัสซีฟ (Passive Aggressive) 72
4.5.11 ผลจากแบบจำลองโครงข่ายประสาทหน่วยความจำระยะสั้นแบบยาว (LSMT) 74
4.5.12 ผลจากแบบจำลองโครงข่ายประสาทหน่วยความจำระยะสั้นแบบยาวแบบสองทิศทาง (Bidirectional LSTM) 76
4.5.13 ผลจากแบบจำลองโครงข่ายประสาทเทียมแบบคอนโวลูชัน (CNN) 78
4.6 ผลการเปรียบเทียบประสิทธิภาพของแบบจำลอง 80
4.7 ผลการนำแบบจำลองไปใช้งาน 82
4.7.1 การตรวจสอบข้อความความคิดเห็นจากผู้ใช้งาน 82
4.8 อภิปรายผล 84
บทที่ 5 สรุปผลการดำเนินงานและข้อเสนอแนะ 85
5.1 สรุปผลการวิจัย 85
5.1.1 การวิเคราะห์คำในแต่ละประเภทความรู้สึก 86
5.1.2 การเปรียบเทียบผลการจำแนกความคิดเห็นโครงการบ้านเพื่อคนไทยตามตัวตัดคำ 87
5.1.3 การเปรียบเทียบแบบจำลองการจำแนกความคิดเห็นต่อโครงการบ้านเพื่อคนไทย 87

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้า

5.2 ข้อเสนอแนะ 88
เอกสารอ้างอิง 89
ประวัติผู้เขียน 95

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

		หน้า
ตารางที่ 3.1	ตารางอธิบายชื่อคุณลักษณะของความคิดเห็นที่ได้จากการดึง API	24
ตารางที่ 3.2	ตารางอธิบายกำหนดเกณฑ์ของความคิดเห็น	26
ตารางที่ 3.3	ตัวอย่างในการนอร์มอลไลซ์ตัวอักษรในภาษาไทย	28
ตารางที่ 3.4	ตัวอย่างขั้นตอนการนอร์มอลไลซ์อักษร	29
ตารางที่ 3.5	ตัวอย่างขั้นตอนการลบเครื่องหมายวรรคตอนและสัญลักษณ์ (Punctuation Removal)	29
ตารางที่ 3.6	ตัวอย่างขั้นตอนการตัดคำ (Tokenization)	30
ตารางที่ 3.7	ตัวอย่างขั้นตอนการกำจัดคำหยุด (Stop Word Removal)	31
ตารางที่ 3.8	ตัวอย่างขั้นตอนการแปลงคำเป็นเวกเตอร์ตัวเลข (Word Vectorize)	31
ตารางที่ 3.9	ช่วงค่าพารามิเตอร์ที่ตั้งค่าสำหรับเพิ่มประสิทธิภาพของแบบจำลอง	37
ตารางที่ 3.10	ค่าพารามิเตอร์ที่ดีที่สุดที่ใช้เพิ่มประสิทธิภาพแบบจำลองที่ใช้ตัวตัดคำ PyThaiNLP	40
ตารางที่ 3.11	ค่าพารามิเตอร์ที่ดีที่สุดที่ใช้เพิ่มประสิทธิภาพแบบจำลองที่ใช้ตัวตัดคำ DeepCut	41
ตารางที่ 4.1	ผลการประเมินประสิทธิภาพแบบจำลองจากการตัวตัดคำ PyThaiNLP	52
ตารางที่ 4.2	ผลการประเมินประสิทธิภาพแบบจำลองจากการตัวตัดคำ DeepCut	53
ตารางที่ 4.3	เมทริกซ์ความสับสนแบบจำลองการถดถอยเชิงโลจิสติคด้วยข้อมูลทดสอบ โดยใช้ตัวตัดคำ PyThaiNLP	55
ตารางที่ 4.4	เมทริกซ์ความสับสนแบบจำลองการถดถอยเชิงโลจิสติคด้วยข้อมูลทดสอบ โดยใช้ตัวตัดคำ DeepCut	56
ตารางที่ 4.5	เมทริกซ์ความสับสนแบบจำลองนาอีฟเบย์ด้วยข้อมูลทดสอบ โดยใช้ตัวตัดคำ PyThaiNLP	57
ตารางที่ 4.6	เมทริกซ์ความสับสนแบบจำลองนาอีฟเบย์ด้วยข้อมูลทดสอบ โดยใช้ตัวตัดคำ DeepCut	58
ตารางที่ 4.7	เมทริกซ์ความสับสนแบบจำลองการหาค่าด้วยเพื่อนบ้านใกล้ที่สุด k จุด ด้วยข้อมูลทดสอบ โดยใช้ตัวตัดคำ PyThaiNLP	59
ตารางที่ 4.8	เมทริกซ์ความสับสนแบบจำลองการหาค่าด้วยเพื่อนบ้านใกล้ที่สุด k จุด ด้วยข้อมูลทดสอบ โดยใช้ตัวตัดคำ DeepCut	60
ตารางที่ 4.9	ผลการทดสอบแบบจำลองซัพพอร์ตเวกเตอร์แมชชีนด้วยข้อมูลทดสอบ โดยใช้ตัวตัดคำ PyThaiNLP	61

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้า

ตารางที่ 4.10 ผลการทดสอบแบบจำลองซัพพอร์ตเวกเตอร์แมชชีนด้วยข้อมูลทดสอบ โดยใช้ตัวตัดคำ DeepCut 62
ตารางที่ 4.11 เมทริกซ์ความสับสนแบบจำลองป่าสุ่มด้วยข้อมูลทดสอบ โดยใช้ตัวตัดคำ PyThaiNLP 63
ตารางที่ 4.12 เมทริกซ์ความสับสนแบบจำลองป่าสุ่มด้วยข้อมูลทดสอบ โดยใช้ตัวตัดคำ DeepCut 64
ตารางที่ 4.13 เมทริกซ์ความสับสนแบบจำลองต้นไม้ตัดสินใจด้วยข้อมูลทดสอบ โดยใช้ตัวตัดคำ PyThaiNLP 65
ตารางที่ 4.14 เมทริกซ์ความสับสนแบบจำลองต้นไม้ตัดสินใจด้วยข้อมูลทดสอบ โดยใช้ตัวตัดคำ DeepCut 65
ตารางที่ 4.15 เมทริกซ์ความสับสนแบบจำลองโครงข่ายประสาทเทียมหลายชั้นด้วยข้อมูลทดสอบ โดยใช้ตัวตัดคำ PyThaiNLP 67
ตารางที่ 4.16 เมทริกซ์ความสับสนแบบจำลองโครงข่ายประสาทเทียมหลายชั้นด้วยข้อมูลทดสอบ โดยใช้ตัวตัดคำ DeepCut 68
ตารางที่ 4.17 เมทริกซ์ความสับสนแบบจำลองเพอร์เซปตรอนด้วยข้อมูลทดสอบ โดยใช้ตัวตัดคำ PyThaiNLP 69
ตารางที่ 4.18 เมทริกซ์ความสับสนแบบจำลองเพอร์เซปตรอนด้วยข้อมูลทดสอบ โดยใช้ตัวตัดคำ DeepCut 70
ตารางที่ 4.19 เมทริกซ์ความสับสนแบบจำลองสโตแคสติกการ์เดียนดิเซนท์ด้วยข้อมูลทดสอบ โดยใช้ตัวตัดคำ PyThaiNLP 71
ตารางที่ 4.20 เมทริกซ์ความสับสนแบบจำลองสโตแคสติกการ์เดียนดิเซนท์ด้วยข้อมูลทดสอบ โดยใช้ตัวตัดคำ DeepCut 72
ตารางที่ 4.21 เมทริกซ์ความสับสนแบบจำลองพาสซีฟ อากัสซีฟด้วยข้อมูลทดสอบ โดยใช้ตัวตัดคำ PyThaiNLP 73
ตารางที่ 4.22 เมทริกซ์ความสับสนแบบจำลองพาสซีฟ อากัสซีฟด้วยข้อมูลทดสอบ โดยใช้ตัวตัดคำ DeepCut 74
ตารางที่ 4.23 เมทริกซ์ความสับสนแบบจำลองโครงข่ายประสาทหน่วยความจำระยะสั้นแบบยาวด้วยข้อมูลทดสอบ โดยใช้ตัวตัดคำ PyThaiNLP 75

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้า

ตารางที่ 4.24 เมทริกซ์ความสับสนแบบจำลองโครงข่ายประสาทหน่วยความจำระยะสั้นแบบยาวด้วยข้อมูลทดสอบ โดยใช้ตัวตัดคำ DeepCut 76
ตารางที่ 4.25 เมทริกซ์ความสับสนแบบจำลองโครงข่ายประสาทหน่วยความจำระยะสั้นแบบยาวแบบสองทิศทางด้วยข้อมูลทดสอบ โดยใช้ตัวตัดคำ PyThaiNLP 77
ตารางที่ 4.26 เมทริกซ์ความสับสนแบบจำลองโครงข่ายประสาทหน่วยความจำระยะสั้นแบบยาวแบบสองทิศทางด้วยข้อมูลทดสอบ โดยใช้ตัวตัดคำ DeepCut 78
ตารางที่ 4.27 เมทริกซ์ความสับสนแบบจำลองโครงข่ายประสาทเทียมแบบคอนโวลูชันด้วยข้อมูลทดสอบ โดยใช้ตัวตัดคำ PyThaiNLP 79
ตารางที่ 4.28 ผลการทดสอบแบบจำลองโครงข่ายประสาทเทียมแบบคอนโวลูชันด้วยข้อมูลทดสอบ โดยใช้ตัวตัดคำ DeepCut 80
ตารางที่ 4.29 ค่าประสิทธิภาพของแบบจำลองจากการใช้ตัวตัดคำ PyThaiNLP 80
ตารางที่ 4.30 ค่าประสิทธิภาพของแบบจำลองจากการใช้ตัวตัดคำ DeepCut 81

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

หน้า

รูปที่ 2.1 ตัวอย่างโค้ด Python ด้วยไลบรารี Word Cloud 17
รูปที่ 2.2 ตัวอย่างการภาพ จาก Word Cloud 17
รูปที่ 2.3 เมทริกซ์ความสับสนที่ใช้มีขนาด 3×3 18
รูปที่ 3.1 Workflows ของภาพรวมการทำงานทั้งหมด 22
รูปที่ 3.2 Google Colaboratory หรือ Google Colab 23
รูปที่ 3.3 อัลกอริทึมการเก็บรวบรวมข้อมูลโดยการดึงข้อมูลจากเว็บไซต์ยูทูป 24
รูปที่ 3.4 ตัวอย่างข้อมูลความคิดเห็นที่ดึงจากเว็บไซต์ยูทูป 25
รูปที่ 3.5 ขั้นตอนการเตรียมข้อมูล 25
รูปที่ 3.6 ตัวอย่างข้อมูลที่ทำการติดฉลากชุดข้อมูลเชิงบวก 27
รูปที่ 3.7 ตัวอย่างข้อมูลที่ทำการติดฉลากชุดข้อมูลเชิงกลาง 27
รูปที่ 3.8 ตัวอย่างข้อมูลที่ทำการติดฉลากชุดข้อมูลเชิงลบ 28
รูปที่ 3.9 ตัวอย่างคำที่ไม่มีความสำคัญภาษาไทย 30
รูปที่ 4.1 จำนวนข้อความในแต่ละประเภทความรู้สึก 44
รูปที่ 4.2 Word Cloud ของข้อความความคิดเห็นเชิงบวก โดยใช้ตัวตัดคำแบบ PyThaiNLP 45
รูปที่ 4.3 Word Cloud ของข้อความความคิดเห็นเชิงกลาง โดยใช้ตัวตัดคำแบบ PyThaiNLP 46
รูปที่ 4.4 Word Cloud ของข้อความความคิดเห็นเชิงลบ โดยใช้ตัวตัดคำแบบ PyThaiNLP 47
รูปที่ 4.5 Word Cloud ของข้อความความคิดเห็นเชิงบวก โดยใช้ตัวตัดคำแบบ DeepCut 48
รูปที่ 4.6 Word Cloud ของข้อความความคิดเห็นเชิงกลาง โดยใช้ตัวตัดคำแบบ DeepCut 49
รูปที่ 4.7 Word Cloud ของข้อความความคิดเห็นเชิงลบ โดยใช้ตัวตัดคำแบบ DeepCut 50
รูปที่ 4.8 การตรวจสอบข้อความเชิงบวกจากผู้ใช้งาน 82
รูปที่ 4.9 การตรวจสอบข้อความเชิงกลางจากผู้ใช้งาน 83
รูปที่ 4.10 การตรวจสอบข้อความเชิงลบจากผู้ใช้งาน 83

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

งานวิจัยนี้มีที่มาที่แสดงถึง ความสำคัญของการกำหนดปัญหาที่ใช้ในการศึกษา และการกำหนดวัตถุประสงค์ในการวิจัย ดังนี้

1.1 ความเป็นมาและความสำคัญของปัญหา

ในยุคปัจจุบันที่เทคโนโลยีสารสนเทศและสื่อสังคมออนไลน์มีบทบาทอย่างยิ่งต่อการสื่อสารและแสดงความคิดเห็นของประชาชน การรับฟังเสียงของสาธารณะกลายเป็นเครื่องมือสำคัญในการประเมินผลกระทบของนโยบายหรือโครงการต่าง ๆ ของภาครัฐ โดยเฉพาะความคิดเห็นในโลกออนไลน์ซึ่งมีความรวดเร็ว เข้าถึงง่าย และสะท้อนมุมมองของผู้คนในวงกว้างได้อย่างแท้จริง (Karamat and Farooq, 2016)

หนึ่งในโครงการสำคัญของรัฐบาลไทยคือ "โครงการบ้านเพื่อคนไทย" ที่พัฒนาโดย บริษัท เอสอาร์ที แอสเสท จำกัด ซึ่งเป็นบริษัทย่อยของการรถไฟแห่งประเทศไทย (รฟท.) โดยใช้ที่ดินรอบสถานีรถไฟเป็นฐานพัฒนา ในรูปแบบที่อยู่อาศัยให้เช่าระยะยาวในราคาที่เอื้อมถึง การดำเนินโครงการระยะแรกเป็นโครงการนำร่องในปี พ.ศ. 2568 – 2569 ครอบคลุม 4 พื้นที่ ได้แก่ สถานีกลางบางซื่อ จังหวัดเชียงใหม่ เชียงราก จังหวัดปทุมธานี และธนบุรี จังหวัดกรุงเทพฯ จุดเด่นอยู่ที่การพัฒนาแบบ Transit-Oriented Development (TOD) ช่วยลดภาระค่าเดินทางและเชื่อมต่อระบบขนส่งสาธารณะอย่างมีประสิทธิภาพ รวมทั้งสนับสนุนเป้าหมายการลดการปล่อยก๊าซคาร์บอน และส่งเสริมคุณภาพชีวิตของผู้อยู่อาศัย นอกจากนี้ โครงการได้ออกแบบที่อยู่อาศัยให้มีความหลากหลายทั้งคอนโดมิเนียมขนาด 30–51 ตารางเมตร และบ้านเดี่ยวชั้นเดียวขนาด 50 ตารางวา เพื่อรองรับกลุ่มผู้มีรายได้ไม่เกิน 50,000 บาทต่อเดือน และไม่เคยมีกรรมสิทธิ์ในที่อยู่อาศัยมาก่อน (บ้านเพื่อคนไทย, 2567)

อย่างไรก็ตาม แม้โครงการดังกล่าวจะมีเจตนารมณ์ที่ดี แต่การตอบรับจากสาธารณะผ่านช่องทางออนไลน์ โดยเฉพาะบนเว็บไซต์ยูทูป ซึ่งเป็นแพลตฟอร์มที่มีผู้ใช้งานจำนวนมากในประเทศไทยและเปิดให้ผู้ใช้งานสามารถแสดงความคิดเห็นได้อย่างเสรี กลับแสดงให้เห็นถึงความคิดเห็นที่หลากหลาย ทั้งในเชิงบวก เชิงกลาง และเชิงลบ ซึ่งสะท้อนมุมมอง ความพึงพอใจ หรือข้อกังวลของประชาชนในหลากหลายแง่มุม การเลือกใช้เว็บไซต์ยูทูปเป็นแหล่งข้อมูลหลักในงานวิจัยนี้ เกิดจากลักษณะของข้อมูลที่เป็น User-Generated Content ซึ่งสามารถเข้าถึงความคิดเห็นในเชิงธรรมชาติได้อย่างแท้จริง อีกทั้งยังมีความหลากหลายทั้งในแง่ของภาษา รูปแบบ และเนื้อหา (Thelwall. et al., 2012)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การวิเคราะห์ความคิดเห็นของประชาชนจากข้อมูลบนเว็บไซต์ยูทูป โดยใช้เทคนิคการวิเคราะห์ความรู้สึก (Sentiment Analysis) ซึ่งเป็นส่วนหนึ่งของเทคโนโลยีการประมวลผลภาษาธรรมชาติ (Natural Language Processing: NLP) จึงเป็นเครื่องมือที่มีศักยภาพในการทำความเข้าใจมุมมองของประชาชนได้อย่างเป็นระบบและแม่นยำ (Liu, 2012) โดยเฉพาะเมื่อใช้ร่วมกับเทคนิคการทำเหมืองข้อความ (Text Mining) และการจำแนกประเภทความคิดเห็นผ่านแบบจำลองการเรียนรู้ของเครื่อง (Machine Learning Models) (Cambria. et al., 2013)

งานวิจัยนี้จึงมุ่งเน้นการวิเคราะห์ความคิดเห็นของประชาชนที่แสดงความคิดเห็นต่อโครงการบ้านเพื่อคนไทยบนเว็บไซต์ยูทูป มีวัตถุประสงค์เพื่อจำแนกความคิดเห็นตามระดับความรู้สึก ได้แก่ เชิงบวก เชิงกลาง และเชิงลบ พร้อมนำเสนอผลการวิเคราะห์ในรูปแบบ Word Cloud เพื่อถ่ายทอดสาระสำคัญของข้อความในแต่ละประเภท นอกจากนี้ยังเปรียบเทียบประสิทธิภาพของตัวตัดคำภาษาไทยระหว่าง PyThaiNLP และ DeepCut ซึ่งเป็นขั้นตอนสำคัญก่อนการประมวลผล ตลอดจนเปรียบเทียบประสิทธิภาพของแบบจำลองการจำแนกความคิดเห็นจำนวน 13 แบบ ทั้งในกลุ่มโมเดลแบบดั้งเดิม (Traditional Machine Learning) และโมเดลเชิงลึก (Deep Learning) เพื่อระบุแนวทางที่เหมาะสมที่สุดสำหรับการวิเคราะห์ความคิดเห็นในบริบทภาษาไทย (Kowsari. et al., 2019)

1.2 วัตถุประสงค์ของงานวิจัย

1) เปรียบเทียบตัวตัดคำภาษาไทยที่เหมาะสมกับวิธีการจำแนกความคิดเห็นต่อโครงการบ้านเพื่อคนไทยระหว่าง PyThaiNLP กับ DeepCut
2) เปรียบเทียบวิธีการจำแนกความคิดเห็นต่อโครงการบ้านเพื่อคนไทยด้วยแบบจำลอง 13 แบบ ได้แก่ การถดถอยเชิงเส้นโลจิสติค (Logistic Regression) นาอีฟเบย์ (Naïve Bayes) การหาค่าด้วยเพื่อนบ้านใกล้ที่สุด k จุด (K-Nearest Neighbors) ซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machine) ป่าสุ่ม (Random Forest) ต้นไม้ตัดสินใจ (Decision Tree) ตัวจำแนกแบบโครงข่ายประสาทเทียมหลายชั้น (Multi-Layer Perceptron) เพอร์เซปตรอน (Perceptron) สโตแคสติกการ์เดียนดิเซนท์ (Stochastic Gradient Descent) พาสซีฟ อากัสซีฟ (Passive Aggressive) โครงข่ายประสาทหน่วยความจำระยะสั้นแบบยาว (Long Short-Term Memory) โครงข่ายประสาทหน่วยความจำระยะสั้นแบบยาวแบบสองทิศทาง (Bidirectional LSTM) โครงข่ายประสาทเทียมแบบคอนโวลูชัน (Convolutional Neural Network)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ขอบเขตของงานวิจัย

1) วิเคราะห์ความรู้สึกของความคิดเห็นต่อโครงการบ้านเพื่อคนไทย โดยใช้ความคิดเห็นจากเว็บไซต์ยูทูป ในช่วงวันที่ 1 ธันวาคม 2567 ถึงวันที่ 30 เมษายน 2568
2) ใช้ภาษาไพธอนในการทำเหมืองข้อมูล การวิเคราะห์ความรู้สึก ตรวจสอบและเปรียบเทียบประสิทธิภาพของแบบจำลอง
3) การเปรียบเทียบแบบจำลองจะใช้ชุดข้อมูลเดียวกัน โดยวัดผลจากค่าความถูกต้อง ค่าความแม่นยำ ค่าความระลึกและค่าความถ่วงดุล
4) เครื่องมือตัดคำที่ใช้ในการศึกษา ได้แก่ PyThaiNLP และ DeepCut เพื่อเปรียบเทียบประสิทธิภาพต่อการประมวลผลของแบบจำลอง

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1) ได้ข้อมูลเชิงลึกเกี่ยวกับความคิดเห็นและความรู้สึกของประชาชนต่อโครงการบ้านเพื่อคนไทย ซึ่งสามารถนำไปใช้เป็นแนวทางในการพัฒนาโครงการในอนาคต
2) ช่วยให้หน่วยงานรัฐหรือผู้พัฒนาโครงการที่มีลักษณะเดียวกันในอนาคต สามารถรับรู้ถึงปัญหา ข้อเสนอแนะ หรือความต้องการของประชาชนจากข้อมูลจริงที่เกิดขึ้นในสื่อสังคมออนไลน์
3) สามารถนำมาเป็นกรณีศึกษาที่สามารถประยุกต์ใช้เทคนิคการประมวลผลภาษาธรรมชาติ (Natural Language Processing) และการขุดเหมืองข้อความ (Text Mining)
4) สนับสนุนการใช้เทคโนโลยีวิเคราะห์ข้อมูลในเชิงกลยุทธ์ เพื่อสร้างความเข้าใจระหว่างภาครัฐและประชาชน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

เนื้อหาในบทนี้นำเสนอรายละเอียดของทฤษฎีและงานวิจัยที่เกี่ยวข้อง ดังต่อไปนี้

2.1 การทำเหมืองข้อมูล (Data Mining)

การทำเหมืองข้อมูล (Data Mining) หมายถึงกระบวนการสกัดความรู้ที่เป็นประโยชน์จากชุดข้อมูลขนาดใหญ่ โดยใช้เทคนิคต่าง ๆ จากหลายศาสตร์ เช่น สถิติ การเรียนรู้ของเครื่อง (Machine Learning) การรู้จำรูปแบบ (Pattern Recognition) และฐานข้อมูล (Database Systems) เพื่อค้นหารูปแบบ แนวโน้ม หรือความสัมพันธ์ที่ซ่อนอยู่ในข้อมูล (Han. et al., 2011)

กระบวนการทำเหมืองข้อมูลประกอบด้วยหลายขั้นตอน เช่น การเลือกข้อมูล (Data Selection), การทำความสะอาดข้อมูล (Data Cleaning), การแปลงข้อมูล (Data Transformation), การทำเหมืองข้อมูล (Data Mining) และการแปลผล (Interpretation) ซึ่งมักรวมอยู่ในกระบวนการที่เรียกว่า Knowledge Discovery in Databases (KDD) (Fayyad. Et al., 1996)

ในปัจจุบัน การทำเหมืองข้อมูลถูกนำมาใช้ในหลายสาขา เช่น ธุรกิจ (เพื่อวิเคราะห์พฤติกรรมลูกค้า), การแพทย์ (เพื่อวินิจฉัยโรคจากข้อมูลผู้ป่วย), การเงิน (เพื่อวิเคราะห์ความเสี่ยงในการให้สินเชื่อ) รวมถึงด้านการประมวลผลภาษาธรรมชาติ (Natural Language Processing) และการวิเคราะห์ความคิดเห็น (Sentiment Analysis) ด้วย (Feldman and Sanger, 2007)

โดยเทคนิคที่นิยมในงานเหมืองข้อมูล ได้แก่ การจำแนกประเภท (Classification), การจัดกลุ่ม (Clustering), การถดถอย (Regression), การค้นหารูปแบบ (Pattern Discovery) และการตรวจจับความผิดปกติ (Anomaly Detection) (Witten. Et al., 2011)

2.2 การเตรียมข้อมูล (Data Preparation)

การเตรียมข้อมูล (Data Preparation) เป็นกระบวนการสำคัญในงานประมวลผลภาษาธรรมชาติและการเรียนรู้ของเครื่อง ซึ่งส่งผลต่อคุณภาพและความแม่นยำของโมเดลอย่างมีนัยสำคัญ โดยในงานวิจัยนี้ ข้อมูลจะถูกเตรียมผ่านหลายขั้นตอน ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1 การติดฉลากข้อมูล (Data Labeling)

การติดฉลากข้อมูลเป็นขั้นตอนพื้นฐานของการเรียนรู้แบบมีผู้สอน (Supervised Learning) ซึ่งเกี่ยวข้องกับการกำหนดป้ายกำกับให้กับข้อมูลดิบ เช่น การระบุว่า ข้อความมีความรู้สึกเป็นบวกหรือลบ เพื่อให้โมเดลสามารถเรียนรู้รูปแบบจากข้อมูลที่ติดฉลากแล้วได้อย่างถูกต้อง (Zaidan, 2022) การติดฉลากข้อมูลที่ถูกต้องและสม่ำเสมอมีความสำคัญอย่างยิ่งต่อประสิทธิภาพของแบบจำลอง เพราะข้อมูลฝึกสอนที่มีคุณภาพสูงจะส่งผลโดยตรงต่อความสามารถของโมเดลในการจำแนกและทำนายผลในข้อมูลใหม่ (Halevy. et al., 2009)

ในงานด้านการประมวลผลภาษาธรรมชาติ (NLP) เช่น การวิเคราะห์ความคิดเห็น (Sentiment Analysis) หรือการจำแนกข้อความประเภทต่าง ๆ (Text Classification) กระบวนการติดฉลากมักอาศัยผู้เชี่ยวชาญหรือผู้ใช้ทั่วไปผ่านระบบ crowdsourcing เพื่อจัดประเภทข้อความตามหมวดหมู่ที่กำหนดไว้ล่วงหน้า (Alonso. et al., 2008)

2.2.2 การนอร์มอลไลซ์ตัวอักษร (Character Normalization)

การนอร์มอลไลซ์ข้อความหมายถึงการแปลงข้อความให้อยู่ในรูปแบบมาตรฐาน เช่น การแปลงรูปแบบของตัวอักษรพิเศษ การแปลงอักขระต่างรูปแบบให้เป็นแบบเดียวกัน เช่น การลบช่องว่างซ้ำ การแปลงอักขระพิเศษ และการแปลงข้อความที่มีลักษณะเหมือนกันแต่เขียนต่างกันให้อยู่ในรูปแบบเดียว เพื่อเพิ่มประสิทธิภาพของการวิเคราะห์ (Jurafsky and Martin, 2023) การทำ Character Normalization ช่วยลด noise และทำให้โมเดล NLP สามารถเรียนรู้และจับความสัมพันธ์ของคำหรือวลีในข้อความได้แม่นยำมากขึ้น โดยเฉพาะอย่างยิ่งในภาษาไทยที่มีการใช้ตัวอักษรพิเศษหรือลักษณะการพิมพ์ซ้ำเพื่อแสดงอารมณ์หรือเสียง (Supnithi. et al., 2013)

2.2.3 การลบเครื่องหมายวรรคตอนและสัญลักษณ์ (Punctuation Removal)

การลบเครื่องหมายวรรคตอนและสัญลักษณ์ต่าง ๆ เช่น !, ?, # เป็นกระบวนการทำความสะอาดข้อความที่ช่วยลด noise ในข้อมูล ทำให้สามารถนำไปประมวลผลได้ง่ายขึ้น โดยเฉพาะอย่างยิ่งในงานวิเคราะห์ข้อความที่มุ่งเน้นไปที่ความหมายของคำ เช่น การจำแนกความคิดเห็นหรือการจัดหมวดหมู่ข้อความ การลบสัญลักษณ์เหล่านี้ช่วยให้โมเดลโฟกัสเฉพาะคำสำคัญที่มีเนื้อหาเชิงความหมาย และลดโอกาสเกิดความคลาดเคลื่อนจากการตีความอักขระที่ไม่เกี่ยวข้อง (Rajaraman and Ullman, 2020) อย่างไรก็ตาม ในบางกรณี เช่น งานด้านอารมณ์หรือบทสนทนา การเลือกคงไว้ซึ่งบางสัญลักษณ์ที่สื่ออารมณ์ เช่น “!” อาจช่วยให้การวิเคราะห์แม่นยำขึ้น จึงควรมีการพิจารณาตามบริบทของงานวิจัย (Pang and Lee, 2008)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.4 การตัดคำ (Tokenization)

การตัดคำเป็นการแบ่งข้อความออกเป็นหน่วยย่อย เช่น คำหรือวลี ซึ่งเป็นขั้นตอนพื้นฐานก่อนการประมวลผลข้อความ เช่น "ฉันรักภาษาไทย" ⟶ [“ฉัน”, “รัก”, “ภาษาไทย”] โดยมักใช้ร่วมกับโมเดลที่ต้องการข้อมูลเชิงโครงสร้าง เช่น การวิเคราะห์ความคิดเห็น การจัดประเภทข้อความ หรือการแยกชื่อหน่วย (Named Entity Recognition) (Jurafsky and Martin, 2023) สำหรับภาษาไทยซึ่งไม่มีการเว้นวรรคระหว่างคำเหมือนภาษาอังกฤษ การตัดคำจึงเป็นความท้าทายสำคัญและจำเป็นต้องใช้เครื่องมือเฉพาะทาง เช่น PyThaiNLP หรือ DeepCut (Sornlertlamvanich. et al., 1997)

2.2.4.1 PyThaiNLP

PyThaiNLP เป็นไลบรารีโอเพนซอร์สที่พัฒนาด้วยภาษา Python สำหรับการประมวลผลภาษาธรรมชาติ (Natural Language Processing: NLP) ของภาษาไทยโดยเฉพาะ ไลบรารีนี้รวบรวมเครื่องมือและฟังก์ชันที่จำเป็นสำหรับการจัดการข้อมูลข้อความภาษาไทย เช่น การตัดคำ (word tokenization), การแปลงคำเป็นเสียงอ่าน (transliteration), การแทนที่คำผิด (spell checking), การแยกวลี, การหาค่าความถี่ของคำ และการแปลงคำเป็นลำดับเสียง (Romanization) เป็นต้น (PyThaiNLP Developers, 2024)

หนึ่งในปัญหาหลักของการประมวลผลภาษาธรรมชาติในภาษาไทย คือ ลักษณะของภาษาไทยที่ไม่มีการเว้นวรรคระหว่างคำ ทำให้การตัดคำกลายเป็นขั้นตอนที่สำคัญและท้าทาย PyThaiNLP จึงพัฒนาและสนับสนุนตัวตัดคำหลายแบบ เช่น newmm, deepcut, ulmfit, และ attacut ซึ่งผู้ใช้สามารถเลือกได้ตามลักษณะของข้อมูลหรือความเหมาะสมของงานวิจัย (Sornlertlamvanich. et al., 2019)

PyThaiNLP ยังสามารถเชื่อมต่อกับโมเดลการเรียนรู้ของเครื่องและโมเดลการเรียนรู้เชิงลึกได้ง่าย ช่วยเพิ่มความสามารถในการวิเคราะห์ข้อความภาษาไทยในเชิงลึก โดยเฉพาะอย่างยิ่งในงานด้านการวิเคราะห์ความรู้สึก การจัดหมวดหมู่ข้อความ และการดึงข้อมูลเชิงความหมาย (Lertniphonphun, 2019)

2.2.4.2 DeepCut

DeepCut เป็นเครื่องมือสำหรับการตัดคำภาษาไทยที่พัฒนาขึ้นโดยใช้แนวคิดของโมเดลการเรียนรู้เชิงลึก (Deep Learning) โดยเฉพาะโครงข่ายประสาทเทียมแบบ LSTM (Long Short-Term Memory) เพื่อเรียนรู้โครงสร้างของคำในภาษาไทยจากข้อมูลตัวอย่างจำนวนมาก (Rakwong and Sornlertlamvanich, 2019)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DeepCut ได้รับการฝึกสอนจากคลังข้อมูลภาษาไทยขนาดใหญ่ เช่น ORCHID และ BEST ซึ่งเป็นชุดข้อมูลมาตรฐานที่ใช้ในการวิจัยด้าน NLP ของภาษาไทย (Sornlertlamvanich et al., 2000) โดยจุดเด่นของ DeepCut คือสามารถตัดคำได้แม่นยำกว่าวิธีแบบพจนานุกรม (dictionary-based) หรือ rule-based โดยเฉพาะในกรณีคำใหม่ คำเฉพาะ หรือคำซ้อนที่มักเกิดในบทสนทนาออนไลน์ (Boonkwan, 2016)

แม้ว่าข้อดีของ DeepCut คือความแม่นยำ แต่ก็มีข้อจำกัดในด้านความเร็วในการประมวลผลเมื่อเทียบกับตัวตัดคำที่เน้นความเร็ว เช่น newmm หรือ attacut อย่างไรก็ตาม การที่ DeepCut ถูกผนวกรวมไว้ใน PyThaiNLP ก็ช่วยให้สามารถเรียกใช้งานได้ง่ายเพียงไม่กี่คำสั่งใน Python ทำให้เหมาะสำหรับการประยุกต์ในงานที่ต้องการความแม่นยำสูงทั้งในบริบทงานวิจัยและภาคอุตสาหกรรม (PyThaiNLP Developers, 2024)

2.2.5 การกำจัดคำหยุด (Stop Word Removal)

คำหยุด คือคำที่ปรากฏบ่อยในภาษาแต่ไม่มีความหมายสำคัญ เช่น “ที่” “และ” “ใน” ซึ่งไม่ช่วยในการแยกแยะเนื้อหาหลักของข้อความ การลบคำเหล่านี้ออกจากข้อความเป็นหนึ่งในขั้นตอนพื้นฐานของการทำความสะอาดข้อมูล โดยช่วยลดจำนวนคำที่ต้องประมวลผลและเน้นคำที่มีความหมายเชิงเนื้อหามากกว่า (Manning. et al., 2008)

การกำจัดคำหยุดช่วยเพิ่มประสิทธิภาพในการวิเคราะห์ข้อความ เช่น การวิเคราะห์ความคิดเห็น (Sentiment Analysis) การจัดหมวดหมู่ข้อความ (Text Classification) และการสร้างเวกเตอร์แทนคำ (Text Vectorization) โดยเฉพาะในภาษาไทยที่มีคำเชื่อมจำนวนมาก การลบคำหยุดจะทำให้แบบจำลองสามารถเรียนรู้ลักษณะเด่นของข้อความได้ดีขึ้น (Maas. et al., 2011)

ในไลบรารีอย่าง PyThaiNLP ก็มีรายการคำหยุดของภาษาไทยเตรียมไว้ให้ใช้งานทันที ซึ่งสามารถปรับแต่งเพิ่มเติมได้ตามลักษณะเฉพาะของงาน (PyThaiNLP Developers, 2024)

2.2.6 การแปลงคำเป็นเวกเตอร์ตัวเลข (Word Vectorize)

การแปลงข้อความเป็นเวกเตอร์ตัวเลขเป็นขั้นตอนสำคัญในงานประมวลผลภาษาธรรมชาติ (NLP) ซึ่งมีหลายเทคนิคให้เลือกใช้งาน เช่น One-Hot Encoding, TF-IDF (Term Frequency-Inverse Document Frequency), และ Word Embedding อย่าง Word2Vec, GloVe, หรือ BERT โดยแต่ละวิธีมีข้อดีแตกต่างกันไปตามลักษณะของข้อมูลและโมเดลที่ใช้ (Mikolov. et al., 2013)

การใช้เวกเตอร์ตัวเลขมีวัตถุประสงค์เพื่อให้โมเดลสามารถเข้าใจความหมายของคำ ความสัมพันธ์ระหว่างคำ รวมถึงบริบทในประโยคหรือเอกสารได้ดีขึ้น ซึ่งเป็นสิ่งที่จำเป็นอย่างยิ่งในงานเช่น การจำแนกความคิดเห็น การตอบคำถาม และการสร้างข้อความอัตโนมัติ (Devlin. et al.,

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2019) โดยเฉพาะ Word Embedding อย่าง Word2Vec และ BERT ซึ่งสามารถจับความหมายเชิงบริบทได้ลึกและแม่นยำกว่าวิธีดั้งเดิม เช่น TF-IDF (Manning. et al., 2008)

2.3 การวิเคราะห์ความรู้สึก (Sentiment analysis)

Sentiment Analysis คือระบบปัญญาประดิษฐ์ (AI) ที่ใช้ในการวิเคราะห์ความรู้สึกของผู้ใช้งานจากข้อความที่ปรากฏในช่องทางต่าง ๆ เช่น รีวิวสินค้า คอมเมนต์ในโซเชียลมีเดีย อีเมล การแชทกับ Chatbot หรือข้อความเสียง เพื่อทำความเข้าใจว่าผู้ใช้งานมีความรู้สึกอย่างไรกับแบรนด์หรือผลิตภัณฑ์ ไม่ว่าจะเป็นความรู้สึกในเชิงบวก เชิงลบ หรือเป็นกลาง รวมไปถึงการวิเคราะห์ความรู้สึกในระดับลึก เช่น ความสุข ความโกรธ ความเศร้า หรือความรัก (Liu, 2015)

ระบบวิเคราะห์ความรู้สึกมักใช้เทคนิคจาก Natural Language Processing (NLP) และ Machine Learning เพื่อช่วยในการตีความอารมณ์และเจตนาจากข้อความ โดยระบบจะสามารถเรียนรู้จากชุดข้อมูลที่ติดฉลาก (labeled data) แล้วนำไปประยุกต์ใช้ในบริบทต่าง ๆ เช่น การจัดหมวดหมู่ความคิดเห็น การประเมินคุณภาพบริการ หรือการเฝ้าระวังภาพลักษณ์ขององค์กรในสื่อสังคม (Medhat. et al., 2014)

2.3.1 การวิเคราะห์การถดถอยโลจิสติก (Logistic Regression)

การถดถอยโลจิสติก (Logistic Regression) เป็นหนึ่งในเทคนิคพื้นฐานของการวิเคราะห์ข้อมูลแบบจำแนกประเภท (Classification) ที่อยู่ภายใต้แนวทางของการเรียนรู้แบบมีผู้สอน (Supervised Learning) โดยมีวัตถุประสงค์เพื่อทำนายความน่าจะเป็นที่ตัวแปรตามจะอยู่ในกลุ่มใดกลุ่มหนึ่ง โดยเฉพาะกรณีที่ตัวแปรตามเป็นข้อมูลประเภทไบนารี (Binary outcome) เช่น ใช่/ไม่ใช่, ผ่าน/ไม่ผ่าน หรือ ป่วย/ไม่ป่วย เป็นต้น (Hosmer. et al., 2013)

ฟังก์ชันหลักที่ใช้ใน Logistic Regression คือ Sigmoid Function ซึ่งจะเปลี่ยนค่าที่ได้จากสมการเชิงเส้น (Linear Combination) ให้กลายเป็นค่าความน่าจะเป็นในช่วงระหว่าง 0 ถึง 1 ตามสูตร:

$$P(y=1|x) = \frac{1}{1+e^{-(\beta_0+\beta_1 x_1+\beta_2 x_2+\cdots+\beta_n x_n)}} \quad (2.1)$$

โดยที่

$P(y=1|x)$ คือ ความน่าจะเป็นที่ $y=1$ ภายใต้ข้อมูล x

β_0 คือ ค่าคงที่ (intercept)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$\beta_1,\ \beta_2, \cdots, \beta_n$ คือ ค่าสัมประสิทธิ์ของตัวแปรอิสระ

$x_1,\ x_2, \cdots, x_n$ คือ ค่าคุณลักษณะ (features) ของข้อมูล

e คือ ค่าคงที่ทางคณิตศาสตร์ (Euler's number)

การตัดสินใจว่าข้อมูลควรถูกจำแนกให้อยู่ในกลุ่มใดจะขึ้นกับเกณฑ์ความน่าจะเป็น เช่น หาก $P \geq 0.5$ อาจถือว่าข้อมูลนั้นอยู่ในกลุ่มที่ต้องการ เช่น $y = 1$ (Bishop, 2006)

การวิเคราะห์การถดถอยโลจิสติก (Logistic Regression) เป็นเทคนิคพื้นฐานที่ได้รับความนิยมอย่างแพร่หลายในงานด้านการจำแนกประเภท (Classification) โดยเฉพาะในบริบทของ การวิเคราะห์ความคิดเห็น (Sentiment Analysis) ซึ่งมีวัตถุประสงค์ในการจำแนกข้อความออกเป็นกลุ่มต่าง ๆ เช่น บวก (Positive), ลบ (Negative) หรือกลาง (Neutral)

ในเชิงคณิตศาสตร์ Logistic Regression ใช้ฟังก์ชันซิกมอยด์ (Sigmoid Function) เพื่อแปลงค่าจากสมการเชิงเส้นให้เป็นค่าความน่าจะเป็นระหว่าง 0 ถึง 1 โดยโมเดลจะเรียนรู้จากชุดข้อมูลที่มีการติดฉลาก (Labeled Data) และกำหนดผลลัพธ์ตามเกณฑ์ของค่าความน่าจะเป็นที่ได้ เช่น หาก $P(y = 1|x) \geq 0.5$ อาจตีความว่าเป็นข้อความเชิงบวก

ในการประมวลผลข้อความสำหรับงาน Sentiment Analysis ข้อความจะถูกแปลงเป็นเวกเตอร์คุณลักษณะ (Feature Vectors) ผ่านกระบวนการ เช่น TF-IDF, Bag-of-Words หรือการฝังเวกเตอร์ (Word Embedding) ก่อนนำไปเข้าสู่โมเดล Logistic Regression เพื่อการฝึกและการทำนายผล (Kim, 2014)

2.3.2 นาอีฟเบย์ (Naïve Bayes)

นาอีฟเบย์ (Naïve Bayes) เป็นหนึ่งในเทคนิคการเรียนรู้แบบมีผู้สอน (Supervised Learning) ที่ใช้สำหรับงานจำแนกประเภท (Classification) โดยอิงจาก ทฤษฎีบทของเบย์ (Bayes' Theorem) ซึ่งใช้คำนวณความน่าจะเป็นแบบมีเงื่อนไข (Conditional Probability) ระหว่างเหตุการณ์ต่าง ๆ โดยตั้งอยู่บนสมมติฐานที่ว่า คุณลักษณะ (Features) ทุกตัวมีความเป็นอิสระต่อกัน (Conditional Independence) ภายใต้คลาสเดียวกัน (Mitchell, 1997)

สมการพื้นฐานของทฤษฎีบทของเบย์

$$P(C \mid X) = \frac{P(x \mid C) \cdot P(C)}{P(X)} \tag{2.2}$$

โดยที่

$P(C \mid X)$ คือ ความน่าจะเป็นที่ตัวอย่าง X จะอยู่ในคลาส C

$P(X \mid C)$ คือความน่าจะเป็นของคุณลักษณะ X เมื่อรู้ว่าข้อมูลอยู่ในคลาส C

$P(C)$ คือ ความน่าจะเป็นล่วงหน้าของคลาส C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$P(X)$ คือ ความน่าจะเป็นของข้อมูลทั้งหมด

เนื่องจาก $P(X)$ มีค่าเท่ากันทุกคลาส จึงสามารถตัดออกได้ในการเปรียบเทียบค่าระหว่างคลาสต่าง ๆ ส่งผลให้โมเดลสามารถจำแนกคลาสที่มีค่า $P(x \mid C) \cdot P(C)$ มากที่สุดว่าเป็นผลลัพธ์ที่ต้องการ (Manning. et al., 2008)

Naïve Bayes ได้รับความนิยมในงานวิเคราะห์ข้อความ เช่น Spam Filtering, Sentiment Analysis และ Text Classification เนื่องจากประสิทธิภาพที่ดีแม้ข้อมูลมี noise และความเร็วที่สูงเมื่อเทียบกับโมเดลอื่น (Sahami. et al., 1998)

2.3.3 การหาค่าด้วยเพื่อนบ้านใกล้ที่สุด k จุด (K-Nearest Neighbors – KNN)

K-Nearest Neighbors (KNN) เป็นหนึ่งในอัลกอริทึมที่นิยมใช้ในงานจำแนกประเภท (Classification) และการถดถอย (Regression) โดยเป็นวิธีแบบ ไม่ใช้พารามิเตอร์ (Non-parametric) และไม่มีการฝึกโมเดลล่วงหน้า (Lazy Learning) กล่าวคือ การตัดสินใจหรือคาดการณ์จะเกิดขึ้นก็ต่อเมื่อมีข้อมูลใหม่เข้ามาเท่านั้น (Cover and Hart, 1967)

หลักการของ KNN คือการหาข้อมูลในชุดฝึก (Training Data) ที่มีความใกล้เคียงกับข้อมูลใหม่มากที่สุด โดยอาศัยการวัดระยะห่าง เช่น Euclidean Distance, Manhattan Distance, หรือ Minkowski Distance เพื่อประเมินความใกล้เคียงของแต่ละจุดข้อมูล (Jain. et al., 2000) จากนั้นจะเลือกข้อมูลที่อยู่ใกล้ที่สุดจำนวน k จุด มาวิเคราะห์โดยใช้หลักการ “เสียงข้างมาก (Majority Voting)” ในกรณีที่เป็นงานจำแนกประเภท หรือใช้ค่าเฉลี่ยหากเป็นงานถดถอย

$$d(p,q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_n - q_n)^2} \quad (2.3)$$

โดยที่ p และ q คือเวกเตอร์ของข้อมูลสองตัวอย่างในพื้นที่ n มิติ

KNN มีข้อดีคือใช้งานง่าย ไม่ซับซ้อน และให้ผลลัพธ์ที่ดีในกรณีที่จำนวนข้อมูลไม่มากหรือมีการกระจายที่ชัดเจน แต่ก็มีข้อเสียในด้านเวลาในการคำนวณเมื่อข้อมูลมีจำนวนมาก หรือเมื่อจำนวนคุณลักษณะ (features) สูง (Dasarathy, 1991)

2.3.4 ซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machine – SVM)

ซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machine – SVM) เป็นเทคนิคการเรียนรู้แบบมีผู้สอน (Supervised Learning) ที่ได้รับความนิยมอย่างสูงสำหรับงานจำแนกประเภท (Classification) โดยเฉพาะในปัญหาที่มีข้อมูลแบบสองคลาส (Binary Classification) SVM มี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แนวคิดหลักคือการหาขอบเขตจำแนก (Hyperplane) ที่สามารถแยกข้อมูลสองกลุ่มออกจากกันได้ดีที่สุด โดยพยายามเพิ่มระยะห่าง (Margin) ระหว่างกลุ่มข้อมูลทั้งสองมากที่สุด (Vapnik, 2000)

ในกรณีที่ข้อมูลไม่สามารถแยกออกได้ในเชิงเส้น SVM สามารถประยุกต์ใช้ Kernel Trick เพื่อแปลงข้อมูลไปยังมิติที่สูงขึ้น (Higher-dimensional Space) เพื่อให้สามารถจำแนกได้ ตัวอย่าง Kernel ที่นิยมใช้ ได้แก่ Linear Kernel, Polynomial Kernel, Radial Basis Function (RBF) Kernel และ Sigmoid Kernel (Cortes and Vapnik, 1995)

$$f(x) = sign(w^T \emptyset(x) + b) \qquad (2.4)$$

โดยที่

w คือ เวกเตอร์ของพารามิเตอร์

$\emptyset(x)$ คือ ฟังก์ชันแปลงข้อมูล

b คือ ค่า bias

$sign\,(\,)$ คือ ฟังก์ชันที่ให้ผลลัพธ์เป็นคลาส

Support Vector Machine ถูกใช้อย่างแพร่หลายในงานด้านการประมวลผลภาษาธรรมชาติ (NLP), Image Recognition, Bioinformatics และโดยเฉพาะอย่างยิ่งในงาน Sentiment Analysis เนื่องจากความสามารถในการจัดการกับข้อมูลที่มีมิติสูงและการแยกคลาสที่มีความซับซ้อน (Joachims, 1998)

2.3.5 ป่าสุ่ม (Random Forest)

ป่าสุ่ม (Random Forest) เป็นอัลกอริทึมการเรียนรู้แบบมีผู้สอน (Supervised Learning) ภายใต้แนวคิดของ การเรียนรู้แบบรวมกลุ่ม (Ensemble Learning) โดยสร้างแบบจำลองจากกลุ่มของต้นไม้ตัดสินใจ (Decision Trees) จำนวนมาก แล้วรวมผลการทำนายของแต่ละต้นไม้เข้าด้วยกัน (ผ่านการลงคะแนนเสียงหรือการหาค่าเฉลี่ย) เพื่อเพิ่มความแม่นยำและลดปัญหา Overfitting ที่มักพบในต้นไม้ตัดสินใจเดี่ยว ๆ (Breiman, 2001)

แนวคิดหลักของ Random Forest ประกอบด้วยสองกระบวนการสำคัญ คือ การสุ่มตัวอย่างข้อมูล (Bootstrap Sampling) สำหรับแต่ละต้นไม้ จะสุ่มตัวอย่างข้อมูลจากชุดฝึกแบบมีการแทนที่ (with replacement) และการสุ่มคุณลักษณะ (Feature Subsampling) ในแต่ละจุดแตกแขนงของต้นไม้ จะสุ่มเลือกเฉพาะบางคุณลักษณะ (features) เพื่อใช้ในการตัดสินใจ (Ho, 1998)

ผลลัพธ์สุดท้ายของการจำแนกประเภท (classification) จะได้จากการโหวตเสียงข้างมากของต้นไม้ทั้งหมด ส่วนในกรณีของการถดถอย (regression) จะได้จากค่าเฉลี่ยของค่าที่ต้นไม้แต่ละต้นทำนาย (Liaw and Wiener, 2002)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Random Forest ได้รับความนิยมอย่างสูงในงานด้าน การจัดหมวดหมู่ข้อความ (Text Classification) และการวิเคราะห์ความคิดเห็น (Sentiment Analysis) เพราะมีความยืดหยุ่นสูง ใช้งานง่าย และให้ผลลัพธ์ที่แม่นยำแม้ในข้อมูลที่มี noise หรือ missing values

2.3.6 ต้นไม้ตัดสินใจ (Decision Tree)

ต้นไม้ตัดสินใจ (Decision Tree) เป็นอัลกอริธึมการเรียนรู้แบบมีผู้สอน (Supervised Learning) ที่ใช้สำหรับงานจำแนกประเภท (Classification) และการถดถอย (Regression) โดยมีโครงสร้างคล้ายต้นไม้ประกอบด้วย โหนดราก (Root Node), โหนดภายใน (Internal Nodes) และโหนดใบ (Leaf Nodes) ซึ่งแต่ละโหนดจะเป็นการตัดสินใจตามเงื่อนไขของคุณลักษณะ (Feature) หนึ่งในข้อมูล (Quinlan, 1986)

การตัดสินใจในแต่ละโหนดจะใช้ตัวชี้วัด (Splitting Criteria) เช่น Entropy และ Information Gain ใช้ใน ID3, Gini Index ใช้ใน CART และ Gain Ratio ใช้ใน C4.5 (Breiman. et al., 1984) เป้าหมายของอัลกอริธึมคือการเลือกคุณลักษณะที่สามารถแบ่งแยกข้อมูลได้ดีที่สุดในแต่ละขั้นตอน เพื่อให้โหนดใบมีข้อมูลที่ เป็นกลุ่มเดียวกัน (Pure) มากที่สุด ซึ่งจะช่วยให้โมเดลมีความสามารถในการคาดการณ์ที่แม่นยำ (Quinlan, 1996)

2.3.7 ตัวจำแนกแบบโครงข่ายประสาทเทียมหลายชั้น (Multi-Layer Perceptron)

โครงข่ายประสาทเทียมหลายชั้น (Multi-Layer Perceptron – MLP) เป็นรูปแบบหนึ่งของโครงข่ายประสาทเทียมแบบป้อนหน้า (Feedforward Neural Network) ที่ประกอบด้วยชั้นของโหนด (neurons) ตั้งแต่สองชั้นขึ้นไป ได้แก่ ชั้นอินพุต (Input Layer), ชั้นซ่อน (Hidden Layers) และ ชั้นเอาต์พุต (Output Layer) โดยโหนดในแต่ละชั้นจะเชื่อมต่อกับโหนดในชั้นถัดไปผ่านน้ำหนัก (weights) และใช้ฟังก์ชันกระตุ้น (Activation Function) เพื่อประมวลผลสัญญาณ (Rumelhart. et al., 1986)

โครงข่ายประสาทเทียมหลายชั้นใช้อัลกอริธึมการแพร่ย้อนกลับ (Backpropagation) ร่วมกับวิธี การเรียนรู้แบบลดค่าความคลาดเคลื่อน (Gradient Descent) เพื่อปรับปรุงน้ำหนักให้โมเดลสามารถเรียนรู้จากข้อมูลได้อย่างมีประสิทธิภาพ (Haykin, 2009) ฟังก์ชันกระตุ้นยอดนิยมใน MLP ได้แก่ Sigmoid, Hyperbolic Tangent (tanh) และ Rectified Linear Unit (ReLU) (LeCun. et al., 2015)

MLP ถือเป็นรากฐานสำคัญของโครงข่ายประสาทเทียมยุคใหม่ และสามารถประยุกต์ใช้ในหลากหลายสาขา เช่น การจำแนกภาพ เสียง ข้อความ และการวิเคราะห์ความคิดเห็น (Sentiment Analysis)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.8 เพอร์เซปตรอน (Perceptron)

เพอร์เซปตรอน (Perceptron) เป็นโครงสร้างพื้นฐานของโครงข่ายประสาทเทียม (Artificial Neural Networks) ที่พัฒนาขึ้นโดย Frank Rosenblatt ในปี ค.ศ. 1958 เพอร์เซปตรอนเป็นโมเดลการเรียนรู้แบบมีผู้สอน (Supervised Learning) ที่ใช้สำหรับการจำแนกข้อมูลเป็นสองกลุ่ม (Binary Classification) โดยทำงานผ่านการคำนวณเชิงเส้นระหว่างเวกเตอร์ของอินพุตกับเวกเตอร์ของน้ำหนัก และผ่านฟังก์ชันการกระตุ้น (Activation Function) เพื่อให้ได้ผลลัพธ์สุดท้าย (Haykin, 2009)

สมการพื้นฐานของเพอร์เซปตรอน

$$y = f(\omega \cdot x + b) \tag{2.5}$$

โดยที่

ω คือ เวกเตอร์น้ำหนัก

x คือ เวกเตอร์อินพุต

b คือ ค่าไบแอส (bias)

f คือ ฟังก์ชันการกระตุ้น เช่น Step Function หรือ Sign Function

เพอร์เซปตรอนจะทำการปรับค่าน้ำหนักในแต่ละรอบการเรียนรู้ (iteration) โดยอาศัยกฎการอัปเดตน้ำหนัก (Weight Update Rule) ดังนี้

$$\omega_{new} = w_{old} + \Delta\omega = \omega_{old} + \eta(y_{true} + y_{pred})x \tag{2.6}$$

โดยที่

η คือ อัตราการเรียนรู้ (learning rate)

y_{true} คือ ค่าผลลัพธ์จริง

y_{pred} คือ ค่าที่โมเดลทำนาย

เพอร์เซปตรอนสามารถจำแนกข้อมูลได้ดีในกรณีที่ข้อมูลสามารถแยกเส้นตรงได้ (Linearly Separable) แต่หากข้อมูลไม่สามารถแบ่งได้ด้วยเส้นตรง เช่น กรณี XOR เพอร์เซปตรอนแบบเดี่ยว (Single-Layer Perceptron) จะไม่สามารถให้ผลลัพธ์ที่ถูกต้องได้ (Hagan. et al., 1996)

2.3.9 สโตแคสติกการ์เดียนดิเซนท์ (Stochastic Gradient Descent)

สโตแคสติกการ์เดียนต์ดิเซนท์ (Stochastic Gradient Descent – SGD) เป็นหนึ่งในเทคนิคการปรับพารามิเตอร์ที่สำคัญในงานเรียนรู้ของเครื่อง (Machine Learning) และโดยเฉพาะอย่างยิ่งใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงข่ายประสาทเทียม สโตแคสติกการ์เดียนต์ดิเซนท์เป็นรูปแบบหนึ่งของ Gradient Descent ซึ่งใช้สำหรับลดค่าฟังก์ชันความสูญเสีย (Loss Function) เพื่อทำให้โมเดลมีความสามารถในการทำนายได้แม่นยำยิ่งขึ้น (Bottou, 2012) เป็นรูปแบบหนึ่งของ Gradient Descent ที่แตกต่างจาก Batch Gradient Descent ตรงที่แทนที่จะคำนวณค่า gradient จากข้อมูลทั้งหมดในแต่ละรอบ SGD จะเลือกข้อมูลเพียง 1 ตัวอย่าง (หรือ mini-batch ขนาดเล็ก) มาใช้ในการอัปเดตน้ำหนัก ทำให้สามารถอัปเดตได้รวดเร็ว และมีโอกาสหลีกเลี่ยงจุดต่ำเฉพาะที่อาจเกิดขึ้นในฟังก์ชัน loss ได้ (LeCun. et al., 1998)

สมการของสโตแคสติกการ์เดียนต์ดิเซนท์

$$\theta_{t+1} = \theta_t - \eta \cdot \nabla_\theta L(\theta_{;x_i,y_i}) \tag{2.7}$$

โดยที่

θ คือ เวกเตอร์ของพารามิเตอร์

η คือ อัตราการเรียนรู้ (Learning Rate)

∇_θ คือ อนุพันธ์ของฟังก์ชันความสูญเสียต่อพารามิเตอร์

x_i, y_i คือ ข้อมูลฝึกชุดย่อยที่สุ่มขึ้นมาในแต่ละรอบ [66]

SGD เป็นพื้นฐานของอัลกอริธึมการเรียนรู้ขั้นสูงอื่น ๆ เช่น Momentum, RMSProp, และ Adam Optimizer ซึ่งช่วยปรับการอัปเดตให้มีประสิทธิภาพมากขึ้นในงาน Deep Learning

2.3.10 พาสซีฟ อากัสซีฟ (Passive Aggressive)

พาสซีฟ อากเกรสซีฟ (Passive–Aggressive, PA) เป็นอัลกอริทึมสำหรับการเรียนรู้แบบออนไลน์ (Online Learning) ที่ถูกเสนอขึ้นโดย Crammer et al. ในปี 2006 (Crammer. et al., 2549) โดยออกแบบมาให้สามารถปรับปรุงแบบจำลองแบบทันที โดยไม่ต้องนำข้อมูลทั้งหมดกลับมาฝึกใหม่ในทุกครั้งที่มีข้อมูลใหม่เข้ามา จึงเหมาะกับระบบที่ข้อมูลเข้ามาอย่างต่อเนื่อง เช่น Social Media, ระบบแนะนำ, หรือระบบการจำแนกความคิดเห็นแบบเรียลไทม์ (Joachims, 2006)

แนวคิด “Passive Aggressive” มาจาก Passive หากโมเดลสามารถทำนายได้ถูกต้องบนข้อมูลใหม่ จะไม่มีการปรับค่าพารามิเตอร์ใด ๆ และ Aggressive หากทำนายผิดหรือค่าความมั่นใจต่ำกว่าที่กำหนด จะมีการอัปเดตพารามิเตอร์ทันทีเพื่อให้โมเดลทำนายถูกในรอบถัดไป

ฟังก์ชันอัปเดตพารามิเตอร์ของ PA สำหรับกรณีการจำแนกแบบไบนารี มีลักษณะดังนี้

$$\omega_{t+1} = \omega_t + \tau_t \cdot y_t \cdot x_t \tag{2.8}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดย

$$\tau_t = \frac{\max(0,1- y_t \cdot \omega_t \cdot X_t)}{\|X_t\|^2} \quad (2.9)$$

ω_t คือ น้ำหนักของโมเดลในรอบที่ t

y_t คือ ค่าจำแนกจริงของข้อมูล

x_t คือ เวกเตอร์ของข้อมูล

τ_t คือ อัตราการอัปเดตที่คำนวณตามค่าความผิดพลาด

Passive Aggressive ได้รับความนิยมในงานที่ต้องการการประมวลผลเร็ว และสามารถรับมือกับข้อมูลสตรีมได้อย่างมีประสิทธิภาพ เช่น การจำแนกอีเมลสแปม การแนะนำข่าว หรือวิเคราะห์ความคิดเห็น

2.3.11 โครงข่ายประสาทหน่วยความจำระยะสั้นแบบยาว (Long Short-Term Memory)

โครงข่ายประสาทหน่วยความจำระยะสั้นแบบยาว (LSTM) เป็นโครงข่ายประสาทเทียมชนิดหนึ่งในกลุ่มของ Recurrent Neural Networks (RNNs) ที่ออกแบบมาเพื่อจัดการกับข้อมูลตามลำดับ (Sequential Data) โดยเฉพาะในกรณีที่มี ความสัมพันธ์ระยะยาว (Long-term dependencies) เช่น ข้อความต่อเนื่อง เวลา หรือเสียง (Hochreiter and Schmidhuber, 1997)

ปัญหาหลักของ RNN ทั่วไป คือการสูญเสียข้อมูลย้อนหลังเมื่อข้อมูลยาวเกินไป (Vanishing Gradient Problem) LSTM ถูกออกแบบมาเพื่อแก้ปัญหาดังกล่าวโดยใช้โครงสร้างภายในที่เรียกว่า หน่วยความจำ (Memory Cell) และกลไกที่ควบคุมการไหลของข้อมูลผ่าน ประตู (Gates) 3 ประเภท ได้แก่ Forget Gate ตัดสินใจว่าข้อมูลใดควรถูกลบจากหน่วยความจำ Input Gate กำหนดว่าข้อมูลใหม่ใดจะถูกเพิ่มเข้ามา Output Gate ตัดสินใจว่าข้อมูลใดจะถูกส่งออกไปยังขั้นตอนถัดไป (Graves, 2008)

การทำงานของ LSTM ช่วยให้สามารถจดจำและใช้งานข้อมูลที่อยู่ห่างจากปัจจุบันในลำดับได้อย่างมีประสิทธิภาพ ซึ่งเป็นข้อได้เปรียบอย่างมากในงานด้าน การวิเคราะห์ความคิดเห็น (Sentiment Analysis) การแปลภาษา (Machine Translation) และ การสร้างข้อความ (Text Generation) (Goldberg, 2016)

2.3.12 โครงข่ายประสาทหน่วยความจำระยะสั้นแบบยาวแบบสองทิศทาง (Bidirectional LSTM)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงข่ายประสาทหน่วยความจำระยะสั้นแบบยาวแบบสองทิศทาง (Bidirectional Long Short-Term Memory – BiLSTM) เป็นการพัฒนาเพิ่มเติมจากโครงข่าย LSTM (Long Short-Term Memory) โดยเพิ่มการประมวลผลข้อมูลแบบ ย้อนกลับ (Backward) ควบคู่กับการประมวลผลแบบเดินหน้า (Forward) ทำให้โมเดลสามารถเรียนรู้ลำดับข้อมูลได้ทั้งจากอดีตและอนาคตในแต่ละจุดของลำดับ (Schuster and Paliwal, 1997)

โมเดล BiLSTM ประกอบด้วย LSTM สองชุดทำงานคู่ขนานกัน คือ LSTM Forward Layer ทำงานจากต้นลำดับไปยังท้ายลำดับ และ LSTM Backward Layer ทำงานจากท้ายลำดับไปยังต้นลำดับผลลัพธ์จากทั้งสองทิศทางจะถูกนำมารวมกันผ่านการ Concatenation หรือ Summation แล้วจึงส่งต่อให้กับชั้นถัดไป เช่น ชั้น Dense หรือ Softmax สำหรับการจำแนก (Graves and Schmidhuber, 2005) โดย BiLSTM ถูกนำไปใช้อย่างแพร่หลายในงานที่เกี่ยวข้องกับ Natural Language Processing (NLP) เช่น การวิเคราะห์ความคิดเห็น (Sentiment Analysis), การจดจำเอนทิตี (Named Entity Recognition), และการสรุปข้อความ (Text Summarization) (Liu et al., 2016)

2.3.13 โครงข่ายประสาทเทียมแบบคอนโวลูชัน (Convolutional Neural Network)

โครงข่ายประสาทเทียมแบบคอนโวลูชัน (Convolutional Neural Network) เป็นหนึ่งในสถาปัตยกรรมของโครงข่ายประสาทเทียมที่ได้รับความนิยมอย่างแพร่หลาย โดยเฉพาะในงานประมวลผลภาพ แต่ยังสามารถนำไปใช้กับข้อมูลประเภทอื่นที่มีลักษณะเป็นลำดับหรือตาราง เช่น ข้อความ (text) หรือ ข้อมูลเวลา (time-series) (LeCun. et al., 2015)

โครงสร้างของ CNN ประกอบด้วย

1) Convolutional Layer ใช้ ฟิลเตอร์ (Filters) หรือ เคอร์เนล (Kernels) เลื่อนผ่านข้อมูลอินพุตเพื่อสกัดคุณลักษณะ (features) โดยอัตโนมัติ
2) Activation Function เช่น ReLU เพื่อแปลงผลลัพธ์จากการคอนโวลูชันให้สามารถเรียนรู้ความสัมพันธ์ที่ไม่เป็นเชิงเส้น
3) Pooling Layer เช่น Max Pooling เพื่อย่อขนาดข้อมูล ลดความซับซ้อน และลดความเสี่ยงของ Overfitting
4) Fully Connected Layer เพื่อรวมข้อมูลที่ผ่านการแปลงจากเลเยอร์ก่อนหน้า แล้วนำไปใช้จำแนกประเภทในขั้นสุดท้าย (Krizhevsky. et al., 2555)

สำหรับการประมวลผล ข้อความ CNN สามารถช่วยดึงคุณลักษณะเชิงลำดับที่สำคัญในแต่ละช่วงของประโยค ทำให้เหมาะกับงาน Sentiment Analysis และ Text Classification เช่นเดียวกับโมเดลที่เน้นลำดับ เช่น RNN หรือ LSTM (Kim, 2014)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 การสร้างแท็กคลาวด์ (Tag Cloud)

การสร้างแท็กคลาวด์ (Tag Cloud) เป็นกระบวนการนำเสนอคำหรือวลีที่ปรากฏบ่อยในชุดข้อมูลข้อความให้อยู่ในรูปแบบภาพ โดยคำที่ปรากฏบ่อยจะมีขนาดใหญ่กว่าคำที่ปรากฏน้อย ช่วยให้สามารถมองเห็นแนวโน้ม หรือประเด็นสำคัญจากข้อมูลได้อย่างรวดเร็วและชัดเจน โดยเฉพาะในการวิเคราะห์ข้อความ เช่น การวิเคราะห์ความรู้สึก (Sentiment Analysis) หรือการวิเคราะห์เนื้อหาข้อความจากแบบสอบถามหรือโซเชียลมีเดีย (Viégas and Wattenberg, 2008)

หนึ่งในเครื่องมือยอดนิยมที่ใช้สร้างแท็กคลาวด์คือไลบรารี wordcloud ในภาษา Python ซึ่งสามารถนำเข้าจากไลบรารี wordcloud โดยรองรับการสร้างภาพจากข้อความที่ผ่านการเตรียมข้อมูลแล้ว เช่น การตัดคำ การลบคำหยุด และการนับความถี่ของแต่ละคำ ก่อนจะนำไปประมวลผลสร้างภาพ ตัวอย่างคำสั่ง ดังรูปที่ 2.1 และผลลัพธ์ที่ได้ ดังรูปที่ 2.2

```
[56] from wordcloud import WordCloud
     import matplotlib.pyplot as plt

     # ตัวอย่างข้อความที่เตรียมไว้แล้ว
     text = "การศึกษา ภาษาไทย ข้อมูล การเรียนรู้ การวิเคราะห์ ความรู้ การเรียนรู้ ภาษาไทย"

     # import font
     fp = '/content/THSarabunNew.ttf'

     # กำหนด regex ใช้ regular expression
     #เพื่อจับคำภาษาไทย(ก-๙)และตัวอักษรภาษาอังกฤษ(a-zA-Z) พร้อมกับอักขระ ' เช่น apostrophe เป็นคำ
     # ทำให้สามารถดึงคำที่เป็นข้อความภาษาไทยหรือภาษาอังกฤษที่เกี่ยวข้องออกจากข้อความ
     reg = r"[ก-๙a-zA-Z']+"

     # สร้างแท็กคลาวด์
     wordcloud = WordCloud(stopwords=thai_stopwords(), background_color = 'white', max_words=2000,
                           height = 2000, width=4000, font_path=fp, regexp=reg).generate(text)

     # แสดงภาพ
     plt.imshow(wordcloud)
     plt.axis("off")
     plt.show()
```

รูปที่ 2.1 ตัวอย่างโค้ด Python ด้วยไลบรารี Word Cloud

รูปที่ 2.2 ตัวอย่างการภาพ จาก Word Cloud

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

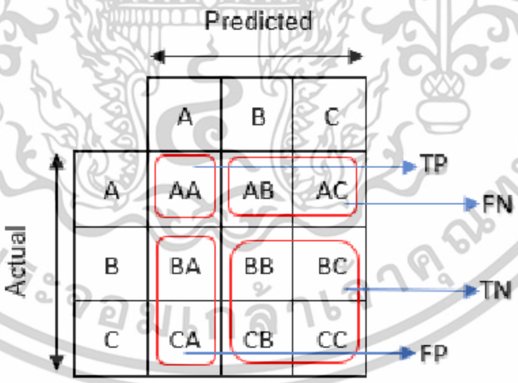
จากภาพที่ได้จะเห็นคำที่มีความถี่สูง เช่น "การเรียนรู้" หรือ "ภาษาไทย" ปรากฏด้วยขนาดที่ใหญ่กว่าคำอื่น ๆ ซึ่งสามารถนำไปใช้ในรายงานหรือ dashboard เพื่อสื่อความหมายได้อย่างรวดเร็วและมีประสิทธิภาพ (Mueller, 2015)

2.5 การประเมินผล (Evaluation)

หลังจากกระบวนการจำแนกประเภทเสร็จสิ้น ขั้นตอนถัดไปคือการประเมินผลลัพธ์ของการจำแนกประเภท โดยทั่วไปนิยมใช้ตัวชี้วัด เช่น ค่าความถูกต้อง (Accuracy) ค่าความแม่นยำ (Precision) ค่าความระลึก (Recall) และ ค่าความถ่วงดุล (F1-Score) ซึ่งเป็นดัชนีสำคัญในการประเมินคุณภาพของแบบจำลอง (Raschka and Mirjalili, 2019) โดยค่าดังกล่าวสามารถคำนวณได้จาก เมทริกซ์ความสับสน (Confusion Matrix) ซึ่งแสดงจำนวนของการพยากรณ์ที่ถูกต้องและผิดพลาดในแต่ละประเภท (Fawcett, 2006)

2.5.1 เมทริกซ์ความสับสน (Confusion Matrix)

เมทริกซ์ความสับสน คือ เมทริกซ์ที่แสดงผลการจำแนกจริงและผลการทำนายของโมเดล โดยมีขนาดเป็น L×L ซึ่ง L คือจำนวนป้ายกำกับ/คลาส ในการศึกษานี้ เมทริกซ์ความสับสนที่ใช้มีขนาด 3×3 เนื่องจากมีจำนวนคลาสทั้งหมด 3 คลาส ส่วนประกอบของเมทริกซ์ความสับสนแบบ 3×3 แสดงไว้ในรูปที่ 2.3 (Brownlee, 2020)



รูปที่ 2.3 เมทริกซ์ความสับสนที่ใช้มีขนาด 3×3

- **ค่าที่อยู่ในแนวทแยง** คือ ค่าที่ทำนายถูกต้อง (True Positive) ได้แก่ AA, BB, CC
- **ค่าที่อยู่นอกแนวทแยง** คือ ค่าที่ทำนายผิด (False Positive/False Negative) ได้แก่ AB, AC, BA, BC, CA และ CB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.2 ค่าความถูกต้อง (Accuracy)

ค่าความถูกต้องเป็นการวัดสัดส่วนของการทำนายที่ถูกต้องจากทั้งหมด

$$\text{ค่าความถูกต้อง (Accuracy)} = \frac{TP+TN}{TP+TN+FP+FN} = \frac{TP_{Total}}{Dataset_{Total}} \quad (2.16)$$

2.5.3 ค่าความแม่นยำ (Precision)

ความแม่นยำเป็นการวัดสัดส่วนของการทำนายที่ถูกต้องในแต่ละประเภทจากการทำนายทั้งหมดที่เป็นประเภทนั้น

$$\text{ค่าความแม่นยำ (Precision)} = \frac{TP}{TP+FP} = \frac{TP}{Prediction_{Total}} \quad (2.17)$$

2.5.4 ค่าความระลึก (Recall)

ค่าความระลึกเป็นการวัดสัดส่วนของการทำนายที่ถูกต้องในแต่ละประเภทขากจำนวนข้อมูลที่เป็นจริงในประเภท

$$\text{ค่าความระลึก (Recall)} = \frac{TP}{TP+FN} = \frac{TP}{Actual_{Total}} \quad (2.18)$$

2.5.5 ค่าความถ่วงดุล (F1-score)

ค่าความถ่วงดุลเป็นการวัดค่ากลางที่ถ่วงดุลระหว่าง Precision และ Recall

$$\text{ค่าความถ่วงดุล (F1-score)} = \frac{2 \times Precision \times Recall}{Precision+Recall} \quad (2.19)$$

2.6 งานวิจัยที่เกี่ยวข้อง

ในหัวข้อนี้จะกล่าวถึงงานวิจัยที่เกี่ยวข้องกับการวิเคราะห์ความรู้สึกต่อโครงการบ้านเพื่อคนไทยจากความคิดเห็นบนยูทูป และนำมาพยากรณ์โดยแบบจำลองต่าง ๆ

ณัฐิมา อภินาทเมธี (2567) การจำแนกการความรู้สึกจากหัวข้อข่าวหุ้นไทยรายวัน โดยทำการเปรียบเทียบการรูปแบบการเก็บ 3 แบบ (A - 0 และการเตรียมข้อความข่าวด้วยตัวตัดคำ ภาษาไทย 2 แบบ คือ PyThaiNLP และ DeepCut (I - II) ทำให้สามารถแบ่งการทดลองเป็น 6 การ ทดลองก่อนเลือกรูปแบบการเตรียมข้อความที่เหมาะสม เพื่อนำมาสร้างตัวแบบการจำแนกทั้ง 10 อัลกอลิทึม ในการศึกษาครั้งนี้เลือกหุ้น 2 ตัว จากตัวอย่างที่สุ่มจากหุ้นในกลุ่ม SET50 ซึ่งได้แก่ หุ้นของ บริษัท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ควอลิตี้เฮ้าส์ จำกัด (มหาชน) และบริษัท ราช กรุ๊ป จำกัด (มหาชน) เก็บรวบรวมข้อมูลทั้ง หัวข้อข่าว และราคาหุ้นรายวัน ตั้งแต่เดือนมกราคม พ.ศ. 2561 ถึงเดือนธันวาคม พ.ศ. 2565 นำหัวข้อ ข่าวที่รวบรวมมาเข้ากระบวนการเตรียมข้อมูลด้วยการลบเครื่องหมายวรรคตอน สัญลักษณ์ และ ตัวเลข การนอร์มอลไลซ์ตัวอักษร การตัดคำ และการสกัดคุณลักษณะ สำหรับการกำหนดแนวโน้ม ราคาหุ้นเป็นบวก เป็นลบ และเป็นกลางจากผลต่างของราคาปิดรายวันผลการวิจัยในขั้นตอนการเตรียม ข้อความพบว่า การเก็บข้อความเฉพาะหัวข้อข่าวของหุ้นที่ สนใจ (A) และเลือกการใช้ตัวตัดคำ PyThaiNLP (I) ให้ค่าประสิทธิภาพการจำแนกในโมเดลตาม พารามิเตอร์พื้นฐานดีที่สุดจึงเลือกการเตรียมข้อความที่ให้ผลดีที่สุดมาแบ่งเป็นชุดข้อมูลเรียนรู้ (ร้อยละ 70) เพื่อการปรับจูนพารามิเตอร์โดยใช้ GridSearchCV และชุดข้อมูลทดสอบ (ร้อยละ 30) เพื่อเปรียบเทียบประสิทธิภาพการจำแนก พบว่าชุดสำหรับหุ้น บริษัท ราช กรุ๊ป จำกัด (มหาชน) (RATCH) แบบจำลองซัพพอร์ตเวกเตอร์แมซชีนมีค่าประสิทธิภาพสูงสุด โดยมีค่าความถูกต้องร้อยละ 62.50 สำหรับหุ้น บริษัท ควอลิตี้เฮ้าส์ จำกัด (มหาชน) (QH) แบบจำลองป่าสุ่ม มีค่าประสิทธิภาพสูงสุดโดย มีค่าความถูกต้องร้อยละ 65

ภานุมาศ ทุกข์จาก (2565) การตัดสินใจที่จะลงทุนในสกุลเงินดิจิทัลนั้นมีหลายปัจจัย ซึ่งความคิดเห็นบนโลกออนไลน์ก็เป็นปัจจัย หนึ่งที่มีผลต่อการตัดสินใจในการลงทุน ทำให้การค้นคว้าอิสระนี้นำการขุดเหมืองข้อความและการ วิเคราะห์ความรู้สึกมาใช้ร่วมกับการสร้างแบบจำลอง โครงข่ายประสาทเทียมแบบวนกลับ (Recurrent Neural Network) เพื่อใช้สำหรับการวิเคราะห์ความคิดเห็นในสกุลเงินดิจิทัลโดยใช้ความคิดเห็นจาก เว็บไซต์คลังคลิปวิดีโอ เพื่อช่วยแนะนำและใช้ประกอบการตัดสินใจในการลงทุนในสกุลเงินดิจิทัล ซึ่งประกอบไปด้วยความคิดเห็นทั้งหมด 7,795 ความคิดเห็น จาก 47 วีดีโอโดยให้แบบจำลองทำนาย ความคิดเห็นออกมาว่าเป็นความคิดเห็นเชิงบวก หรือความคิดเห็นเชิงลบ พบว่าแบบจำลองโครงข่ายประสาทเทียมแบบวนกลับแบบ BERT (Bidirectional Encoder Representations from Transformer) ซึ่งให้ค่าความถูกต้องมากกว่าแบบอื่นและมีค่าอยู่ที่ 76 - 79 เปอร์เซ็นต์ นอกจากนี้ยังมีการนำความคิดเห็นที่ทำนายมาแล้วไปประกอบเป็นคุณลักษณะหนึ่งในการนำคุณลักษณะต่าง ๆ ของวิดีโอที่มีการพูดถึงสกุลเงินดิจิทัลทั้งหมด 15 คุณลักษณะ จึงใช้เทคนิคการวิเคราะห์องค์ประกอบ หลัก (Principal Component Analysis) เพื่อลดจำนวนตัวแปรที่จะใช้ประกอบการตัดสินใจในการ ลงทุนในสกุลเงินดิจิทัล จากการทดลองพบว่าใช้ Components เท่ากับ 4 จะสามารถอธิบายความแปรปรวน และสามารถอธิบายข้อมูลทั้งหมดได้ 78.05 เปอร์เซ็นต์

พรนิภา ตั้งพิทักษ์เสมอ (2567) การศึกษาวิจัยการค้นคว้าอิสระนี้ได้ทำการศึกษาและพัฒนาแบบจำลองเพื่อทำนายราคาขึ้นลง ของหุ้นโดยการวิเคราะห์จากข้อมูลข่าวของหุ้นกลุ่มอุตสาหกรรมธุรกิจการเงิน (ธนาคาร) แบบจำลอง สามารถทำการทำนายทำนายราคาของหุ้นได้ 3 คลาส คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

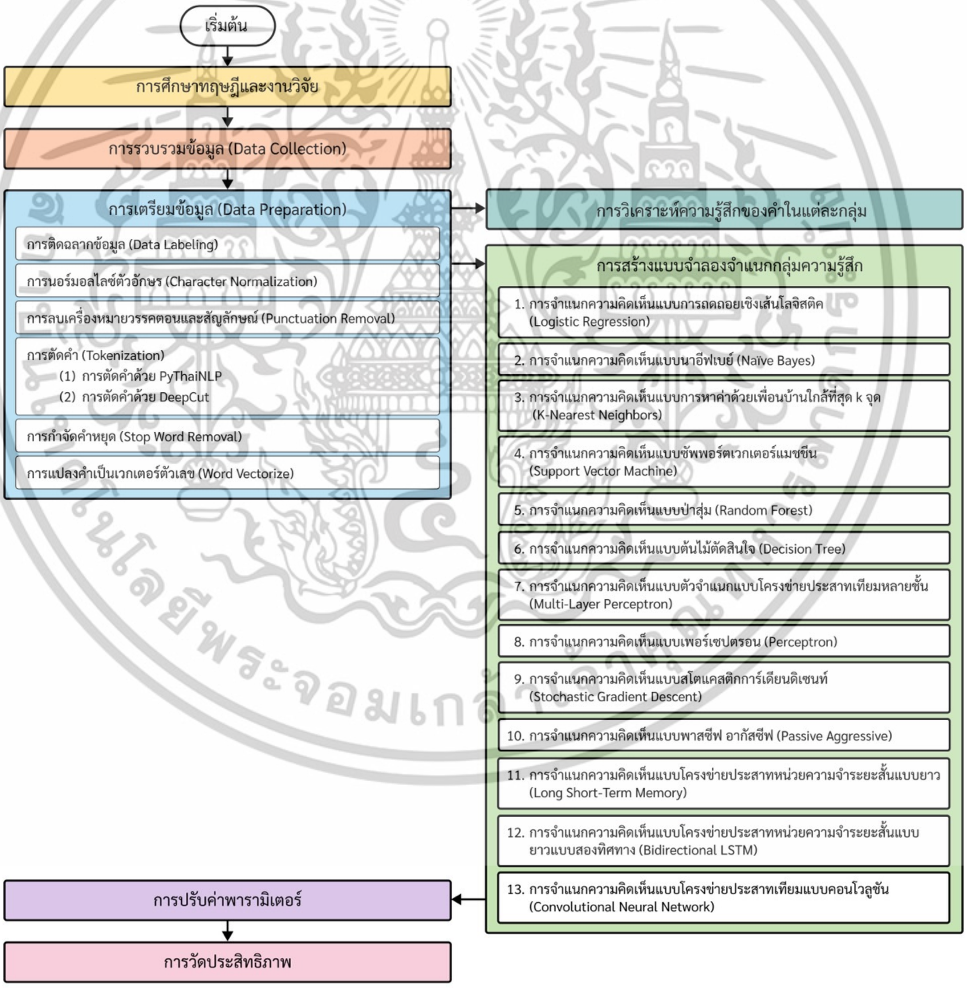
ทิศทางเป็นบวก ทิศทางเป็นลบ และทิศทางเป็นกลาง และทำการสร้างแบบจำลองหน่วยความจำระยะยาว-ระยะสั้น (Long Short-Term Memory) และแบบจำลองหน่วยเวียนกลับแบบมีประตู (Gate Recurrent Unit) และเปรียบเทียบ ประสิทธิภาพแบบจำลองจากข้อมูลของค่าความแม่นยำ (Accuracy) ค่าความเที่ยง (Precision) ค่าความไวและค่าระลึก (Recall) และค่าเอฟวัน (F1-Score) แบบจำลองที่ได้ค่าความแม่นยำมากที่สุด คือ แบบจำลองหน่วยความจำระยะยาว-ระยะสั้น ที่มีข้อมูลหัวข่าวสมดุล ซึ่งค่าความแม่นยำเท่ากับ 95.99% ค่าความเที่ยงเท่ากับ 96.00% ค่าความไวและค่าระลึก เท่ากับ 95.99% ค่าเอฟวันเท่ากับ 95.99%

Bagus Sujiwo. et al. (2021) ได้ทำการวิเคราะห์ความรู้สึก (Sentiment Analysis) ของนโยบายรัฐบาลอินโดนีเซียเกี่ยวกับการรับมือสถานการณ์ Covid-19 โดยอาศัยข้อมูลจากทวิตเตอร์ ซึ่งเป็นแหล่งข้อมูลความคิดเห็นของประชาชนในรูปแบบข้อความที่เผยแพร่บนแพลตฟอร์มสื่อสังคมออนไลน์ งานวิจัยนี้ได้นำเสนอการเปรียบเทียบประสิทธิภาพของโมเดลจำแนกประเภท 3 แบบ ได้แก่ แบบจำลองซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machine) นาอีฟเบย์ (Naïve Bayes) และโครงข่ายประสาทหน่วยความจำระยะสั้นแบบยาว (Long Short-Term Memory) ผลการทดลองพบว่า กลุ่มผู้ใช้ทวิตเตอร์ในประเทศอินโดนีเซียแสดงความรู้สึกในเชิงลบต่อมาตรการของภาครัฐเป็นส่วนใหญ่ โดยแบบจำลองที่ให้ผลแม่นยำที่สุดในการจำแนกความรู้สึก คือแบบจำลองซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machine) ซึ่งมีความถูกต้องสูงถึง 88.5%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
วิธีการดำเนินงานวิจัย

ในบทนี้จะนำเสนอวิธีการดำเนินงานวิจัยการวิเคราะห์ความรู้สึกต่อโครงการบ้านเพื่อคนไทยจากความคิดเห็นบนยูทูป โดยจะแบ่งออกเป็น 6 ขั้นตอน ได้แก่ การศึกษาทฤษฎีและงานวิจัย การรวบรวมข้อมูล การเตรียมข้อมูลโดยจะมีการใช้ตัวตัดคำ 2 แบบ คือ การใช้ตัวตัดคำ PyThaiNLP และการใช้ตัวตัดคำ DeepCut การวิเคราะห์คำในแต่ละกลุ่ม การสร้างแบบจำลองวิเคราะห์ความรู้สึก การปรับค่าพารามิเตอร์ และการวัดประสิทธิภาพของแบบจำลอง เพื่อให้การวิจัยในครั้งนี้เป็นไปตามวัตถุประสงค์จะแบ่งขั้นตอนการดำเนินงาน ดังรูปที่ 3.1 Workflows ของภาพรวมการทำงานทั้งหมด



รูปที่ 3.1 Workflows ของภาพรวมการทำงานทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1 การศึกษาทฤษฎีและงานวิจัย

วิธีการศึกษาเพื่อดำเนินงานวิจัยการขุดเหมืองข้อความและการวิเคราะห์ความรู้สึกต่อโครงการบ้านเพื่อคนไทยโดยใช้ความคิดเห็นจากเว็บไซต์ยูทูป ผู้ศึกษาได้ศึกษาและวิเคราะห์กระบวนการดำเนินงานจากงานวิจัยที่เกี่ยวข้องพร้อม ทั้งทฤษฎีต่าง ๆ จากเอกสาร โดยมีรายละเอียดดังนี้

1. ศึกษาข้อมูลเกี่ยวกับการขุดเหมืองข้อความ (Data Mining)
2. ศึกษาข้อมูลเกี่ยวกับการเตรียมข้อมูล (Data Preparation)
3. ศึกษาข้อมูลเกี่ยวกับการวิเคราะห์ความรู้สึก (Sentiment Analysis)
4. ศึกษาการนำเสนอข้อมูลด้วยแท็กคลาวด์ (Tag Cloud)
5. ศึกษาการประเมินผล (Evaluation)
6. ศึกษาและวิเคราะห์จากงานวิจัยที่เกี่ยวข้อง

3.2 การรวบรวมข้อมูล (Data Collection)

ในขั้นตอนนี้ ผู้พัฒนาได้ทำการดึงข้อมูลความคิดเห็นและรายละเอียดของวิดีโอจากเว็บไซต์ยูทูป โดยใช้ภาษา Python บน Google Colab ซึ่งเป็นเครื่องมือพัฒนาโปรแกรมที่ทำงานบนคลาวด์ (Cloud) ในการดึงข้อมูลจาก API (Application Programing Interface) ของ Google ซึ่ง YouTube Data APL v3 เป็น API ที่ให้การเข้าถึงข้อมูลของเว็บไซต์ YouTube โดยทำการดึงข้อมูลความคิดเห็นของโครงการบ้านเพื่อคนไทยในช่วงวันที่ 1 ธันวาคม 2567 ถึงวันที่ 30 เมษายน 2568 โดยมีอัลกอริทึมการเก็บรวบรวมข้อมูลโดยการดึงข้อมูลจากเว็บไซต์ยูทูป ดังรูปที่ 3.3

รูปที่ 3.2 Google Colaboratory หรือ Google Colab

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
[34] import googleapiclient.discovery
     import pandas as pd

     api_service_name = "youtube"
     api_version = "v3"
     DEVELOPER_KEY = "AIzaSyDW2KZxo30vQxgyLWv0Db_tXuFgPZzuvtc"

     youtube = googleapiclient.discovery.build(
         api_service_name, api_version, developerKey=DEVELOPER_KEY)

     request = youtube.commentThreads().list(
         part = "snippet",
         videoId = "Lou0KVyW868",
         maxResults = 500
     )

     response = request.execute()

     comments = []

     for item in response['items']:
       comment = item['snippet']['topLevelComment']['snippet']
       comments.append([
           comment['authorDisplayName'],
           comment['publishedAt'],
           comment['updatedAt'],
           comment['likeCount'],
           comment['textDisplay']
       ])

       df = pd.DataFrame(comments, columns=['author','published_at','updatedAt','likeCount','textDisplay'])

       df.to_csv('CommentFromYoutube.csv', index=False)
       df.to_excel('CommentFromYoutube.xlsx')
```

รูปที่ 3.3 อัลกอริทึมการเก็บรวบรวมข้อมูลโดยการดึงข้อมูลจากเว็บไซต์ยูทูป

จากรูปที่ 3.3 สามารถสรุปผลการเก็บข้อมูลความคิดเห็นได้ 2,236 ความคิดเห็น โดยมีรายละเอียดของความคิดเห็นประกอบด้วยคุณลักษณะ ดังตารางที่ 3.1 และมีตัวอย่างรายละเอียดของความคิดเห็น ดังรูปที่ 3.4

ตารางที่ 3.1 ตารางอธิบายชื่อคุณลักษณะของความคิดเห็นที่ได้จากการดึง API

ลำดับที่	ชื่อคอลัมน์	คำอธิบาย
1	author	ชื่อที่แสดงของผู้ใช้ที่โพสต์ความคิดเห็น
2	publishedAt	วันที่และเวลาที่อัปเดตความคิดเห็นครั้งแรก
3	updatedAt	วันที่และเวลาที่อัปเดตความคิดเห็นครั้งล่าสุด
4	likeCount	จำนวนการกดชอบทั้งหมด (การให้คะแนนเชิงบวก) ที่ความคิดเห็นได้รับ
5	textDisplay	ข้อความของความคิดเห็น

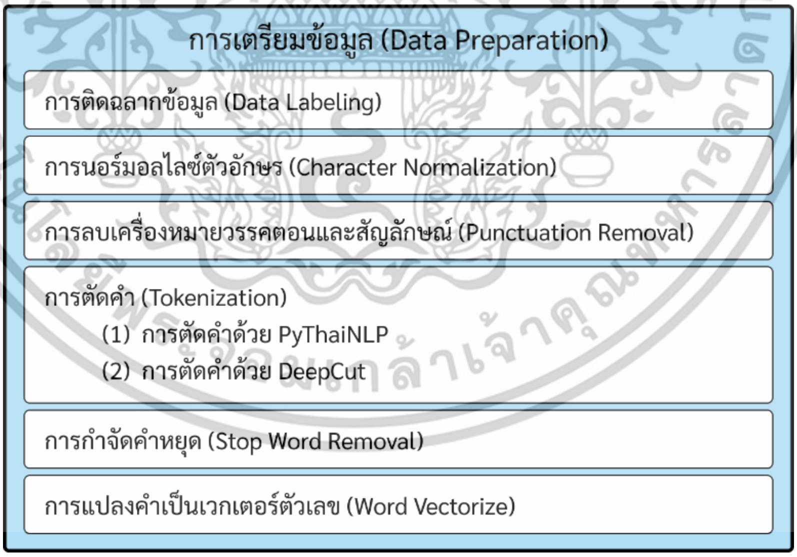
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	A	B	C	D	E	F
1		author	published_at	updatedAt	likeCount	textDisplay
2	1	@ดอกรักสีดำ555	20/01/25 03:48:08	20/01/25 03:48:08	0	มีบ้านเพื่อคนพม่าไหมครับ😂😂😂
3	2	@tatpichamichaud4813	20/01/25 03:49:50	20/01/25 03:49:50	151	บาปเพราะขัดขวางคนไทยอยากมีบ้านนี่แหละ บ้านและคอนโดส่วนใหญ่ควรเป็นคนไทยเป็นเจ้าของ ไม่ใช่ชาวต่างชาติ
4	3	@TungTangmay	20/01/25 03:50:39	20/01/25 03:50:39	41	มีบุญที่จะมีบ้าน
5	4	@pakiadame8800	20/01/25 03:51:03	20/01/25 03:51:03	0	แก่แล้วยังเห็นแก่ตัว มนุษย์ลุง
6	5	@thetoutychannel5789	20/01/25 03:53:13	20/01/25 03:53:13	4	บาปอีหยัง555
7	6	@vitsanurodbunma6266	20/01/25 03:55:25	20/01/25 03:55:25	355	ไม่น่าเชื่อว่าคำพูดแบบนี้ จะออกมาจากปาก ดร. เวรกรรมของชาติไทย
8	7	@baansuanranong7940	20/01/25 03:56:01	20/01/25 03:56:01	13	มันเป็นใคร งง
9	8	@สุภาพเรือนคำ-จ7บ	20/01/25 03:56:24	20/01/25 03:56:24	117	ตัวถ่วงความเจริญประเทศ
10	9	@pakpsh2887	20/01/25 03:56:32	20/01/25 03:56:32	2	พระที่ล้อศรีกาละครับที่คนแห่ไปทำบุญกันอ่ะมันบาปไหม
11	10	@boripatl.2057	20/01/25 03:56:49	20/01/25 03:56:49	4	ยังจำบ้านเอื้ออาทรได้ไหมจ๊ะ 😂😂😂
12	11	@prasopporn_sankla5592	20/01/25 03:57:07	20/01/25 03:57:07	0	คนพวกนี้ชอบกดคนไทยให้จนๆ มันน่าส่งกลับไปอยู่กับบรรพบุรุษที่เมืองจีนจริงๆ จ่ายค่าเช่าแค่30ปี อยู่ต่อชั่วลูกชั่วหลานอีก69ปี ไม่ต้องจ่ายตังค์ อันนี้ไม่คุ้มตรงไหน!
13	12	@ChadamaneeKrongsit	20/01/25 03:57:09	20/01/25 03:57:09	0	แล้วจะถามว่าให้ท่านนี้หาที่ไม่บาปให้รัฐบาลสร้างได้มั้ย
14	13	@wichaikhamkhoon2483	20/01/25 03:57:48	20/01/25 03:57:48	4	ตอนนี้โครงสร้างใหญ่ๆน่ากลัวหมดอ่ะเพราะถ้าใช้เหล็กของบริษัทจีน ที่อยู่ในไทย
15	14	@socool1627	20/01/25 03:58:06	20/01/25 03:58:06	6	พูดเอาหล่อ
16	15	@vivatyodsaga9922	20/01/25 03:58:57	20/01/25 03:58:57	102	ช่วยให้คนจนมีบ้าน เดินทางสะดวกไปทำงาน มีเวลาอยู่กับครอบครัวมากขึ้น มันบาปจริงเหรอ ที่ว่าบ้านมีแล้วเยอะแยะในกรุงเทพ มันก็ต้องไปกู้แบงค์ซื้อ ซึ่งเขาก็ปล่อยยาก และต้องมีเงินก้อนไว้ดาวอีก ดร.
17	16	@เหมียวผู้น่ารัก4642	20/01/25 03:59:23	20/01/25 03:59:23	78	ดร.มีบ้านแล้ว พูดยังไงก็ได้ ต้องนึกถึงคนไม่มีบ้านบ้ ราคาที่รัฐบาล ให้ หาไม่ได้แล้ว ไปซื้อบ้านเอกชน มีเงินดาวน์ รายได้น้อยซื้อไม่ไหวค่า ดร.😂
18	17	@heinrichkubis3167	20/01/25 04:00:57	20/01/25 04:00:57	0	บาปยังไง?
19	18	@TarisFireMage	20/01/25 04:01:05	20/01/25 04:01:05	0	จะดึงคะแนนเสียงที่เชียงใหม่กลับแน่นอน 😂
20	19	@Tservicee-d7e	20/01/25 04:01:27	20/01/25 04:01:27	3	ทำไม ถึงเป็นแบบนี้ วะ งง
21	20	@รัชนกมาเยอะ-ภ6ฆ	20/01/25 04:02:20	20/01/25 04:02:20	0	บาปน้อยกว่าถ้าให้คนต่างชาติซื้อบ้านอยู่
22	21	@พจน์บัวนาคhug4ix2it6U	20/01/25 04:02:41	20/01/25 04:02:41	0	สนับสนุนรัฐบาลอิ๊งทำโครงการบ้านเพื่อคนไทย
23	22	@สำราญเกษิต-ม9ภ	20/01/25 04:02:58	20/01/25 04:02:58	0	ให้ประชาชนมีที่อยู่ที่อาศัยเถอะท่าน
24	23	@uixbxjxjdjdj4532	20/01/25 04:03:02	20/01/25 04:03:02	0	พวกนักวิชาการเห็นได้ผลประโยชน์มันขัดขวางทุกอย่างอย่าไปเชื่อมัน
25	24	@mr.songvitsamorn6075	20/01/25 04:03:10	20/01/25 04:03:10	0	อย่าไปเชื่อ มันบาปหนักจากปาก ดร.
26	25	@waroonh4291	20/01/25 04:03:18	20/01/25 04:03:18	41	ผมว่า ดร. คนนี้ต้องมีผลประโยชน์ เกี่ยวกับอสังหาแพงๆ ที่เค้าไม่ซื้อกันตามปรกติ เพราะเค้ากู้ไม่ผ่าน ธนาคารไม่ปล่อย แล้วมันเกี่ยวกับบาปกรรมยังไง หล่ะครับ 55555555
27	26	@UdjcVdjdb	20/01/25 04:03:26	20/01/25 04:03:26	0	ไม่รู้จะพูดไปทำไมนะ ใครจะไม่ซื้อเขาก็ซื้อกันทั้งนั้นแหละ พูดไปทำไม พูดแล้วใครจะหยุดซื้อน่ะไม่หยุดแน่นอน
28	27	@BasNaHaa1	20/01/25 04:03:53	20/01/25 04:03:53	3	มันบาปรุป่าวไม่รู้ แต่การสร้างบ้านไม่เหมือนเกมที่จะกดปุ่มแล้วสร้างได้เลย รบ.ไม่เห็นพูดเรื่องที่ดิน(ล้านยูนิต) ธนาคาร(ไม่มีรายไหนออกมาขานรับ) มีแต่บ้าน**ตัวอย่างขายฝัน** กับการลงทะเบียน(ไม่ต่างไรจาก DWเงินหมื่น)เสาเข็ม การเข้าครม. การอนุมัติงบ เข้าใจว่าอยากมีบ้านแต่ช่วยคิดถึงจุดนี้ด้วย ว่าเค้าขายฝันหรือป่าว
29	28	@Nuemc	20/01/25 04:04:14	20/01/25 04:04:14	303	ส่วนตัวมีบ้านแล้ว ไม่คิดจะซื้อบ้านโครงการนี้ ไม่คิดจะขัดขวางคนที่จะซื้อบ้านโครงการนี้ แต่ก็อยากให้คนที่อยากได้บ้านโครงการนี้ได้กันทุกๆคน เพราะ**บ้านคือปัจจัย4ที่**สำคัญมากๆ เป็นกำลังใจให้คนที่ยังไม่มีบ้านทุกๆคน
30	29	@shamabird	20/01/25 04:04:55	20/01/25 04:04:55	8	มันบาปหนัก เพราะคนอสังหา ทำบ้าน จะขายได้น้อยลง ตอนนี้ธนาคารก็ยังปล่อยกู้ยาก เห็นใจคนทำ คอนโด ทำบ้านขาย หน่อยครับ อย่าทำโครงการพวกนี้**เลย มันบาปมาก** 55+ วงการอสังหา มีแต่เสือ จระเข้ทั้งนั้น บาปไหมถามใจดู ถ้าไม่บาปถ้าไปซื้อกับคนทำอสังหา บ้านแพงๆเงินกู้ดอกเยอะๆ นั่นเหละไม่บาป ธนาคารรวย **คนสร้างบ้านขายรวย**
31	30	@uixbxjxjdjdj4532	20/01/25 04:05:02	20/01/25 04:05:02	0	คิดอะไรไม่ออกพวกนักวิชาการ**โง่ทั้งนั้น**

รูปที่ 3.4 ตัวอย่างข้อมูลความคิดเห็นที่ดึงจากเว็บไซต์ยูทูป

3.3 การเตรียมข้อมูล (Data Preparation)

หลังจากทำการเก็บข้อมูลจากเว็บไซต์เรียบร้อย ชุดข้อมูลนั้นไม่สามารถนำไปใช้ในการวิเคราะห์ได้ทันทีจะต้องทำกระบวนการเตรียมข้อมูลก่อน โดยการเตรียมข้อมูลให้มีรูปแบบเป็นเวกเตอร์ แบ่งได้เป็น 6 ขั้นตอน ดังรูปที่ 3.5



รูปที่ 3.5 ขั้นตอนการเตรียมข้อมูล

ตัวอย่างการเตรียมข้อมูลนี้ ประโยคที่ใช้คือประโยค “^^ดีใจและดีใจกับพ่อแม่พี่น้องประชาชนคนไทย
ที่มีสิ่งดีๆ ที่เป็นจุดสูงสุดของมนุษย์ทุกคนก็คือบ้าน อยากมีบ้าน คำนี้เคยได้ยินคนพูดกัน
นายกและรัฐบาลทุกท่านทำได้แล้วเห็นผลงาน
ขึ้นทุกทีทุกวันแล้ว ดีจังเลยครับ” ซึ่งจะผ่านขั้นตอนการเตรียม 6 ขั้นตอน ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.1 การติดฉลากข้อมูล (Data Labeling)

หลังจากได้ทำการเก็บรวบรวมข้อมูลความคิดเห็นจากเว็บไซต์ยูทูปมาเรียบร้อยแล้ว ต่อมาต้องทำการติดฉลากให้ความคิดเห็น เนื่องจากต้องนำความคิดเห็นพร้อมผลเฉลยไปเป็นข้อมูลชุดการเรียนรู้ (Tanning Set) สำหรับใช้เทรนแบบจำลอง เพื่อสร้างแบบจำลองวิเคราะห์ความรู้สึกอัตโนมัติ

การทำติดสลากความคิดเห็นทั้งหมดจำนวน 2,236 รายการ โดยแบ่งออกเป็น 3 ผลเฉลย ประกอบด้วย Pos คือ ความคิดเห็นเชิงบวก Neu คือ ความคิดเห็นเชิงกลาง และ Neg คือ ความคิดเห็นเชิงลบ สามารถกำหนดเกณฑ์ได้ตามเนื้อหาและน้ำเสียงของข้อความตารางที่ 3.2 ตารางอธิบายกำหนดเกณฑ์ของความคิดเห็น

ตารางที่ 3.2 ตารางอธิบายกำหนดเกณฑ์ของความคิดเห็น

ประเภท	หลักเกณฑ์	ตัวอย่างข้อความ
ความคิดเห็นเชิงบวก (Positive)	- แสดงความพึงพอใจ ชื่นชม สนับสนุน หรือมีความรู้สึกในเชิงบวกต่อโครงการ - ใช้ภาษาที่สะท้อนถึงความหวัง ความคาดหวัง ความดีงามของโครงการ หรือความประทับใจ	“โครงการดีมาก เหมาะกับคนรายได้น้อย” “อยากได้บ้านแบบนี้มานานแล้ว” “ราคาถูกมากเมื่อเทียบกับคุณภาพ”
ความคิดเห็นเชิงกลาง (Neutral)	- แสดงความคิดเห็นในทางลักษณะ สอบถาม ข้อสงสัย เสนอแนะ หรือบอกเล่าข้อเท็จจริงโดยไม่แสดงอารมณ์หรือทัศนคติชัดเจน - ไม่มีถ้อยคำชื่นชมโดยตรงหรือแสดงความไม่พอใจ	“ขอรายละเอียดเพิ่มเติมเกี่ยวกับการจอง” “เหมือนเป็นโครงการเช่าใช่ไหม” “มีโครงการในจังหวัดอื่นไหม”
ความคิดเห็นเชิงลบ (Negative)	- แสดงความไม่พอใจ ตำหนิ ไม่เห็นด้วย หรือมีความรู้สึกในเชิงลบต่อโครงการ - ใช้ภาษาที่สื่อถึงความไม่ไว้วางใจ ความผิดหวัง หรือวิจารณ์ในเชิงสร้างสรรค์หรือรุนแรง	“โครงการดีแต่สงสัยว่าจะได้บ้านจริงหรือเปล่า” “กลัวโดนหลอกอีกแล้ว เหมือนโครงการก่อน” “บ้านดูเล็กเกินไปสำหรับครอบครัว”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยจะแสดงตัวอย่างการติดฉลากให้ชุดข้อมูล ตัวอย่างข้อมูลที่ทำการติดฉลากความคิดเห็นเชิงบวก ความคิดเห็นเชิงกลาง และความคิดเห็นเชิงลบ ดังรูปที่ 3.6 – 3.8 ตามลำดับ

	A	B
1	**text**	**sentiment**
2	สวดยอด	Pos
3	สุดยอด	Pos
4	ชอบผลงานรัฐบาลเพื่อไทย ทำงานเป็น จบงานฉับไวย#เป็นกรรมลังใจให้เพื่อทวาย	Pos
5	บ้านคนเพื่อไทย ชัดเจนจริงๆ 😆😆😆	Pos
6	👏👏👏	Pos
7	รัฐบาลที่มาทำหน้าที่เพื่อคนไทยจริงๆ	Pos
8	สลิ่ม ส้มเน่า ตัวไหนยังไม่มีบ้านอยู่จองได้น่ะคนับ	Pos
9	เริ่ด!ทำดีมากค่ะ เดือนละ 4 พันบาท น้อยคนที่จะอยู่ได้ถึง 90 ปี ถือว่าคุ้มมาก	Pos
10	อยาดจองแต่ติดทอดกล้วย	Pos
11	ด้อมส้วมลงกันตรึม	Pos
12	ใครได้บอกนะครับ อยากไปรีวิว	Pos
13	คนจอง12ล้านคนใน1ชั่วโมง	Pos
14	มีให้ขนาดนี้ถือว่าครบคันสุดยอดแล้ว	Pos
15	14:01 เว็บล่ม เขาจองกันด้วยวิธีไหน กัน ผมอยากจองบ้างจัง	Pos
16	กูเห็นมีแต่คนด่าแล้วไปจองกัน😂😂😂	Pos
17	เป็นหนี้ไปพร้อมๆกันค่ะผ่อนจนตายกันไปข้างนึง99ปีว่าซ่าน❗ 😂😂😂	Pos
18	กดจองดิ้	Pos
19	👍👍❤️	Pos
20	ราคานี้ ทำเลนี้ ห้องเปล่าก็น่าสนใจอยู่ดีนั่นแหละ	Pos

รูปที่ 3.6 ตัวอย่างข้อมูลที่ทำการติดฉลากชุดข้อมูลเชิงบวก

	A	B
1	**text**	**sentiment**
2	ดีหมด กลัวแค่คนมาเอาสิทธิ์แล้วแอบไปปล่อยเช่า มันจะผิดวัตถุประสงค์ของโครงการ	Neu
3	ใครเป็นรับเหมาโครงการนี้คะ	Neu
4	ศึกษาข้อมูลดีๆก่อนซื้อเช่านะครับ	Neu
5	เข้าใจแซว	Neu
6	ภาพชวนเชื่อ	Neu
7	มีใครด่ารัฐบาลแล้วจองยกมือขึ้น	Neu
8	สะดวกชายสบายหญิง😂😂😂	Neu
9	อยากถามค่ำ. เดียว. ถ้าเป็นคอนโด จะจอดรถตรงไหน. 2000ห้อง. ที่จอดรถต้องขนาดไหน	Neu
10	อยากให้มีหน่วยงานอิสระดีๆตรวจสอบโครงการเหลือเกิน🤓	Neu
11	จะซื้อก็ซื้อ ไม่ซื้อก็ไม่ซื้อ ใช้สมอง!อย่าแซะ	Neu
12	ชอบตรงที่ต้องจับฉลากนี่ละเหมือนต้องมีบุญด้วยถึงจะได้	Neu
13	เงื่อนไข เค้าบอกว่าถ้าคุณสมบัติผ่าน ต่อมาจะต้องลุ้นจับสลากกันอีก ว่าใครจะได้ 😂😂	Neu
14	ทำมารับรองใครกันน่า	Neu
15	ถ้าชักโครกธรรมดาตัวเทียบราคามันเยอะครับ	Neu
16	ขายให้จีน?	Neu
17	555นึกว่าห้องสำหรับคนที่อาศัยอยู่คนเดียวและครัวทำกับข้าวไม่มีหรือค่ะ555	Neu
18	มีสายฉีดดากไหม	Neu
19	อยู่ชั้น14มั้ยครับ	Neu
20	เข้าใจทำโลโก้โครงการ ประชาสัมผัสพรรคไปในตัว55	Neu

รูปที่ 3.7 ตัวอย่างข้อมูลที่ทำการติดฉลากชุดข้อมูลเชิงกลาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	A	B
1	**text**	**sentiment**
2	กำลังขื้อยู่ช็อตดูดละมึงเอ้ย ไม่อยากจะคิด555	Neg
3	ทำมาให้แย่งกัน..เผลอๆคนที่ได้ก็พวกมีบ้านอยู่แล้วอีกนั้นแหละ..	Neg
4	อันนี้เหี้ยจัดเอาชื่อพรรคมาสร้างโครงการ ตอนที่ประยุทธ์ทำบัตรประชารัฐเพื่อไทยด่ายับ พอตัวเองเป็นนี่ ทำได้แบบหน้าด้านชิบผายเหมือนเอาเงินพรรคตัวเองมาทำ	Neg
5	คนไทยอยากเห็นนายกรัฐมนตรีลาออก	Neg
6	นึกว่าของฟรี ผ่อนแบบไม่มีอนาคต	Neg
7	ซากปรักหักพังเอื้ออาทรภาคแรกยังเป็นอนุสรณ์วัฒนา เมืองสุข ศาลพิพากษาจำคุก99ปี คดีทุจริตคอรัปชั่นบ้านเอื้ออาทร!!	Neg
8	กลัวไฟดูดดูด😅😅😅	Neg
9	จองชาตินี้.....เป็นหนี้ถึงชาติ.....	Neg
10	สุดท้ายเป็น condo สลัมเพราะไม่มีนิติบุคคลดูแล ไม่มีเก็บค่าส่วนกลาง	Neg
11	ใครคิดว่าคุ้มก็ซื้อค่ะ ใครว่าไม่คุ้มก็ไม่ต้องซื้อ ส่วนตัวไม่ซื้อค่ะ เพราะไม่มีตังค์	Neg
12	ดูแล้วเหมือนโครงการผลาญภาษีคอรัปชั่นที่ดินสาธารณะมากกว่า ในประเทศนี้มีบ้านร้างบ้านหลุดจำนองเยอะแยะทำไมไม่วางแผนเอามาทำให้เป็นประโยชน์ ทำใหม่ไปเรื่อย	Neg
13	ผ่อนจนตายยังไม่หมดต้องไปเกิดใหม่มาผ่อนต่ออิๆๆๆ	Neg
14	แหล่งแดกเงินทอนอีกระ	Neg
15	เอาลุงตู่กลับมาสะ555	Neg
16	ทำสัญญาถึง99ปี ตายไปตกนรกยังต้องชดใช้หนี้อยู่ เวรกรรมมันติดตัว อย่าหาทำ🤣🤣🤣	Neg
17	โครงการรัฐ / โครงการสวัสดิ์การ ต้องมีการประมูล ยกตัวอย่าง แฟลตตำรวจ สุดท้าย ตอนนี้ใครได้กันแน่	Neg
18	โครงการเพื่อไทยจริงๆ งานนี้คงหลายหมื่นล้านครับ	Neg
19	กลัวจะเหลือแต่เสาเหมือนพักลุงกำนัน	Neg
20	เชิญตามบายจร้า เช่าไปจร้า	Neg

รูปที่ 3.8 ตัวอย่างข้อมูลที่ทำการติดฉลากชุดข้อมูลเชิงลบ

จากตัวอย่างข้อมูลความคิดเห็นประโยค "^^ดีใจและดีใจกับพ่อแม่พี่น้องประชาชนคนไทย
ที่มีสิ่งดีๆ ที่เป็นจุดสูงสุดของมนุษย์ทุกคนก็คือบ้าน อยากมีบ้าน คำนี้เคยได้ยินคนพูดกัน
 นายกและรัฐบาลทุกท่านทำได้แล้วเห็นผลงาน
ขึ้นทุกทีทุกวันแล้ว ดีจังเลยครับ" จะเห็นได้ว่าประโยคมีความหมายไปในทางบวกต่อโครงการบ้านเพื่อคนไทย เลยติดฉลากเป็น "Pos"

3.3.2 การนอร์มอลไลซ์ตัวอักษร (Character Normalization)

การนอร์มอลไลซ์ตัวอักษรคือขั้นตอนในการทำให้ตัวอักษรหรือข้อความที่เขียนต่างกันแต่มีความหมายเดียวกัน ถูกแปลงให้อยู่ในรูปแบบเดียวกัน เพื่อให้ข้อมูลมีความสม่ำเสมอและสะอาดก่อนนำไปวิเคราะห์ต่อ ช่วยลดความหลากหลายของรูปแบบการเขียนที่อาจหมายถึงสิ่งเดียวกัน ซึ่งสำคัญมากในการประมวลผลภาษาธรรมชาติ (NLP)

ตารางที่ 3.3 ตัวอย่างในการนอร์มอลไลซ์ตัวอักษรในภาษาไทย

ก่อน Normalize	หลัง Normalize	ความหมาย
ดีมั้ยยยยย	ดีมั้ย	ตัดอักขระซ้ำ
ก้อ ก็นะ	ก็	แปลงคำให้เป็นคำมาตรฐาน

จากตัวอย่างประโยค "^^ดีใจและดีใจกับพ่อแม่พี่น้องประชาชนคนไทย
ที่มีสิ่งดีๆ ที่เป็นจุดสูงสุดของมนุษย์ทุกคนก็คือบ้าน อยากมีบ้าน คำนี้เคยได้ยินคนพูดกัน
นายกและรัฐบาลทุกท่านทำได้แล้วเห็นผลงาน
ขึ้นทุกทีทุกวันแล้ว ดีจังเลยครับ" จะเห็นได้ว่าภายในข้อความไม่มี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอักษรประโยคที่เขียนตัวอักษรซ้ำและคำที่ไม่เป็นมาตรฐานจึงได้ข้อความเหมือนเดิม แสดงตัวอย่างผลลัพธ์ที่ได้จากการนอร์มอลไลซ์ตัวอักษร ดังตารางที่ 3.4

ตารางที่ 3.4 ตัวอย่างขั้นตอนการนอร์มอลไลซ์อักษร

ขั้นตอน	คำอธิบาย	ผลลัพธ์
การนอร์มอลไลซ์อักษร	การทำให้ตัวอักษรหรือข้อความที่เขียนต่างกันแต่มีความหมายเดียวกัน	^^ดีใจและดีใจกับพ่อแม่พี่น้องประชาชนคนไทย ที่มีสิ่งดีๆ ที่เป็นจุดสูงสุดของมนุษย์ทุกคนก็คือบ้าน อยากมีบ้าน คำนี้เคยได้ยินคนพูดกัน นายกและรัฐบาลทุกท่านทำได้แล้วเห็นผลงาน ขึ้นทุกทีทุกวันแล้ว ดีจังเลยครับ

3.3.3 การลบเครื่องหมายวรรคตอนและสัญลักษณ์ (Punctuation Removal)

จากตัวอย่างประโยค "^^ดีใจและดีใจกับพ่อแม่พี่น้องประชาชนคนไทย
ที่มีสิ่งดีๆ ที่เป็นจุดสูงสุดของมนุษย์ทุกคนก็คือบ้าน อยากมีบ้าน คำนี้เคยได้ยินคนพูดกัน
นายกและรัฐบาลทุกท่านทำได้แล้วเห็นผลงาน
ขึ้นทุกทีทุกวันแล้ว ดีจังเลยครับ" จะเห็นได้ว่าภายในข้อความมีอักขระพิเศษ สัญลักษณ์ที่ไม่ได้ช่วยเพิ่มความหมายให้กับแบบจำลอง แสดงตัวอย่างผลลัพธ์ที่ได้จากการลบเครื่องหมายวรรคตอนและสัญลักษณ์ ดังตารางที่ 3.5

ตารางที่ 3.5 ตัวอย่างขั้นตอนการลบเครื่องหมายวรรคตอนและสัญลักษณ์ (Punctuation Removal)

ขั้นตอน	คำอธิบาย	ผลลัพธ์
การลบเครื่องหมายวรรคตอนและสัญลักษณ์	ลบเครื่องหมายวรรคตอน เช่น "!", "?", "+", "ๆ" ลบแท็ก HTML เช่น , <span>, <div> และสัญลักษณ์ที่ไม่ได้ช่วยเพิ่มความหมายให้กับแบบจำลอง	ดีใจและดีใจกับพ่อแม่พี่น้องประชาชนคนไทยที่มีสิ่งดีๆ ที่เป็นจุดสูงสุดของมนุษย์ทุกคนก็คือบ้าน อยากมีบ้าน คำนี้เคยได้ยินคนพูดกันนายกและรัฐบาลทุกท่านทำได้แล้วเห็นผลงานขึ้นทุกทีทุกวันแล้ว ดีจังเลยครับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.4 การตัดคำ (Tokenization)

การตัดคำคือการนำประโยคมาแบ่งออกเป็นคำต่างๆ ในงานวิจัยนี้จะใช้ตัวตัดคำ 2 แบบ ดังนี้ แบบที่ 1 การตัดคำด้วย PyThaiNLP และแบบที่ 2 การตัดคำด้วย DeepCut ในการตัดคำ จากตัวอย่างประโยค “ดีใจและดีใจกับพ่อแม่พี่น้องประชาชนคนไทยที่มีสิ่งดีๆ ที่เป็นจุดสูงสุดของมนุษย์ทุกคนก็คือบ้าน อยากมีบ้าน คำนี้เคยได้ยินคนพูดกันนายกและรัฐบาลทุกท่านทำได้แล้วเห็นผลงานขึ้นทุกทีทุกวันแล้ว ดีจังเลยครับ” เมื่อนำมาตัดคำแล้ว แสดงตัวอย่างผลลัพธ์ที่ได้จากการตัดคำ ดังตารางที่ 3.6

ตารางที่ 3.6 ตัวอย่างขั้นตอนการตัดคำ (Tokenization)

ขั้นตอน	ตัวตัดคำ	ผลลัพธ์
การตัดคำ	PyThaiNLP	ดีใจ และ ดีใจ กับ พ่อแม่ พี่น้อง ประชาชน คน ไทย ที่ มี สิ่ง ดี ๆ ที่ เป็น จุดสูงสุด ของ มนุษย์ ทุกคน ก็ คือ บ้าน อยาก มี บ้าน คำ นี้ เคย ได้ ยิน คนพูด กัน นายก และ รัฐบาล ทุกท่าน ทำได้ แล้ว เห็น ผลงาน ขึ้น ทุกที ทุกวัน แล้ว ดี จัง เลย ครับ
	DeepCut	ดีใจ และ ดีใจ กับ พ่อแม่พี่น้อง ประชาชน คน ไทย ที่ มี สิ่ง ดี ๆ ที่ เป็น จุด สูง สุด ของ มนุษย์ ทุก คน ก็ คือ บ้าน อยาก มี บ้าน คำ นี้ เคย ได้ ยิน คน พูด กัน นายก และ รัฐบาล ทุก ท่าน ทำ ได้ แล้ว เห็น ผล งาน ขึ้น ทุก ที ทุก วัน แล้ว ดี จัง เลย ครับ

3.3.5 การกำจัดคำหยุด (Stop Word Removal)

การกำจัดคำหยุดคือการลบคำที่ไม่มีความสำคัญต่อการวิเคราะห์ความหมายของประโยคออกไป เช่นคำที่มักปรากฏบ่อย แต่ไม่ได้สื่อเนื้อหาสาระเฉพาะเจาะจง ตัวอย่างคำดังรูปที่ 3.9

```
# ดึงคำหยุดภาษาไทย (stopwords) มาเก็บเป็น list
thaiStopords = list(thai_stopwords())
thaiStopords
```

```
['คราที่',
 'ฯลฯ',
 'ช่วงแรก',
 'บางที่',
 'ฝ่ายใด',
 'พวกมึง',
 'แล้วแต่',
 'โดย',
```

รูปที่ 3.9 ตัวอย่างคำที่ไม่มีความสำคัญภาษาไทย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากทำการกำจัดคำหยุดเสร็จจะได้ประโยค ที่มีเฉพาะคำที่มีความหมายต่อความรู้สึกของโครงการ ดังตารางที่ 3.7

ตารางที่ 3.7 ตัวอย่างขั้นตอนการกำจัดคำหยุด (Stop Word Removal)

ขั้นตอน	คำอธิบาย	ตัวตัดคำ	ผลลัพธ์
การกำจัดคำหยุด	ลบคำที่ไม่มีความสำคัญต่อการวิเคราะห์ความหมายของประโยคออกไป เช่น “และ” “กับ” “ที่” “มี” “สิ่ง” “ของ” เป็นต้น	PyThaiNLP	ดีใจ ดีใจ พ่อแม่ พี่น้อง ประชาชน คนไทย ดี ๆ จุดสูงสุด มนุษย์ บ้าน บ้าน ได้ยิน คนพูด นายก รัฐบาล ทุกท่าน ทำได้ ผลงาน ดี
		DeepCut	ดีใจ ดีใจ พ่อแม่ พี่น้อง ประชาชน คน ไทย ดี ๆ จุด มนุษย์ คน บ้าน บ้าน ได้ยิน คน นายก รัฐบาล ท่าน ทำ งาน ดี

3.3.6 การแปลงคำเป็นเวกเตอร์ตัวเลข (Word Vectorize)

การสร้างแบบจำลองป่าสุ่มจะใช้ไลบรารี่ของ Sklearn คือ sklearn.ensemble ที่ใช้สำหรับการสร้างแบบจำลองป่าสุ่มสำหรับงานประเภทการจำแนก

เครื่องมือหรือเทคนิคที่ใช้แปลง "คำ" หรือ "ข้อความ" ให้อยู่ในรูปแบบของ เวกเตอร์ตัวเลข เพื่อให้สามารถนำไปใช้กับโมเดล Machine Learning หรือ Deep Learning ได้ เนื่องจากแบบจำลองเหล่านี้ไม่สามารถเข้าใจข้อความได้โดยตรง ต้องแปลงเป็นตัวเลขก่อน แสดงตัวอย่างผลลัพธ์ที่ได้จากการตัดคำ ดังตารางที่ 3.8

ตารางที่ 3.8 ตัวอย่างขั้นตอนการแปลงคำเป็นเวกเตอร์ตัวเลข (Word Vectorize)

ขั้นตอน	คำอธิบาย	ตัวตัดคำ	ผลลัพธ์
การกำจัดคำหยุด	ลบคำที่ไม่มีความสำคัญต่อการวิเคราะห์ความหมายของประโยคออกไป เช่น “และ” “กับ” “ที่” “มี” “สิ่ง” “ของ” เป็นต้น	PyThaiNLP	{'ดีใจ': 4, 'พ่อแม่': 12, 'พี่น้อง': 11, 'ประชาชน': 9, 'คนไทย': 1, 'ดี': 3, 'ๆ': 16, 'จุดสูงสุด': 2, 'มนุษย์': 13, 'บ้าน': 8, 'ได้ยิน': 15, 'คนพูด': 0, 'นายก': 7, 'รัฐบาล': 14, 'ทุกท่าน': 6, 'ทำได้': 5, 'ผลงาน': 10}

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.8 ตัวอย่างขั้นตอนการแปลงคำเป็นเวกเตอร์ตัวเลข (Word Vectorize) (ต่อ)

ขั้นตอน	คำอธิบาย	ตัวตัดคำ	ผลลัพธ์
การกำจัดคำหยุด	ลบคำที่ไม่มีความสำคัญต่อการวิเคราะห์ความหมายของประโยคออกไป เช่น “และ” “กับ” “ที่” “มี” “สิ่ง” “ของ” เป็นต้น	DeepCut	{'ดีใจ': 4, 'พ่อแม่': 11, 'พี่น้อง': 10, 'ประชาชน': 9, 'คน': 0, 'ไทย': 15, 'ดี': 3, 'ๆ': 16, 'จุด': 2, 'มนุษย์': 12, 'บ้าน': 8, 'ได้ยิน': 14, 'นายก': 7, 'รัฐบาล': 13, 'ท่าน': 6, 'ทำ': 5, 'งาน': 1}

3.4 การวิเคราะห์ความรู้สึกของคำในแต่ละกลุ่ม

ในการวิจัยครั้งนี้ได้มีการวิเคราะห์ข้อความความคิดเห็นต่อโครงการบ้านเพื่อคนไทย โดยแบ่งออกเป็น 3 กลุ่ม ได้แก่ ความคิดเห็นเชิงบวก ความคิดเห็นเชิงกลางและความคิดเห็นเชิงลบ เพื่อให้สามารถทำความเข้าใจเนื้อหาที่ความสำคัญในแต่ละมุมมองได้อย่างชัดเจน โดยนำข้อมูลความถี่ของคำในแต่ละกลุ่มมาสร้างเป็นภาพ Word Cloud ซึ่งเป็นเครื่องมือที่ช่วยแสดงผลคำที่พบบ่อยในรูปแบบที่เข้าใจง่าย โดยคำที่ปรากฏบ่อยจะมีขนาดใหญ่กว่าคำที่ปรากฏน้อย ช่วยให้สามารถมองเห็นภาพรวมของประเด็นให้ความสำคัญในแต่ละกลุ่มความคิดเห็นได้อย่างชัดเจน

การวิเคราะห์คำในลักษณะนี้ช่วยให้สามารถเข้าใจทัศนคติและมุมมองในแต่ละกลุ่มความคิดเห็นได้ดียิ่งขึ้น และยังเป็นประโยชน์ต่อการวางนโยบาย การสื่อสารสาธารณะ หรือการวางแผนพัฒนาในด้านต่าง ๆ ที่เกี่ยวข้องได้อีกด้วย

3.5 การจำแนกความรู้สึกจากความคิดเห็น

หลังจากได้ทำการแปลงข้อมูลเป็นเวกเตอร์ด้วยภาษาไพธอนแล้ว ข้อมูลเหล่านี้จะถูกนำมาเข้ากระบวนการฝึกฝนข้อมูลด้วยอัลกอริทึมการจำแนกประเภททั้งหมด 13 แบบ คือ การถดถอยเชิงเส้นโลจิสติค (Logistic Regression) นาอีฟเบย์ (Naïve Bayes) การหาค่าด้วยเพื่อนบ้านใกล้ที่สุด k จุด (K-Nearest Neighbors) ซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machine) ป่าสุ่ม (Random Forest) ต้นไม้ตัดสินใจ (Decision Tree) ตัวจำแนกแบบโครงข่ายประสาทเทียมหลายชั้น (Multi-Layer Perceptron) เพอร์เซปตรอน (Perceptron) สโตแคสติกการ์เดียนดิเซนท์ (Stochastic Gradient Descent) พาสซีฟ อากัสซีฟ (Passive Aggressive) โครงข่ายประสาทหน่วยความจำระยะสั้นแบบยาว(Long Short-Term Memory) โครงข่ายประสาทหน่วยความจำระยะสั้นแบบยาวแบบสองทิศทาง (Bidirectional LSTM) โครงข่ายประสาทเทียมแบบคอนโวลูชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Convolutional Neural Network) โดยใช้ไลบรารี่ Sklearn และ TensorFlow ซึ่งเป็นไลบรารี่สำหรับสร้างโมเดลการเรียนรู้โดยภาษาไพธอน ซึ่งหลังจากข้อมูลผ่านกระบวนการเตรียมข้อมูลในหัวข้อที่ 3.3 และนำข้อมูลมาแบ่งเป็นชุดข้อมูลเรียนรู้ 70% ชุดทดสอบ 30% การแบ่งข้อมูลแบบนี้มีข้อดีหลายประการโดยเฉพาะกับข้อมูลขนาดเล็ก แบ่งโดยการสุ่ม หลังจากนั้นจะถูกนำไปสร้างโมเดล

3.5.1 การถดถอยเชิงเส้นโลจิสติค (Logistic Regression)

การสร้างแบบจำลองการถดถอยเชิงเส้นโลจิสติคจะใช้หนึ่งในไลบรารี่ย่อยของ Sklearn คือ sklearn.linear_model.LogisticRegression ในงานวิจัยนี้มีการปรับจูนค่า C ซึ่งค่าพารามิเตอร์ C ใน Logistic Regression เป็นพารามิเตอร์ที่ใช้ในการควบคุมการแพร่กระจาย (Regularization) ของแบบจำลอง Regularization ช่วยในการป้อนกันปัญหา Overfitting การหาค่าพารามิเตอร์ที่เหมาะสมที่สุดสำหรับการถดถอยเชิงเส้นโลจิสติค ค่า penalty ที่เป็นวิธีการ Regularization และปรับจูนค่า solver คืออัลกอริทึมสำหรับการปรับค่าพารามิเตอร์ โดยใช้ OptunaSearchCV หาค่าพารามิเตอร์ที่ดีที่สุดโดยจะกำหนดช่วงของค่า C ที่ต้องการทดลอง คือ 0.01 ถึง 100 ค่า penalty ที่ต้องการทดลอง คือ ‘12’ และค่า solver คือ ‘lbfgs’

3.5.2 นาอีฟเบย์ (Naïve Bayes)

การสร้างแบบจำลองนาอีฟเบย์จะใช้ไลบรารี่ของ Sklearn คือ sklearn.naive_bayes ในงานวิจัยนี้มีการปรับจูนค่า alpha ซึ่งเป็นค่าพารามิเตอร์ที่ช่วยในการแก้ปัญหาข้อมูลที่ไม่มีในชุดฝึก โดยการเพิ่มค่าเล็กน้อยเข้าไปในจำนวนการนับของแต่ลักษณะ เพื่อไม่ให้ค่าความน่าจะเป็นเป็นศูนย์ ค่า alpha ที่เหมาะสมที่สุดสำหรับแบบจำลองนาอีฟเบย์ โดยใช้ OptunaSearchCV หาค่าพารามิเตอร์ที่ดีที่สุด โดยกำหนดช่วงของค่า alpha ที่ต้องการทดลองที่ 0.01 ถึง 1

3.5.3 การหาค่าด้วยเพื่อนบ้านใกล้ที่สุด k จุด (K-Nearest Neighbors)

การสร้างแบบจำลองการหาค่าด้วยเพื่อนบ้านใกล้ที่สุด k จุด จะใช้ไลบรารี่ของ Sklearn คือ sklearn.neighbors ในงานวิจัยนี้มีการปรับจูนค่า n_neighbors คือจำนวนของเพื่อนบ้าน เป็นค่าพารามิเตอร์ที่สำคัญที่สุดของ K-Nearest Neighbors และค่า weights คือวิธีถ่วงน้ำหนักหรือการให้คะแนนเพื่อนบ้าน โดยใช้ OptunaSearchCV หาค่าพารามิเตอร์ที่ดีที่สุด ในขั้นตอนการปรับจูนโดยกำหนดช่วงของค่า n_neighbors ที่ต้องการทดลองที่ คือ จำนวนเต็มระหว่าง 3 ถึง 9 และค่า weights ที่ต้องการทดสอบ คือ ‘uniform’ และ ‘distance’

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.4 ซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machine)

การสร้างแบบจำลองซัพพอร์ตเวกเตอร์แมชชีนจะใช้ไลบรารี่ของ Sklearn คือ sklearn.svm ในงานวิจัยนี้มีการปรับจูนค่า C ซึ่งเป็นค่าพารามิเตอร์ที่ควบคุม trade-off ระหว่างการมีขอบเขตที่กว้างกับการจัดกลุ่มข้อมูลอย่างถูกต้อง ค่าที่ต่ำจะทำให้ขอบเขตกว้างขึ้นแต่มีการยอมให้ข้อมูลบางจุดอยู่ผิดกลุ่ม ในขณะที่ค่าสูงจะพยายามจัดกลุ่มข้อมูลให้ถูกต้องมากที่สุด ฟังก์ชัน kernel จะใช้แปลงข้อมูลให้อยู่ในมิติที่สูงขึ้น โดยใช้ OptunaSearchCV หาค่าพารามิเตอร์ที่ดีที่สุด โดยจะกำหนดช่วงของค่า C ที่ต้องการทดลอง คือ 0.1 ถึง 100 ค่า kernel ที่ทดลอง คือ ‘linear’ และ ‘rbf’ และค่า gamma ที่ทดลอง คือ ‘scale’ และ ‘auto’

3.5.5 ป่าสุ่ม (Random Forest)

การสร้างแบบจำลองป่าสุ่มจะใช้ไลบรารี่ของ Sklearn คือ sklearn.ensemble ในงานวิจัยนี้มีการปรับจูนคือ n_etimators ซึ่งคือจำนวนต้นไม้ที่จะถูกสร้างในแบบจำลองป่าสุ่ม ซึ่งจำนวนต้นไม้มากขึ้นความแม่นยำจะดีขึ้น แต่ถ้าจำนวนต้นไม้มากเกินไปก็ไม่ได้สามารถเพิ่มประสิทธิภาพแต่กลับใช้เวลาและทรัพยากรมากขึ้น และค่า max_depth คือความลึกสูงสุดของแต่ละต้นไม้ในป่าสุ่ม ประโยชน์คือสามารถควบคุมความซับซ้อนของต้นไม้ ยิ่งต้นไม้ที่ลึกมากก็จะซับซ้อนมาก แบบจำลองจะเข้าใจข้อมูลเรียนรู้ดี ข้อเสียคือถ้าค่า max_depth มากเกินไปเสี่ยง Overfitting แต่ถ้ายิ่งค่า max_depth น้อยเกินไป อาจจะไม่สามารถเรียนรู้ pattern สำคัญได้ เสี่ยง underfitting และค่า max_features คือจำนวนคุณลักษณะที่ใช้แยก node โดยใช้ OptunaSearchCV หาค่าพารามิเตอร์ที่ดีที่สุด โดยกำหนดช่วงของค่า n_etimators ที่ต้องการทดลอง คือ เลขจำนวนเต็มระหว่าง 100 ถึง 300 ค่า max_depth ที่ต้องการทดลอง คือ เลขจำนวนเต็มระหว่าง 10 ถึง 30 และค่า max_features ที่ทดลอง คือ ‘sqrt’ และ ‘log2’

3.5.6 ต้นไม้ตัดสินใจ (Decision Tree)

การสร้างแบบจำลองต้นไม้ตัดสินใจจะใช้ไลบรารี่ของ Sklearn คือ sklearn.tree ที่ใช้สำหรับการสร้างแบบจำลองต้นไม้ตัดสินใจสำหรับงานประเภทการจำแนก

ในงานวิจัยนี้มีการปรับจูนคือ max_depth คือความลึกของต้นไม้ และใช้ OptunaSearchCV หาค่าพารามิเตอร์ที่ดีที่สุด โดยกำหนดค่า max_depth ที่ต้องการทดลอง คือ เลขจำนวนเต็มระหว่าง 10 ถึง 30

3.5.7 โครงข่ายประสาทเทียมหลายชั้น (Multi-Layer Perceptron)

การสร้างแบบจำลองโครงข่ายประสาทเทียมหลายชั้นเป็นอัลกอริทึมการเรียนรู้แบบเครือข่ายประสาทเทียม (Multi-Layer Perceptron) ที่ใช้งานง่ายและมีประสิทธิภาพสูงในการจำแนก จะใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไลบรารี่ของ Sklearn คือ sklearn.tree ในงานวิจัยนี้มีการปรับจูนคือค่า hidden_layer_size ซึ่งเป็นค่าพารามิเตอร์ที่กำหนดโครงสร้างของ hidden layer ค่า activation คือฟังก์ชันกระตุ้น และค่า alpha คือค่าควบคุม regularization โดยใช้ OptunaSearchCV ค่าพารามิเตอร์ที่ดีที่สุด โดยกำหนดค่า hidden_layer_size ที่ต้องการทดลอง คือ (50,) (100,) (50,100) และ (100,50) ค่า activation ที่ต้องการทดลอง คือ ‘relu’ และ ‘tanh’ และค่า alpha ที่ต้องการทดลอง คือ 0.0001 ถึง 1

3.5.8 เพอร์เซปตรอน (Perceptron)

การสร้างแบบจำลองเพอร์เซปตรอนจะใช้ไลบรารี่ของ Sklearn คือ sklearn.linear_model. Perceptron ในงานวิจัยนี้มีการปรับจูนคือ alpha ซึ่งเป็นค่าพารามิเตอร์ที่ใช้ในการควบคุมอัตราการเรียนรู้หรือค่าปรับแก้ของน้ำหนักในกระบวนการฝึกสอน ค่าที่สูงกว่าจะทำให้การเปลี่ยนแปลงของน้ำหนักใหญ่ขึ้น แต่อาจทำให้การฝึกสอนไม่เสถียร ขณะที่ค่าที่ต่ำกว่าจะทำให้การเปลี่ยนแปลงของน้ำหนักเล็กลงและกระบวนการฝึกสอนเสถียรมากขึ้น และค่า penalty คือวิธีการ regularization โดยใช้ OptunaSearchCV หาค่าพารามิเตอร์ที่ดีที่สุด โดยกำหนดช่วงของค่า alpha ที่ต้องการทดลอง คือ 0.0001 ถึง 0.01 และค่า penalty ที่ต้องการทดลอง คือ None 'l2' 'l1' และ 'elasticnet'

3.5.9 สโตแคสติกการ์เดียนดิเซนท์ (Stochastic Gradient Descent)

การสร้างแบบจำลองสโตแคสติกการ์เดียนดิเซนท์ จะใช้ไลบรารี่ของ Sklearn คือ sklearn.linear_model.SGDClassifier ในงานวิจัยนี้มีการปรับจูนค่า alpha คือค่าพารามิเตอร์การเรียนรู้ที่ใช้ในการป้องกันการ Overfitting การปรับน้ำหนัก ค่า alpha ที่ต่ำหมายถึงการลดการปรับน้ำหนักอย่างมาก ในขณะที่ค่า alpha ที่สูงหมายถึงการลดการปรับน้ำหนักน้อยลง ค่า loss คือฟังก์ชันการสูญเสีย และค่า penalty วิธีการ regularization โดยใช้ OptunaSearchCV หาค่าพารามิเตอร์ที่ดีที่สุด โดยกำหนดช่วงของค่า alpha ที่ต้องการทดลองคือ 0.0001 ถึง 0.01 ฟังก์ชัน loss ที่ต้องการทดลอง คือ ‘hinge’ และ ‘log_loss’ และค่า penalty ที่ต้องการทดลอง คือ ‘l2’ ‘l1’ และ ‘elasticnet’

3.5.10 พาสซีฟ อากัสซีฟ (Passive Aggressive)

การสร้างแบบจำลองพาสซีฟ อากัสซีฟ จะใช้ไลบรารี่ของ Sklearn คือ sklearn.linear_model. PassiveAggressiveClassifier ในงานวิจัยนี้มีการปรับจูนค่า C ซึ่งคือค่าพารามิเตอร์ที่ใช้ในการควบคุมอัตราการอัปเดตของแบบจำลอง ค่า C ต่ำหมายถึงการอัปเดตที่น้อยและระมัดระวัง ในขณะที่ค่า C สูงหมายถึงการอัปเดตที่มากและรวดเร็ว ดังนั้นควรเลือกค่า C ให้เหมาะสมเป็นสิ่งสำคัญต่อประสิทธิภาพของแบบจำลอง และค่า loss คือฟังก์ชันการสูญเสีย โดยจะใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OptunaSearchCV หาค่าพารามิเตอร์ที่ดีที่สุด โดยกำหนดช่วงของค่า C ที่ต้องการทดลองที่ คือ 0.01 ถึง 10 และฟังก์ชัน loss ที่ต้องการทดลอง คือ 'hinge' และ 'squared_hinge'

3.5.11 โครงข่ายประสาทหน่วยความจำระยะสั้นแบบยาว (Long Short-Term Memory)

การสร้างแบบจำลองโครงข่ายประสาทหน่วยความจำระยะสั้นแบบยาว จะใช้ไลบรารี่ของ TensorFlow ในงานวิจัยนี้มีการปรับจูนค่า units ใช้กำหนดความซับซ้อนของแบบจำลอง ยิ่งค่ามากยิ่งจำข้อมูลได้ละเอียดและซับซ้อนขึ้นและค่า learning_rate เป็นค่าควบคุมความเร็วที่โมเดลเรียนรู้ ถ้าค่ามากเกินไปแบบจำลองอาจกระโดดข้ามค่าที่ดีที่สุด ค่าน้อยเกินไป ค่า dropout_rate คืออัตราการสุ่มปิดหน่วยประสาทระหว่างการฝึก เพื่อป้องกัน overfitting แบบจำลองเรียนช้า ค่า epochs ใช้กำหนดจำนวนรอบฝึก และค่า batch_size คือจำนวนข้อมูลในแต่ละรอบที่โมเดลจะนำมาคำนวณและอัปเดตน้ำหนัก โดยถ้ามีขนาด batch ที่เหมาะสมช่วยเพิ่มประสิทธิภาพการฝึกและช่วยให้การอัปเดตน้ำหนักมีความเสถียร แต่ถ้า batch size ใหญ่เกินไป ต้องใช้หน่วยความจำสูงและ batch size เล็กเกินไป ทำให้การฝึกไม่นิ่งและอัปเดตน้ำหนักบ่อยเกินไป โดยจะใช้ OptunaSearchCV หาค่าพารามิเตอร์ที่ดีที่สุด โดยกำหนดช่วงของค่า units ที่ต้องการทดลองที่ คือ 64 และ 128 ค่า learning_rate ที่ต้องการทดลอง คือ จำนวนจริงในช่วง [0.0001, 0.01] ค่า dropout_rate ที่ต้องการทดลองคือจำนวนจริงในช่วง 0.1 ถึง 0.5 ค่า epochs ที่ต้องการทดลอง คือ 5 และ 50 และค่า batch_size ที่ต้องการทดลอง คือ 32 64 และ 128

3.5.12 โครงข่ายประสาทหน่วยความจำระยะสั้นแบบยาวแบบสองทิศทาง (Bidirectional LSTM)

การสร้างแบบจำลองโครงข่ายประสาทหน่วยความจำระยะสั้นแบบยาวแบบสองทิศทาง จะใช้ไลบรารี่ของ TensorFlow ในงานวิจัยนี้มีการปรับจูนค่า units ใช้กำหนดความซับซ้อนของแบบจำลอง ยิ่งค่ามากยิ่งจำข้อมูลได้ละเอียดและซับซ้อนขึ้นและค่า learning_rate เป็นค่าควบคุมความเร็วที่โมเดลเรียนรู้ ถ้าค่ามากเกินไปแบบจำลองอาจกระโดดข้ามค่าที่ดีที่สุด ค่าน้อยเกินไป ค่า dropout_rate คืออัตราการสุ่มปิดหน่วยประสาทระหว่างการฝึก เพื่อป้องกัน overfitting แบบจำลองเรียนช้า ค่า epochs ใช้กำหนดจำนวนรอบฝึก และค่า batch_size คือจำนวนข้อมูลในแต่ละรอบที่โมเดลจะนำมาคำนวณและอัปเดตน้ำหนัก โดยถ้ามีขนาด batch ที่เหมาะสมช่วยเพิ่มประสิทธิภาพการฝึกและช่วยให้การอัปเดตน้ำหนักมีความเสถียร แต่ถ้า batch size ใหญ่เกินไป ต้องใช้หน่วยความจำสูงและ batch size เล็กเกินไป ทำให้การฝึกไม่นิ่งและอัปเดตน้ำหนักบ่อยเกินไป โดยจะใช้ OptunaSearchCV หาค่าพารามิเตอร์ที่ดีที่สุด โดยกำหนดช่วงของค่า units ที่ต้องการทดลองที่ คือ 64 และ 128 ค่า learning_rate ที่ต้องการทดลอง คือ จำนวนจริงในช่วง [0.0001, 0.01] ค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

dropout_rate ที่ต้องการทดลองคือจำนวนจริงในช่วง 0.1 ถึง 0.5 ค่า epochs ที่ต้องการทดลอง คือ 5 และ 50 และค่า batch_size ที่ต้องการทดลอง คือ 32 64 และ 128

3.5.13 โครงข่ายประสาทเทียมแบบคอนโวลูชัน (Convolutional Neural Network)

การสร้างแบบจำลองโครงข่ายประสาทเทียมแบบคอนโวลูชัน จะใช้ไลบรารี่ของ TensorFlow ในงานวิจัยนี้มีการปรับจูนค่า filters ฟิลเตอร์ใช้สำหรับจับ pattern จากข้อมูล ปรับจูนค่า kernel_size เป็นขนาดของ filter ใช้กำหนดจำนวนตำแหน่งของ input ที่แต่ละฟิลเตอร์จะดูพร้อมกัน ปรับจูนค่า learning_rate คืออัตราการเรียนรู้หรือความเร็วในการปรับน้ำหนัก ปรับจูนค่า dropout_rate คืออัตราการสุ่มปิดหน่วยประสาทระหว่างการฝึก เพื่อป้องกัน overfitting แบบจำลองเรียนช้า ปรันจูนค่า epochs ใช้กำหนดจำนวนรอบฝึก และค่า batch_size คือจำนวนข้อมูลในแต่ละรอบที่โมเดลจะนำมาคำนวณและอัปเดตน้ำหนัก โดยถ้ามีขนาด batch ที่เหมาะสมช่วยเพิ่มประสิทธิภาพการฝึกและช่วยให้การอัปเดตน้ำหนักมีความเสถียร แต่ถ้า batch size ใหญ่เกินไปต้องใช้หน่วยความจำสูงและ batch size เล็กเกินไป ทำให้การฝึกไม่นิ่งและอัปเดตน้ำหนักบ่อยเกินไป โดยจะใช้ OptunaSearchCV หาค่าพารามิเตอร์ที่ดีที่สุด โดยกำหนดช่วงของค่า filters ที่ต้องการทดลองที่ คือ 32 64 และ 128 ค่า kernel_size ที่ต้องการทดลอง คือ 3 5 และ 7 ค่า learning_rate ที่ต้องการทดลอง คือ จำนวนจริงในช่วง [0.0001, 0.01] ค่า dropout_rate ที่ต้องการทดลองคือจำนวนจริงในช่วง 0.1 ถึง 0.5 ค่า epochs ที่ต้องการทดลอง คือ 5 และ 50 และค่า batch_size ที่ต้องการทดลอง คือ 32 64 และ 128

3.6 การปรับค่าพารามิเตอร์

การปรับค่าพารามิเตอร์เป็นขั้นตอนสำคัญในการพัฒนาแบบจำลอง เพื่อให้ได้แบบจำลองที่มีประสิทธิภาพที่สุดงานวิจัยนี้มีการปรับค่าพารามิเตอร์โดยใช้ OptunaSearchCV ที่เป็นกระบวนการค้นหาค่าที่ดีที่สุดของพารามิเตอร์ (Best Hyperparameters) โดยใช้ Optuna ร่วมกับ Sklearn โดยการทดสอบหลายๆ ค่าที่กำหนดไว้ล่วงหน้าและประเมินผลด้วยวิธี cross-validation

กำหนดช่วงค่าแต่ละแบบจำลองแสดงได้ดังตารางที่ 3.9 และค่าพารามิเตอร์ที่ดีที่สุดที่ใช้เพิ่มประสิทธิภาพแบบจำลองที่ใช้ตัวตัดคำ PyThaiNLP แสดงได้ดังตารางที่ 3.10 และค่าพารามิเตอร์ที่ดีที่สุดที่ใช้เพิ่มประสิทธิภาพแบบจำลองที่ใช้ตัวตัดคำ DeepCut แสดงได้ดังตารางที่ 3.11

ตารางที่ 3.9 ช่วงค่าพารามิเตอร์ที่ตั้งค่าสำหรับเพิ่มประสิทธิภาพของแบบจำลอง

แบบจำลอง	ค่าพารามิเตอร์	คำอธิบาย	ช่วงค่าที่พิจารณา
Logistic Regression	C	ค่าตรงข้ามของความแรงของการ regularization	FloatDistribution(0.01, 100, log=True)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.9 ช่วงค่าพารามิเตอร์ที่ตั้งค่าสำหรับเพิ่มประสิทธิภาพของแบบจำลอง (ต่อ)

แบบจำลอง	ค่าพารามิเตอร์	คำอธิบาย	ช่วงค่าที่พิจารณา
Logistic Regression	Penalty	วิธีการ regularization	12
	solver	อัลกอริทึมสำหรับการปรับค่าพารามิเตอร์	lbfgs
Naive Bayes	alpha	ค่าการปรับ Laplace smoothing เพื่อหลีกเลี่ยงปัญหาความน่าจะเป็นเป็นศูนย์	FloatDistribution(0.01, 1.0)
KNN	n_neighbors	จำนวนเพื่อนบ้านที่ใช้ในการตัดสิน	IntDistribution(3, 9)
	weights	วิธีถ่วงน้ำหนัก การให้คะแนนเพื่อนบ้าน	‘uniform’, ‘distance’
SVM	C	ควบคุม trade-off ระหว่าง margin กับการจัดประเภทผิดพลาด	FloatDistribution(0.01, 100, log=True)
	Kernel	ชนิดของ kernel function	‘linear’, ‘rbf’
	gamma	ควบคุมระยะของผลกระทบของแต่ละตัวอย่าง	‘scale’, ‘auto’
Random Forest	n_estimators	จำนวน decision trees ใน forest	IntDistribution(100, 300)
	max_depth	ความลึกสูงสุดของแต่ละต้นไม้	IntDistribution(10, 30)
	max_features	จำนวนคุณลักษณะที่ใช้ในการแยก node	‘sqrt, ‘log2’
Decision Tree	Max_depth	จำกัดความลึกของต้นไม้	IntDistribution(10, 30)
Multi-Layer Perceptron	hidden_layer_sizes	โครงสร้างของ hidden layers	(50,), (100,), (50, 100), (100, 50)
	activation	ฟังก์ชันกระตุ้น	‘relu’, ‘tanh’

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.9 ช่วงค่าพารามิเตอร์ที่ตั้งค่าสำหรับเพิ่มประสิทธิภาพของแบบจำลอง (ต่อ)

แบบจำลอง	ค่าพารามิเตอร์	คำอธิบาย	ช่วงค่าที่พิจารณา
Multi-Layer Perceptron	alpha	ค่าควบคุมการ regularization	FloatDistribution(0.0001, 0.01)
Perceptron	alpha	ค่าควบคุมการ regularization	FloatDistribution(0.0001, 0.01)
	penalty	วิธีการ regularization	None, 'l2', 'l1', 'elasticnet'
Stochastic Gradient Descent	alpha	ค่าควบคุมการ regularization	FloatDistribution(0.0001, 0.01)
	loss	ฟังก์ชันการสูญเสีย	'hinge', 'log_loss'
Passive Aggressive	penalty	วิธีการ regularization	'l2', 'l1', 'elasticnet'
	loss	ฟังก์ชันการสูญเสีย	'hinge', 'squared_hinge'
LSTM	units	จำนวนหน่วยความจำ จำนวนเซลล์ใน LSTM layer	64, 128
	learning_rate	อัตราการเรียนรู้ของอัลกอริธึมปรับพารามิเตอร์	1e-4, 1e-2
	dropout_rate	อัตราการสุ่มปิดหน่วยประสาทระหว่างการฝึกเพื่อป้องกัน overfitting	0.1, 0.5
	epochs	จำนวนรอบที่ฝึกแบบจำลอง	5, 50
	batch_size	ขนาดของชุดข้อมูลย่อยในแต่ละรอบการอัปเดตพารามิเตอร์	32, 64, 128
BiLSTM	units	จำนวนเซลล์ในแต่ละทิศทางของ BiLSTM	64, 128
	learning_rate	อัตราการเรียนรู้ของอัลกอริธึมปรับพารามิเตอร์	1e-4, 1e-2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.9 ช่วงค่าพารามิเตอร์ที่ตั้งค่าสำหรับเพิ่มประสิทธิภาพของแบบจำลอง (ต่อ)

แบบจำลอง	ค่าพารามิเตอร์	คำอธิบาย	ช่วงค่าที่พิจารณา
BiLSTM	dropout_rate	อัตราการสุ่มปิดหน่วยประสาทระหว่างการฝึกเพื่อป้องกัน overfitting	0.1, 0.5
	epochs	จำนวนรอบที่ฝึกแบบจำลอง	5, 50
	batch_size	ขนาดของชุดข้อมูลย่อยในแต่ละรอบการอัปเดตพารามิเตอร์	32, 64, 128
CNN	filters	จำนวนฟิลเตอร์ในชั้น Convolution	32, 64, 128
	kernel_size	ขนาดของฟิลเตอร์	3, 5, 7
	learning_rate	อัตราการเรียนรู้ของอัลกอริธึมปรับพารามิเตอร์	1e-4, 1e-2
	dropout_rate	อัตราการสุ่มปิดหน่วยประสาทระหว่างการฝึกเพื่อป้องกัน overfitting	0.1, 0.5
	epochs	จำนวนรอบที่ฝึกแบบจำลอง	5, 50
	batch_size	ขนาดของชุดข้อมูลย่อยในแต่ละรอบการอัปเดตพารามิเตอร์	32, 64, 128

ตารางที่ 3.10 ค่าพารามิเตอร์ที่ดีที่สุดที่ใช้เพิ่มประสิทธิภาพแบบจำลองที่ใช้ตัวตัดคำ PyThaiNLP

ลำดับ	แบบจำลอง	ค่าพารามิเตอร์ที่ดีที่สุด
1	Logistic Regression	C = 5.487019853844503, penalty = 'l2', solver = 'lbfgs'
2	Naïve Bayes	alpha = 0.5981900016310614
3	KNN	n_neighbors = 4, weights = ‘uniform’
4	SVM	C = 2.2204151590602295, kernel = 'rbf', gamma = 'scale'
5	Random Forest	n_estimators = 179, max_depth = 27, max_features = 'sqrt'

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.10 ค่าพารามิเตอร์ที่ดีที่สุดที่ใช้เพิ่มประสิทธิภาพแบบจำลองที่ใช้ตัวตัดคำ PyThaiNLP (ต่อ)

ลำดับ	แบบจำลอง	ค่าพารามิเตอร์ที่ดีที่สุด
6	Decision Tree	max_depth = 30
7	Multi-Layer Perceptron	hidden_layer_sizes = (100,), activation = 'relu', alpha = 0.006076304590109201
8	Perceptron	alpha = 0.0002851505094955989, 'penalty': 'l2'
9	Stochastic Gradient Descent	alpha = 0.00011281916225850828, loss = 'log_loss', penalty = 'l1'
10	Passive Aggressive	C = 0.011290944461967474, loss = 'hinge'
11	LSTM	units = 128, learning_rate = 0.00012948370133259137, dropout_rate = 0.12855298825056743, epochs = 50, batch_size = 32
12	BiLSTM	units = 128, learning_rate = 0.00017301884422386325, dropout_rate = 0.19908126865254286, epochs = 50, batch_size = 32
13	CNN	Filters = 64, kernel_size = 3, learning_rate = 0.00020325945152856084, dropout_rate = 0.11946224922115506, epochs = 5, batch_size = 64

ตารางที่ 3.11 ค่าพารามิเตอร์ที่ดีที่สุดที่ใช้เพิ่มประสิทธิภาพแบบจำลองที่ใช้ตัวตัดคำ DeepCut

ลำดับ	แบบจำลอง	ค่าพารามิเตอร์ที่ดีที่สุด
1	Logistic Regression	C = 1.1658506857780309, penalty = 'l2', solver = 'lbfgs'
2	Naïve Bayes	alpha = 0.7614289078151467
3	KNN	n_neighbors = 4, weights = ‘distance’

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.11 ค่าพารามิเตอร์ที่ดีที่สุดที่ใช้เพิ่มประสิทธิภาพแบบจำลองที่ใช้ตัวตัดคำ DeepCut (ต่อ)

ลำดับ	แบบจำลอง	ค่าพารามิเตอร์ที่ดีที่สุด
4	SVM	C = 1.0528370643398832, kernel = 'rbf', gamma = 'scale'
5	Random Forest	n_estimators = 191, max_depth = 30, max_features = 'sqrt'
6	Decision Tree	max_depth = 30
7	Multi-Layer Perceptron	hidden_layer_sizes = (100,), activation = 'relu', alpha = 0.00391812721018846
8	Perceptron	alpha = 0.00043977073432509593, penalty = None
9	Stochastic Gradient Descent	alpha = 0.0010838329977700823, loss ='hinge', penalty = 'elasticnet'
10	Passive Aggressive	C = 0.0103542765775437788, loss = 'squared_hinge'
11	LSTM	units = 128, learning_rate = 0.0022833953700037565, dropout_rate = 0.12935980043289855, epochs = 5, batch_size = 32
12	BiLSTM	Units = 128, learning_rate = 0.0033656720953600503, dropout_rate = 0.18542877799017332, epochs = 5, batch_size = 128
13	CNN	Filters = 64, kernel_size = 3, learning_rate = 0.0008822714279961056, 'dropout_rate = 0.40426525673778807, epochs = 5, batch_size = 64

3.7 การวัดประสิทธิภาพ

วิธีการประเมินและเปรียบประสิทธิภาพของแต่ละแบบจำลองนั้น การประเมินและเปรียบเทียบประสิทธิภาพของแต่ละแบบจำลองที่ใช้ในการจำแนกประเภทความคิดเห็นในงานวิจัย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฉบับนี้ จะใช้ตัววัดคือตาราง Confusion Matrix ซึ่งเป็นเครื่องมือที่ใช้เปรียบเทียบระหว่างค่าจำแนกจริงและค่าที่ได้จากการทำนายของแบบจำลอง โดยจะนำผลการทำนายมาคำนวณหา ค่าความถูกต้อง (Accuracy) เพื่อแสดงสัดส่วนของการทำนายที่ถูกต้องทั้งหมด ค่าความแม่นยำ (Precision) เพื่อวัดความถูกต้องของผลการทำนายในแต่ละคลาส ค่าความระลึก (Recall) เพื่อวัดความสามารถของแบบจำลองในการค้นหาข้อมูลที่เป็นจริง ค่าความถ่วงดุล (F1-Score) เพื่อประเมินความสมดุลระหว่างค่าความแม่นยำและค่าความครอบคลุม

ผลลัพธ์จากการคำนวณจะถูกนำมาวิเคราะห์และเปรียบเทียบเพื่อระบุว่าแบบจำลองใดให้ผลลัพธ์ที่มีประสิทธิภาพดีที่สุดสำหรับงานวิจัยนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
ผลการวิจัยและอภิปรายผล

ในบทนี้จะกล่าวถึงผลลัพธ์จากการดำเนินงาน ซึ่งประกอบไปด้วย การเก็บรวบรวมข้อมูลที่ใช้ในการวิจัย การวิเคราะห์ความรู้สึกต่อโครงการบ้านเพื่อคนไทยจากความคิดเห็นบนยูทูป และผลการเปรียบเทียบประสิทธิภาพของแบบจำลองด้วยการวัดผลลัพธ์จากความถูกต้อง ค่าความแม่นยำ ค่าความระลึก ค่าความถ่วงดุล และการอภิปรายผล

4.1 ข้อมูลที่ใช้ในการทดลอง

ข้อมูลที่ใช้ในการศึกษาเป็นความคิดเห็นต่อโครงการบ้านเพื่อคนไทยที่ดึงมาจากเว็บไซต์ยูทูป ในช่วงวันที่ 1 ธันวาคม 2567 ถึงวันที่ 30 เมษายน 2568 และทำการเตรียมข้อมูลความคิดเห็นทั้งหมดจำนวน 2,236 รายการ โดยแบ่งออกเป็น 3 ผลเฉลย ประกอบด้วย ความคิดเห็นเชิงบวก จำนวน 555 รายการ ความคิดเห็นเชิงกลาง จำนวน 762 รายการ และความคิดเห็นเชิงลบ จำนวน 919 รายการ ดังรูปที่ 4.1

รูปที่ 4.1 จำนวนข้อความในแต่ละประเภทความรู้สึก

4.2 การเปรียบเทียบการวิเคราะห์คำในแต่ละประเภทความรู้สึกของแต่ละวิธีการตัดคำ

การจำแนกความรู้สึกต่อโครงการบ้านเพื่อคนไทย เริ่มจากการเปรียบเทียบตัวตัดคำภาษาไทยตามชุดข้อมูลของข้อความความคิดเห็นที่ได้ทำการเก็บมาในหัวข้อ 4.1 ที่ได้มาการนำข้อมูลที่ได้มาวิเคราะห์คำในแต่ละประเภทความรู้สึกโดยดูจากความถี่ของคำด้วย Word Cloud เพื่อเป็นการแสดงให้เห็นความถี่ของคำที่ปรากฏในกลุ่มข้อมูล โดยคำที่มีความถี่มากจะมีขนาดใหญ่กว่า และคำที่มีความถี่น้อยจะมีขนาดเล็กกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.1 ผลวิเคราะห์คำในแต่ละประเภทความรู้สึกของการใช้ตัวตัดคำ PyThaiNLP

4.2.1.1 ผลวิเคราะห์คำในแต่ละประเภทความรู้สึกเชิงบวก

จากการวิเคราะห์ข้อความความคิดเห็นที่มีลักษณะเชิงบวกโดยใช้ตัวตัดคำแบบ PyThaiNLP แสดงความคิดเห็นต่อโครงการบ้านเพื่อคนไทย จากรูปที่ 4.2 พบว่าคำที่ปรากฏบ่อย ได้แก่

“ดีมาก” คำนี้จะสะท้อนให้เห็นถึงความพึงพอใจต่อโครงการ เช่น เป็นโครงการที่ดีมาก ดีมาก ดีใจกับคนที่มีสิทธิ์จองได้ ปรากฏในบริบทของการแสดงเจตจำนงในการเข้าร่วมโครงการ เช่น ขอจอง สนใจพร้อมจอง ซึ่งแสดงให้เห็นถึงระดับความต้องการที่อยู่อาศัยภายใต้โครงการอย่างชัดเจน

“อยากได้” คำนี้จะสะท้อนให้เห็นถึงความต้องการในการเป็นเจ้าของที่อยู่อาศัย เช่น โครงการดีมีแต่คนอยากได้

“สุดยอด” มักใช้ในการยกย่องคุณภาพของโครงการ เช่น สุดยอดครับ มีแต่สิ่งดี ๆ แสดงถึงการรับรู้เชิงบวกและความคาดหวังในผลลัพธ์ของโครงการบ้านเพื่อคนไทย

จากคำที่ปรากฏในความคิดเห็นเชิงบวกสะท้อนถึงความชื่นชม ความสนใจ และความอยากได้ต่อโครงการบ้านเพื่อคนไทย ดังรูปที่ 4.2

รูปที่ 4.2 Word Cloud ของข้อความความคิดเห็นเชิงบวก โดยใช้ตัวตัดคำแบบ PyThaiNLP

4.2.1.2 ผลวิเคราะห์คำในแต่ละประเภทความรู้สึกเชิงกลาง

จากการวิเคราะห์ข้อความความคิดเห็นที่มีลักษณะเชิงกลางโดยใช้ตัวตัดคำแบบ PyThaiNLP เกี่ยวกับโครงการบ้านเพื่อคนไทย จากรูปที่ 4.3 พบว่าคำที่ปรากฏบ่อย ได้แก่

“ผ่อน” คำนี้จะสะท้อนให้เห็นถึงข้อพิจารณาเกี่ยวกับเงื่อนไขการผ่อนชำระ เช่น ระยะเวลาหรือจำนวนงวดที่เหมาะสม

“จอง” และ “ซื้อ” ปรากฏในความคิดเห็นเชิงกลางหลายข้อความ แสดงให้เห็นถึงความสนใจและความตื่นตัวของสาธารณชนต่อโครงการ แต่ในขณะเดียวกันก็สะท้อนถึงความไม่แน่ใจและความ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต้องการข้อมูลที่ลึกซึ้งมากขึ้นเกี่ยวกับกลไกและความโปร่งใสของกระบวนการจองที่อยู่อาศัยในโครงการบ้านเพื่อคนไทย โดยมักใช้ในลักษณะของการตั้งข้อสังเกตหรือแสดงความสงสัยเกี่ยวกับพฤติกรรมของผู้ซื้อบ้านในโครงการ อาทิ การตั้งคำถามว่า ผู้ซื้อบ้านในโครงการได้ศึกษารายละเอียดมากน้อยเพียงใดหรือกลุ่มเป้าหมายที่ทำการจองมีลักษณะอย่างไร รวมถึงมีเจตนาซื้อเพื่ออยู่อาศัยจริงหรือเพื่อเก็งกำไรในอนาคต นอกจากนี้ยังสะท้อนความกังวลด้านเทคนิค เช่น ความหนาแน่นของผู้ซื้อในช่วงวันเปิดจอง ซึ่งอาจส่งผลให้ระบบออนไลน์ล่มได้หากไม่มีการเตรียมการรองรับที่เพียงพอและ

จากคำที่ปรากฏในความคิดเห็นเชิงกลางสามารถสะท้อนว่าถึงความสนใจในรายละเอียดหรือเงื่อนไขของโครงการ โดยเฉพาะในเรื่องของทางเลือกในการเป็นเจ้าของ เช่น การจอง การผ่อน การซื้อบ้าน แต่อาจยังไม่มีความคิดเห็นที่ชัดเจนในเชิงบวกหรือลบ ซึ่งถือเป็นข้อมูลเชิงพฤติกรรมที่สามารถนำไปพัฒนานโยบายหรือเงื่อนไขให้สอดคล้องกับความต้องการของประชาชนมากขึ้น ดังรูปที่ 4.3

รูปที่ 4.3 Word Cloud ของข้อความความคิดเห็นเชิงกลาง โดยใช้ตัวตัดคำแบบ PyThaiNLP

4.2.1.3 ผลวิเคราะห์คำในแต่ละประเภทความรู้สึกเชิงลบ

จากการวิเคราะห์ข้อความความคิดเห็นที่มีลักษณะเชิงลบโดยใช้ตัวตัดคำแบบ PyThaiNLP เกี่ยวกับโครงการบ้านเพื่อคนไทย จากรูปที่ 4.4 พบว่าคำที่ปรากฏบ่อย ได้แก่

"เอื้ออาทร" มักถูกใช้ในบริบทของการเปรียบเทียบกับโครงการในอดีต เช่น โครงการบ้านเอื้ออาทร ที่ประชาชนเคยมีประสบการณ์หรือมีความรับรู้ในด้านลบมาก่อน อาทิ ในแง่ของคุณภาพของที่อยู่อาศัย ความโปร่งใสในกระบวนการจัดสรรสิทธิ หรือปัญหาสังคมที่เกิดขึ้นภายหลังการเข้าอยู่อาศัย จึงอาจกล่าวได้ว่าการกล่าวถึง "บ้านเอื้ออาทร" ในที่นี้ สะท้อนถึงความกังวล ความไม่มั่นใจ และการตั้งข้อสังเกตเกี่ยวกับความยั่งยืนของโครงการใหม่ในลักษณะเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

“เช่า” คำนี้จะสะท้อนให้เห็นถึงความไม่เข้าใจในเงื่อนไขของโครงการ เช่น ความคลุมเครือระหว่างการซื้อกับการถือครองในลักษณะการเช่า หรือการเช่าระยะยาว 99 ปี

“ผ่อน” คำนี้จะสะท้อนให้เห็นถึงความไม่มั่นใจในเงื่อนไขการผ่อนชำระ ตัวอย่างประโยค เช่น ผ่อนนานเกินไป ผ่อนแพงเกินไป

จากคำที่ปรากฏในความคิดเห็นเชิงลบสะท้อนถึงความรู้สึกไม่มั่นคง ความกังวลด้านความโปร่งใสของโครงการ และความไม่แน่ใจในผลสัมฤทธิ์ของนโยบายเมื่อเทียบกับประสบการณ์ในอดีต ซึ่งเป็นประเด็นสำคัญที่ภาครัฐควรนำไปพิจารณาในการสื่อสารข้อมูล และออกแบบนโยบายที่ตอบโจทย์ประชาชนอย่างแท้จริง ดังรูปที่ 4.4

รูปที่ 4.4 Word Cloud ของข้อความความคิดเห็นเชิงลบ โดยใช้ตัวตัดคำแบบ PyThaiNLP

4.2.2 ผลวิเคราะห์คำในแต่ละประเภทความรู้สึกของการใช้ตัวตัดคำ DeepCut

4.2.2.1 ผลวิเคราะห์คำในแต่ละประเภทความรู้สึกเชิงบวก (Positive)

จากการวิเคราะห์ข้อความความคิดเห็นที่มีลักษณะเชิงบวกโดยใช้ตัวตัดคำแบบ DeepCut เกี่ยวกับโครงการบ้านเพื่อคนไทย จากรูปที่ 4.5 พบว่าคำที่ปรากฏบ่อย ได้แก่

“จอง” แสดงถึงความของต้องการและความสนใจอย่างจริงจังในโครงการ มีทัศนคติเชิงบวกถึงขั้นจองหรือแสดงความยินดีกับผู้ที่จองได้ มักปรากฏในประโยคที่แสดงความสนใจเข้าร่วม เช่น จองด้วย สนใจจะจองทางไหนคะ ยินดีกับคนที่จองได้

“ซื้อ” แสดงถึงความต้องการในการเป็นเจ้าของที่อยู่อาศัย เช่น อยากซื้อสักห้อง น่าซื้อมาก

“ดี” แสดงถึงการยืนยันคุณภาพ ทั้งในด้านราคา ทำเล เงื่อนไข และความเหมาะสมกับกลุ่มเป้าหมาย พบบ่อยในรูปแบบคำว่า ดีมาก ดีครับ ดีและเหมาะมา

“ยอด” แสดงถึงการชื่นชมระดับสูงสุด อารมณ์ตื่นเต้น ภาคภูมิใจ หรือประทับใจอย่างมาก มักมาคู่กับ “สุด” เช่น สุดยอดมาก ผลงานนี้สุดยอด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

“งาน” คำนี้อาจจะสื่อถึงบริบทต่อโครงการหรืองานที่ดำเนินการโดยรัฐบาล พบในรูปแบบประโยค เช่น งานนี้สุดยอด มั่นใจในเทคนิคในการทำงานมาโดยตลอด คนรุ่นนี้สบายมีหน่วยงานภาครัฐมาช่วย

จากคำที่ปรากฏในความคิดเห็นเชิงบวกที่สะท้อนถึงความสนใจ ความเชื่อมั่น การตัดสินใจ และความพึงพอใจ ต่อโครงการในหลายมิติ ทั้งในเชิงคุณภาพ ความคุ้มค่า และผลลัพธ์ที่ได้รับจริง ดังรูปที่ 4.5

1

รูปที่ 4.5 Word Cloud ของข้อความความคิดเห็นเชิงบวก โดยใช้ตัวตัดคำแบบ DeepCut

4.2.2.2 ผลวิเคราะห์คำในแต่ละประเภทความรู้สึกเชิงกลาง (Neutral)

จากการวิเคราะห์ข้อความความคิดเห็นที่มีลักษณะเชิงกลางโดยใช้ตัวตัดคำแบบ DeepCut เกี่ยวกับโครงการบ้านเพื่อคนไทย จากรูปที่ 4.6 พบว่าคำที่ปรากฏบ่อย ได้แก่

“ผ่อน” คำนี้สะท้อนให้เห็นถึงความใส่ใจในรายละเอียดทางการเงิน โดยเฉพาะเงื่อนไขการผ่อนชำระ ความยืดหยุ่น และความมั่นคงในระยะยาวของโครงการ ซึ่งเป็นสิ่งสำคัญสำหรับกลุ่มที่มีข้อจำกัดทางการเงินหรือประวัติเครดิตไม่ดี พบบ่อยในประโยคเชิงกลาง ได้แก่ เริ่มผ่อนช่วงไหน ติดเครดิตบูโรผ่อนได้ไหม ผ่อนไป 5 ปี แล้วไม่เอาแล้วขายต่อได้มั้ย จากประโยคแสดงให้เห็นถึงลักษณะความคิดเห็นเชิงกลางในรูปแบบการสอบถามข้อมูลโดยไม่แสดงอารมณ์ในเชิงบวกหรือลบ

“จอง” คำนี้สะท้อนถึงแสดงถึงความต้องการเข้าร่วม แต่ก็มีความกังวลเรื่องระบบและความชัดเจน เช่น จองที่ไหน ลงทะเบียนจองที่ไหน ถ้าจองออนไลน์ล่มแน่ จากประโยคแสดงถึงลักษณะของการตั้งข้อสังเกตในลักษณะตรรกะเชิงเหตุผล โดยไม่ปรากฏการแสดงอารมณ์ในลักษณะชื่นชมหรือวิพากษ์วิจารณ์อย่างชัดเจน

“ซื้อ” คำนี้สะท้อนถึงประเด็นเรื่องกรรมสิทธิ์ การเป็นเจ้าของ และผลประโยชน์ในระยะยาว พบบ่อยในเชิงการตั้งคำถามเพื่อความเข้าใจ เช่น ซื้อหรือเช่า ซื้อแล้วเป็นเจ้าของมั้ย จากประโยค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แสดงถึงลักษณะของการสอบถามเพื่อความเข้าใจ โดยไม่แสดงอารมณ์ในลักษณะสนับสนุนหรือคัดค้าน

"ขาย" คำนี้สะท้อนถึงประเด็นในเชิงนโยบายและความคาดหวังเกี่ยวกับการบริหารจัดการทรัพย์สินของรัฐ เช่น การนำของเก่ามาพัฒนา การเข้าถึงในราคาที่ย่อมเยา หรือความเป็นห่วงเกี่ยวกับการขายให้ต่างชาติ จากประโยคแสดงถึงลักษณะของการเสนอแนะในเชิงเหตุผลหรือสะท้อนข้อสังเกต โดยไม่ได้มีลักษณะโจมตีหรือตำหนิชัดเจน

จากคำที่ปรากฏในความคิดเห็นเชิงกลางสะท้อนถึงความสนใจต่อประเด็นที่เกี่ยวข้องกับโครงการที่อยู่อาศัย โดยเฉพาะในมิติของการเข้าถึง การผ่อนชำระ สิทธิในการเป็นเจ้าของ และการจัดการทรัพย์สินของรัฐ แม้จะไม่แสดงความรู้สึกเชิงอารมณ์ที่ชัดเจน แต่ความคิดเห็นเหล่านี้สะท้อนถึง ความต้องการข้อมูลที่ชัดเจน ระบบที่โปร่งใส และ โอกาสที่เท่าเทียม ซึ่งเป็นพื้นฐานที่สำคัญต่อการสร้างความเชื่อมั่นและการมีส่วนร่วมของประชาชนในโครงการภาครัฐ ดังรูปที่ 4.6

รูปที่ 4.6 Word Cloud ของข้อความความคิดเห็นเชิงกลาง โดยใช้ตัวตัดคำแบบ DeepCut

4.2.2.3 ผลวิเคราะห์คำในแต่ละประเภทความรู้สึกเชิงลบ (Negative)

จากการวิเคราะห์ข้อความความคิดเห็นที่มีลักษณะเชิงลบโดยใช้ตัวตัดคำแบบ DeepCut เกี่ยวกับโครงการบ้านเพื่อคนไทย จากรูปที่ 4.7 พบว่าคำที่ปรากฏบ่อย ได้แก่

"เอื้ออาทร" คำนี้สะท้อนภาพลักษณ์เชิงลบที่ผู้คนมีต่อโครงการในอดีต โดยแสดงความกังวลว่าจะเกิดปัญหาเดิมซ้ำรอย มักจะถูกใช้ในลักษณะเปรียบเปรยหรืออุปมาเพื่อสื่อถึง ความล้มเหลวของโครงการในอดีต โดยเน้นความทรงจำเชิงลบ เช่น บ้านร้าง การทุจริต การบริหารผิดพลาด และการไม่ได้ประโยชน์จริงจากโครงการ ความคิดเห็นเหล่านี้ชี้ให้เห็นถึงความไม่ไว้วางใจต่อภาครัฐและ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นโยบายที่เกี่ยวข้อง ตัวอย่างประโยค เช่น จะเหมือนเอื้ออาทรไหม ซากบ้านเอื้ออาทรร้างเต็มไปหมด โกงแบบบ้านเอื้ออาทร

“ผ่อน” คำนี้สะท้อนปัญหาความโปร่งใสในการสื่อสารของรัฐ โดยเฉพาะการให้ข้อมูลไม่ครบถ้วนเกี่ยวกับโฉนด การเป็นเจ้าของ และผลประโยชน์ที่แท้จริงหลังการผ่อนจ่าย ทำให้ประชาชนรู้สึกไม่มั่นใจ และมองว่า “ผ่อน” อาจเป็นเพียงภาพลวงตาทางการตลาดและถูกใช้ในเชิงตั้งคำถามและเสียดสีระบบของโครงการ โดยแสดงถึงความไม่พอใจต่อ ความไม่ชัดเจนของสิทธิในกรรมสิทธิ์ ผู้แสดงความคิดเห็นจำนวนมากรู้สึกว่าตนอาจต้องรับภาระการผ่อนจ่ายในระยะยาว แต่สุดท้ายก็ไม่ได้ครอบครองบ้านจริง ตัวอย่างประโยค เช่น ผ่อนแต่ไม่มีโฉนด เหมือนเช่าอยู่ ผ่อนจบ ไม่ใช่ของเราอีกนะ ผ่อนแล้วไม่ได้เป็นเจ้าของ ผ่อนทำไม

จากคำที่ปรากฏในความคิดเห็นเชิงลบสะท้อนถึงความกังวลต่อความโปร่งใสในการดำเนินงานของภาครัฐและประสบการณ์เชิงลบจากโครงการในอดีต เช่น โครงการบ้านเอื้ออาทร ที่กลายเป็นภาพจำด้านลบเกี่ยวกับการทุจริตและความล้มเหลวในการบริหาร ขณะเดียวกัน ประเด็นเรื่อง “การผ่อนชำระ” โดยไม่มีความชัดเจนในกรรมสิทธิ์ เป็นประเด็นที่ถูกวิพากษ์อย่างหนักในเชิงความรู้สึกว่าถูกทำให้เข้าใจผิด และไม่คุ้มค่ากับภาระทางการเงินที่ประชาชนต้องแบกรับในระยะยาว ดังรูปที่ 4.7

รูปที่ 4.7 Word Cloud ของข้อความความคิดเห็นเชิงลบ โดยใช้ตัวตัดคำแบบ DeepCut

4.3 ผลการวิเคราะห์คำในแต่ละประเภทความรู้สึก

จากผลการวิเคราะห์คำในแต่ละประเภทความรู้สึกของแต่ละวิธีการตัดคำในหัวข้อ 4.2 พบความสำคัญของคำที่ปรากฏในแต่ละความรู้สึก ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.1 ผลวิเคราะห์คำในแต่ละประเภทความรู้สึกเชิงบวก

จากการวิเคราะห์ข้อความความคิดเห็นที่มีลักษณะเชิงบวกโดยการใช้ตัวตัดคำแบบ PyThaiNLP ดังรูปที่ 4.2 และการใช้ตัวตัดคำแบบ DeepCut ดังรูปที่ 4.5 พบคำปรากฏบ่อยและโด่นเด่นมักเกี่ยวกับคำชื่นชมต่อโครงการบ้านเพื่อคนไทย เช่น “ดีมาก” “สุดยอด” โดยส่วนใหญ่จะเป็นการชื่นชมเกี่ยวกับทำเลที่ดีของโครงการ มีการเดินทางโทรคมนาคมที่สะดวก ราคาที่ถูกในทำเลที่ที่ดินราคาแพง การชื่นชมและสนับสนุนต่อโครงการของภาครัฐ มีการแสดงถึงการอยากเข้าร่วมโครงการ เช่น “จอง” “อยากได้”

4.3.2 ผลวิเคราะห์คำในแต่ละประเภทความรู้สึกเชิงกลาง

จากการวิเคราะห์ข้อความความคิดเห็นที่มีลักษณะเชิงกลางโดยการใช้ตัวตัดคำแบบ PyThaiNLP ดังรูปที่ 4.3 และการใช้ตัวตัดคำแบบ DeepCut ดังรูปที่ 4.6 พบคำปรากฏบ่อยมักเกี่ยวกับเชิงพฤติกรรม เช่น “จอง” “ผ่อน” “ขาย” และ “ซื้อ” ซึ่งจากผลสำรวจจากข้อมูลเชิงกลาง คำปรากฏบ่อยจะอยู่ในรูปแบบประโยคคำถาม ข้อสงสัย และการรอข้อมูลเพิ่มเติมโดยส่วนใหญ่ เช่น มีขั้นตอนการจองยังไง มีรูปแบบการผ่อนชำระยังไง เงื่อนไขในการซื้อ รูปแบบการขายต่อในอนาคต เนื่องจากทางหน่วยงานภาครัฐอาจให้ข้อมูลที่กำกวม ไม่ชัดเจน ทำให้ประชาชนจำนวนหนึ่ง เกิดคำถามและข้อสงสัยต่อโครงการ และต้องการข้อมูลที่ชัดเจนเพิ่มเติม

4.3.3 ผลวิเคราะห์คำในแต่ละประเภทความรู้สึกเชิงลบ

จากการวิเคราะห์ข้อความความคิดเห็นที่มีลักษณะเชิงลบโดยการใช้ตัวตัดคำแบบ PyThaiNLP ดังรูปที่ 4.4 และการใช้ตัวตัดคำแบบ DeepCut ดังรูปที่ 4.7 พบว่าคำที่โดดเด่นจะเกี่ยวกับภาพลักษณ์เชิงลบที่ผู้คนมีต่อโครงการในอดีต นั้นก็คือคำว่า “เอื้ออาทร” ซึ่งจากผลสำรวจจากข้อมูลเชิงลบ คำนี้มักถูกใช้ในบริบทของการเปรียบเทียบกับโครงการในอดีต โดยแสดงความกังวลว่าจะเกิดปัญหาเดิมซ้ำรอย มักจะถูกใช้ในลักษณะเปรียบเปรยถึงความล้มเหลวของโครงการในอดีต โดยเน้นความทรงจำเชิงลบ เช่น เช่น บ้านร้าง การทุจริต การบริหารผิดพลาด และการไม่ได้ประโยชน์จริงจากโครงการ ประสบการณ์หรือมีความรับรู้ในด้านลบมาก่อน และขณะเดียวกันประเด็นเรื่องการผ่อนชำระ โดยไม่มีความชัดเจนในกรรมสิทธิ์ เป็นประเด็นที่ถูกวิพากษ์อย่างหนักในเชิงความรู้สึกว่าถูกทำให้เข้าใจผิด และไม่คุ้มค่ากับภาระทางการเงินที่ประชาชนต้องแบกรับในระยะยาว โดยพบคำว่า “ผ่อน” มีความถี่สูงสุดอันดับ 2 จากผลสำรวจจากข้อมูลเชิงลบ และคำนี้มักถูกใช้ในบริบทของการสงสัยสิทธิในการอยู่อาศัยที่กำหนดระยะเวลาการอยู่อาศัย การเป็นเจ้าของ และผลประโยชน์ที่แท้จริงหลังการผ่อนจ่าย ทำให้ประชาชนรู้สึกไม่มั่นใจ ไม่พอใจต่อ ความไม่ชัดเจนของสิทธิในกรรมสิทธิ์ ผู้แสดงความคิดเห็นจำนวนมากรู้สึกว่าตนอาจต้องรับภาระการผ่อนจ่ายในระยะยาว แต่สุดท้ายก็ไม่ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ครอบครองบ้านจริง ตัวอย่างประโยค เช่น ผ่อนแต่ไม่มีโฉนด เหมือนเช่าอยู่ ผ่อนจบไม่ได้เป็นเจ้าของ จากข้อมูลสะท้อนว่าประชาชนมีทั้งความคาดหวัง ความลังเล และความไม่ไว้วางใจต่อโครงการบ้านเพื่อคนไทย

4.4 การเปรียบเทียบผลการจำแนกความคิดเห็นโครงการบ้านเพื่อคนไทยตามตัวตัดคำ

การจำแนกความคิดเห็นต่อโครงการบ้านเพื่อคนไทย เริ่มจากการเปรียบเทียบตัวตัดคำภาษาไทยตามชุดข้อมูลของข้อความความคิดเห็นที่ได้ทำการเก็บมาในหัวข้อ 4.1 ที่ได้มาการนำมาใช้งานวิจัยโดยผลส่วนนี้จะเป็นคำตอบที่สอดคล้องกับวัตถุประสงค์ที่ 1 ของงานวิจัยนี้

ผลการตัวตัดคำจะออกมาตามแบบจำลองที่ได้นำมาใช้ในงานวิจัย 13 แบบ ได้แก่ การถดถอยเชิงเส้นโลจิสติค (Logistic Regression) นาอีฟเบย์ (Naïve Bayes) การหาค่าด้วยเพื่อนบ้านใกล้ที่สุด k จุด (K-Nearest Neighbors) ซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machine) ป่าสุ่ม (Random Forest) ต้นไม้ตัดสินใจ (Decision Tree) ตัวจำแนกแบบโครงข่ายประสาทเทียมหลายชั้น (Multi-Layer Perceptron) เพอร์เซปตรอน (Perceptron) สโตแคสติกการ์เดียนดิเซนท์ (Stochastic Gradient Descent) พาสซีฟ อากัสซีฟ (Passive Aggressive) โครงข่ายประสาทหน่วยความจำระยะสั้นแบบยาว(Long Short-Term Memory) โครงข่ายประสาทหน่วยความจำระยะสั้นแบบยาวแบบสองทิศทาง (Bidirectional LSTM) และโครงข่ายประสาทเทียมแบบคอนโวลูชัน (Convolutional Neural Network) ได้แสดงค่าการวัดประสิทธิภาพค่าความถูกต้อง (Accuracy) ค่าความแม่นยำ (Precision) ค่าความระลึก (Recall) และค่าความถ่วงดุล (F1-Score) โดยแยกเป็นตัวตัดคำ PyThaiNLP ได้ผลตามตารางที่ 4.1 และตัวตัดคำ DeepCut ได้ผลตามตารางที่ 4.2

ตารางที่ 4.1 ผลการประเมินประสิทธิภาพแบบจำลองจากการตัวตัดคำ PyThaiNLP

ลำดับ	แบบจำลอง	Accuracy	Precision	Recall	F1-Score
1	Logistic Regression	57.38	58.49	57.38	56.32
2	Naïve Bayes	55.89	60.02	55.89	53.04
3	KNN	42.77	49.31	42.77	41.85
4	SVM	**59.25**	**59.58**	**59.25**	**59.47**
5	Random Forest	56.48	55.90	56.48	55.91
6	Decision Tree	48.29	47.84	48.29	47.87
7	Multi-Layer Perceptron	56.48	56.49	56.48	56.49
8	Perceptron	59.93	56.79	56.93	56.85

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.1 ผลการประเมินประสิทธิภาพแบบจำลองจากการตัวตัดคำ PyThaiNLP (ต่อ)

ลำดับ	แบบจำลอง	Accuracy	Precision	Recall	F1-Score
9	Stochastic Gradient Descent	56.18	55.86	56.18	55.93
10	Passive Aggressive	50.82	51.90	50.82	51.08
11	LSTM	54.40	60.93	54.40	49.91
12	BiLSTM	56.33	59.07	56.33	54.24
13	CNN	56.03	59.08	56.04	53.34

ตารางที่ 4.2 ผลการประเมินประสิทธิภาพแบบจำลองจากการตัวตัดคำ DeepCut

ลำดับ	แบบจำลอง	Accuracy	Precision	Recall	F1-Score
1	Logistic Regression	57.97	57.48	57.97	57.48
2	Naïve Bayes	53.95	56.35	53.95	51.10
3	KNN	42.32	49.27	42.32	42.22
4	SVM	**59.17**	**59.43**	**59.17**	**58.36**
5	Random Forest	54.99	54.53	54.99	53.90
6	Decision Tree	49.93	50.40	49.93	49.77
7	Multi-Layer Perceptron	53.20	53.30	53.20	53.11
8	Perceptron	53.35	53.87	53.35	52.97
9	Stochastic Gradient Descent	56.33	56.45	56.33	56.15
10	Passive Aggressive	52.61	52.66	52.61	52.35
11	LSTM	51.71	55.00	51.71	46.66
12	BiLSTM	55.14	56.74	55.14	52.86
13	CNN	53.95	56.70	53.95	50.56

ผลการวัดประสิทธิภาพแต่ละแบบจำลองทั้ง 13 แบบ โดยแบบจำลองที่ใช้การตัวตัดคำแบบ PyThaiNLP มีค่าความถูกต้องมากกว่าแบบจำลองที่ใช้การตัวตัดคำแบบ DeepCut อยู่ 9 แบบจำลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คือ นาอีฟเบย์ (Naïve Bayes) การหาค่าด้วยเพื่อนบ้านใกล้ที่สุด k จุด (K-Nearest Neighbors) ซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machine) ป่าสุ่ม (Random Forest) ตัวจำแนกแบบโครงข่ายประสาทเทียมหลายชั้น (Multi-Layer Perceptron) เพอร์เซปตรอน (Perceptron) โครงข่ายประสาทหน่วยความจำระยะสั้นแบบยาว(Long Short-Term Memory) โครงข่ายประสาทหน่วยความจำระยะสั้นแบบยาวแบบสองทิศทาง (Bidirectional LSTM) และโครงข่ายประสาทเทียมแบบคอนโวลูชัน (Convolutional Neural Network) และแบบจำลองที่ใช้การตัวตัดคำ DeepCut มีค่าความถูกต้องมากกว่าแบบจำลองที่ใช้การตัวตัดคำ PyThaiNLP อยู่ 4 แบบจำลอง คือ การถดถอยเชิงเส้นโลจิสติค (Logistic Regression) ต้นไม้ตัดสินใจ (Decision Tree) สโตแคสติกการ์เดียนดิเซนท์ (Stochastic Gradient Descent) พาสซีฟ อากัสซีฟ (Passive Aggressive)

สรุปผลการวัดประสิทธิภาพตัวตัดคำจากแบบจำลองทั้ง 13 แบบ ได้ว่า การตัวตัดคำแบบ PyThaiNLP มีประสิทธิภาพดีกว่าการตัวตัดคำแบบ DeepCut

4.5 ผลการจำแนกความคิดเห็นต่อโครงการบ้านเพื่อคนไทยจากแบบจำลอง

ผลการจำแนกความคิดเห็นในส่วนนี้จะเป็นคำตอบที่สอดคล้องกับวัตถุประสงค์ที่ 2 ของงานวิจัยนี้ โดยแบบจำลองที่ใช้ในการจำแนกความคิดเห็นต่อโครงการบ้านเพื่อคนไทย ได้แก่ การถดถอยเชิงเส้นโลจิสติค (Logistic Regression) นาอีฟเบย์ (Naïve Bayes) การหาค่าด้วยเพื่อนบ้านใกล้ที่สุด k จุด (K-Nearest Neighbors) ซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machine) ป่าสุ่ม (Random Forest) ต้นไม้ตัดสินใจ (Decision Tree) ตัวจำแนกแบบโครงข่ายประสาทเทียมหลายชั้น (Multi-Layer Perceptron) เพอร์เซปตรอน (Perceptron) สโตแคสติกการ์เดียนดิเซนท์ (Stochastic Gradient Descent) พาสซีฟ อากัสซีฟ (Passive Aggressive) โครงข่ายประสาทหน่วยความจำระยะสั้นแบบยาว(Long Short-Term Memory) โครงข่ายประสาทหน่วยความจำระยะสั้นแบบยาวแบบสองทิศทาง (Bidirectional LSTM) และโครงข่ายประสาทเทียมแบบคอนโวลูชัน (Convolutional Neural Network) ซึ่งหลังจากข้อมูลผ่านกระบวนการเตรียมข้อมูลในหัวข้อ 3.2 แล้วนำข้อมูลจะมีการแบ่งชุดข้อมูลเป็น ชุดข้อมูลเรียนรู้ 70% ชุดข้อมูลทดสอบ 30% แบ่งโดยการสุ่มหลังจากนั้นจะถูกนำไปสร้างแบบจำลอง

ข้อมูลสำหรับการฝึกสอนนั้นจะถูกนำไปใช้สอนแบบจำลองต่าง ๆ เพื่อให้แบบจำลองเกิดการเรียนรู้ เพื่อให้สามารถจำแนกความคิดเห็นเชิงบวก ความคิดเห็นเชิงกลาง ความคิดเห็นเชิงลบ ซึ่งแบบจำลองที่ผ่านการเรียนรู้นั้นจะถูกนำมาทดสอบด้วยข้อมูลทดสอบ 30% โดยการวัดผลสำหรับแบบจำลองจะให้ชุดข้อมูลทดสอบซึ่งใช้ตัววัด 4 แบบ ได้แก่ ความถูกต้อง (Accuracy) ค่าความแม่นยำ (Precision) ค่าความระลึก (Recall) และค่าความถ่วงดุล (F1-Score)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5.1 ผลจากแบบจำลองการถดถอยเชิงโลจิสติค (Logistic Regression)

4.5.1.1 ผลจากแบบจำลองการถดถอยเชิงโลจิสติค (Logistic Regression) โดยใช้ตัวตัดคำ PyThaiNLP

จากผลในการหาพารามิเตอร์ที่ดีที่สุดของไลบรารี่ OptunaSearchCV ค่าพารามิเตอร์ที่ใช้ในการเพิ่มประสิทธิภาพแบบจำลองการถดถอยเชิงโลจิสติค คือ C = 5.487019853844503, penalty = ‘l2’ และ solver = ‘lbfgs’ โดยหลังจากปรับค่าพารามิเตอร์แล้วได้ผลการทดลอบประสิทธิภาพของการถดถอยเชิงโลจิสติค ด้วยข้อมูลทดสอบสามารถแสดงได้ดังตารางที่ 4.3 พบว่าจากความคิดเห็นต่อโครงการบ้านเพื่อคนไทยสำหรับทดสอบ 671 ข้อความ ดังนี้

จำนวนข้อความที่เป็น Positive จริงและทำนายว่าเป็น Positive	=	187	ครั้ง
จำนวนข้อความที่เป็น Positive จริงและทำนายว่าเป็นอย่างอื่น	=	113	ครั้ง
จำนวนข้อความที่เป็น Negative จริงและทำนายว่าเป็น Negative	=	108	ครั้ง
จำนวนข้อความที่เป็น Negative จริงและทำนายว่าเป็นอย่างอื่น	=	98	ครั้ง
จำนวนข้อความที่เป็น Neutral จริงและทำนายว่าเป็น Neutral	=	111	ครั้ง
จำนวนข้อความที่เป็น Neutral จริงและทำนายว่าเป็นอย่างอื่น	=	59	ครั้ง

ตารางที่ 4.3 เมทริกซ์ความสับสนแบบจำลองการถดถอยเชิงโลจิสติคด้วยข้อมูลทดสอบ โดยใช้ตัวตัดคำ PyThaiNLP

ผลทำนาย / ผลจริง	Positive	Negative	Neutral	รวม
Positive	182	67	24	273
Negative	83	108	35	226
Neutral	30	31	111	172
รวม	295	206	170	671

จากตารางที่ 4.3 สามารถคำนวณตัววัดประสิทธิภาพทั้ง 4 ตัววัด ดังนี้

1. ค่าความถูกต้อง (Accuracy) = 59.76%
2. ค่าความแม่นยำ (Precision) = 59.50%
3. ค่าความระลึก (Recall) = 59.76%
4. ค่าความถ่วงดุล (F1-Score) = 59.55%

4.5.1.2 ผลจากแบบจำลองการถดถอยเชิงโลจิสติค (Logistic Regression) โดยใช้ตัวตัดคำ DeepCut

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากผลในการหาพารามิเตอร์ที่ดีที่สุดของไลบรารี่ OptunaSearchCV ค่าพารามิเตอร์ที่ใช้ในการเพิ่มประสิทธิภาพแบบจำลองการถดถอยเชิงโลจิสติค คือ C = 1.1658506857780309 , penalty = 'l2' และ solver = 'lbfgs' โดยหลังจากปรับค่าพารามิเตอร์แล้วได้ผลการทดลอบประสิทธิภาพของการถดถอยเชิงโลจิสติค ด้วยข้อมูลทดสอบสามารถแสดงได้ดังตารางที่ 4.4 พบว่าจากความคิดเห็นต่อโครงการบ้านเพื่อคนไทยสำหรับทดสอบ 671 ข้อความ ดังนี้

จำนวนข้อความที่เป็น Positive จริงและทำนายว่าเป็น Positive = 190 ครั้ง
จำนวนข้อความที่เป็น Positive จริงและทำนายว่าเป็นอย่างอื่น = 119 ครั้ง
จำนวนข้อความที่เป็น Negative จริงและทำนายว่าเป็น Negative = 103 ครั้ง
จำนวนข้อความที่เป็น Negative จริงและทำนายว่าเป็นอย่างอื่น = 96 ครั้ง
จำนวนข้อความที่เป็น Neutral จริงและทำนายว่าเป็น Neutral = 91 ครั้ง
จำนวนข้อความที่เป็น Neutral จริงและทำนายว่าเป็นอย่างอื่น = 72 ครั้ง

ตารางที่ 4.4 เมทริกซ์ความสับสนแบบจำลองการถดถอยเชิงโลจิสติคด้วยข้อมูลทดสอบ โดยใช้ตัวตัดคำ DeepCut

ผลทำนาย / ผลจริง	Positive	Negative	Neutral	รวม
Positive	190	58	25	273
Negative	88	103	47	238
Neutral	31	38	91	160
รวม	309	199	163	671

จากตารางที่ 4.4 สามารถคำนวณตัววัดประสิทธิภาพทั้ง 4 ตัววัด ดังนี้

1. ค่าความถูกต้อง (Accuracy) = 57.23%
2. ค่าความแม่นยำ (Precision) = 56.69%
3. ค่าความระลึก (Recall) = 57.23%
4. ค่าความถ่วงดุล (F1-Score) = 56.72%

4.5.2 ผลจากแบบจำลองนาอีฟเบย์ (Naïve Bayes)

4.5.2.1 ผลจากแบบจำลองนาอีฟเบย์ (Naïve Bayes) โดยใช้ตัวตัดคำ PyThaiNLP

จากผลในการหาพารามิเตอร์ที่ดีที่สุดของไลบรารี่ OptunaSearchCV ค่าพารามิเตอร์ที่ใช้ในการเพิ่มประสิทธิภาพแบบจำลองนาอีฟเบย์ คือ alpha = 0.5981900016310614 โดยหลังจากปรับค่าพารามิเตอร์แล้วได้ผลการทดลอบประสิทธิภาพของนาอีฟเบย์ ด้วยข้อมูลทดสอบสามารถแสดงได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังตารางที่ 4.5 พบว่าจากความคิดเห็นต่อโครงการบ้านเพื่อคนไทยสำหรับทดสอบ 671 ข้อความ ดังนี้

จำนวนข้อความที่เป็น Positive จริงและทำนายว่าเป็น Positive = 228 ครั้ง
จำนวนข้อความที่เป็น Positive จริงและทำนายว่าเป็นอย่างอื่น = 202 ครั้ง
จำนวนข้อความที่เป็น Negative จริงและทำนายว่าเป็น Negative = 87 ครั้ง
จำนวนข้อความที่เป็น Negative จริงและทำนายว่าเป็นอย่างอื่น = 67 ครั้ง
จำนวนข้อความที่เป็น Neutral จริงและทำนายว่าเป็น Neutral = 60 ครั้ง
จำนวนข้อความที่เป็น Neutral จริงและทำนายว่าเป็นอย่างอื่น = 27 ครั้ง

ตารางที่ 4.5 เมทริกซ์ความสับสนแบบจำลองนาอีฟเบย์ด้วยข้อมูลทดสอบ โดยใช้ตัวตัดคำ PyThaiNLP

ผลทำนาย / ผลจริง	Positive	Negative	Neutral	รวม
Positive	228	36	9	273
Negative	121	87	18	238
Neutral	81	31	60	160
รวม	430	154	87	671

จากตารางที่ 4.5 สามารถคำนวณตัววัดประสิทธิภาพทั้ง 4 ตัววัด ดังนี้

1. ค่าความถูกต้อง (Accuracy) = 55.89%
2. ค่าความแม่นยำ (Precision) = 58.27%
3. ค่าความระลึก (Recall) = 55.89%
4. ค่าความถ่วงดุล (F1-Score) = 53.68%

4.5.2.2 ผลจากแบบจำลองนาอีฟเบย์ (Naïve Bayes) โดยใช้ตัวตัดคำ DeepCut

จากผลในการหาพารามิเตอร์ที่ดีที่สุดของไลบรารี OptunaSearchCV ค่าพารามิเตอร์ที่ใช้ในการเพิ่มประสิทธิภาพแบบจำลองนาอีฟเบย์ คือ alpha = 0.7614289078151467 โดยหลังจากปรับค่าพารามิเตอร์แล้วได้ผลการทดลอบประสิทธิภาพของนาอีฟเบย์ ด้วยข้อมูลทดสอบสามารถแสดงได้ดังตารางที่ 4.6 พบว่าจากความคิดเห็นต่อโครงการบ้านเพื่อคนไทยสำหรับทดสอบ 671 ข้อความ ดังนี้

จำนวนข้อความที่เป็น Positive จริงและทำนายว่าเป็น Positive = 224 ครั้ง
จำนวนข้อความที่เป็น Positive จริงและทำนายว่าเป็นอย่างอื่น = 208 ครั้ง
จำนวนข้อความที่เป็น Negative จริงและทำนายว่าเป็น Negative = 95 ครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จำนวนข้อความที่เป็น Negative จริงและทำนายว่าเป็นอย่างอื่น = 73 ครั้ง
จำนวนข้อความที่เป็น Neutral จริงและทำนายว่าเป็น Neutral = 46 ครั้ง
จำนวนข้อความที่เป็น Neutral จริงและทำนายว่าเป็นอย่างอื่น = 25 ครั้ง

ตารางที่ 4.6 เมทริกซ์ความสับสนแบบจำลองนาอีฟเบย์ด้วยข้อมูลทดสอบ โดยใช้ตัวตัดคำ DeepCut

ผลทำนาย / ผลจริง	Positive	Negative	Neutral	รวม
Positive	224	39	10	273
Negative	128	95	15	238
Neutral	80	34	46	160
รวม	432	168	71	671

จากตารางที่ 4.6 สามารถคำนวณตัววัดประสิทธิภาพทั้ง 4 ตัววัด ดังนี้

1. ค่าความถูกต้อง (Accuracy) = 54.40%
2. ค่าความแม่นยำ (Precision) = 56.60%
3. ค่าความระลึก (Recall) = 54.40%
4. ค่าความถ่วงดุล (F1-Score) = 51.95%

4.5.3 ผลจากแบบจำลองการหาค่าด้วยเพื่อนบ้านใกล้ที่สุด k จุด (K-Nearest Neighbors)

4.5.3.1 ผลจากแบบจำลองการหาค่าด้วยเพื่อนบ้านใกล้ที่สุด k จุด (K-Nearest Neighbors) โดยใช้ตัวตัดคำ PyThaiNLP

จากผลในการหาพารามิเตอร์ที่ดีที่สุดของไลบรารี่ OptunaSearchCV ค่าพารามิเตอร์ที่ใช้ในการเพิ่มประสิทธิภาพแบบจำลองการหาค่าด้วยเพื่อนบ้านใกล้ที่สุด k จุด คือ n_neighbors = 4 และ weights = ‘uniform’ โดยหลังจากปรับค่าพารามิเตอร์แล้วได้ผลการทดลอบประสิทธิภาพของการหาค่าด้วยเพื่อนบ้านใกล้ที่สุด k จุด ด้วยข้อมูลทดสอบสามารถแสดงได้ดังตารางที่ 4.7 พบว่าจากความคิดเห็นต่อโครงการบ้านเพื่อคนไทยสำหรับทดสอบ 671 ข้อความ ดังนี้

จำนวนข้อความที่เป็น Positive จริงและทำนายว่าเป็น Positive = 94 ครั้ง
จำนวนข้อความที่เป็น Positive จริงและทำนายว่าเป็นอย่างอื่น = 53 ครั้ง
จำนวนข้อความที่เป็น Negative จริงและทำนายว่าเป็น Negative = 70 ครั้ง
จำนวนข้อความที่เป็น Negative จริงและทำนายว่าเป็นอย่างอื่น = 86 ครั้ง
จำนวนข้อความที่เป็น Neutral จริงและทำนายว่าเป็น Neutral = 125 ครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จำนวนข้อความที่เป็น Neutral จริงและทำนายว่าเป็นอย่างอื่น = 368 ครั้ง

ตารางที่ 4.7 เมทริกซ์ความสับสนแบบจำลองการหาค่าด้วยเพื่อนบ้านใกล้ที่สุด k จุด ด้วยข้อมูลทดสอบ โดยใช้ตัวตัดคำ PyThaiNLP

ผลทำนาย / ผลจริง	Positive	Negative	Neutral	รวม
Positive	94	55	124	273
Negative	37	70	119	226
Neutral	16	31	125	172
รวม	147	156	368	671

จากตารางที่ 4.7 สามารถคำนวณตัววัดประสิทธิภาพทั้ง 4 ตัววัด ดังนี้

1. ค่าความถูกต้อง (Accuracy) = 43.07%
2. ค่าความแม่นยำ (Precision) = 49.84%
3. ค่าความระลึก (Recall) = 43.07%
4. ค่าความถ่วงดุล (F1-Score) = 42.42%

4.5.3.2 ผลจากแบบจำลองการหาค่าด้วยเพื่อนบ้านใกล้ที่สุด k จุด (K-Nearest Neighbors)โดยใช้ตัวตัดคำ DeepCut

จากผลในการหาพารามิเตอร์ที่ดีที่สุดของไลบรารี่ OptunaSearchCV ค่าพารามิเตอร์ที่ใช้ในการเพิ่มประสิทธิภาพแบบจำลองการหาค่าด้วยเพื่อนบ้านใกล้ที่สุด k จุด คือ n_neighbors = 4 และ weights = 'distance' โดยหลังจากปรับค่าพารามิเตอร์แล้วได้ผลการทดลอบประสิทธิภาพของการหาค่าด้วยเพื่อนบ้านใกล้ที่สุด k จุด ด้วยข้อมูลทดสอบสามารถแสดงได้ดังตารางที่ 4.8 พบว่าจากความคิดเห็นต่อโครงการบ้านเพื่อคนไทยสำหรับทดสอบ 671 ข้อความ ดังนี้

จำนวนข้อความที่เป็น Positive จริงและทำนายว่าเป็น Positive = 113 ครั้ง
จำนวนข้อความที่เป็น Positive จริงและทำนายว่าเป็นอย่างอื่น = 57 ครั้ง
จำนวนข้อความที่เป็น Negative จริงและทำนายว่าเป็น Negative = 74 ครั้ง
จำนวนข้อความที่เป็น Negative จริงและทำนายว่าเป็นอย่างอื่น = 71 ครั้ง
จำนวนข้อความที่เป็น Neutral จริงและทำนายว่าเป็น Neutral = 116 ครั้ง
จำนวนข้อความที่เป็น Neutral จริงและทำนายว่าเป็นอย่างอื่น = 240 ครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.8 เมทริกซ์ความสับสนแบบจำลองการหาค่าด้วยเพื่อนบ้านใกล้ที่สุด k จุด ด้วยข้อมูลทดสอบ โดยใช้ตัวตัดคำ DeepCut

ผลทำนาย / ผลจริง	Positive	Negative	Neutral	รวม
Positive	113	38	122	273
Negative	46	74	118	238
Neutral	11	33	116	160
รวม	170	145	356	671

จากตารางที่ 4.8 สามารถคำนวณตัววัดประสิทธิภาพทั้ง 4 ตัววัด ดังนี้

1. ค่าความถูกต้อง (Accuracy) = 45.16%
2. ค่าความแม่นยำ (Precision) = 52.92%
3. ค่าความระลึก (Recall) = 45.16%
4. ค่าความถ่วงดุล (F1-Score) = 45.18%

4.5.4 ผลจากแบบจำลองซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machine)

4.5.4.1 ผลจากแบบจำลองซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machine) โดยใช้ตัวตัดคำ PyThaiNLP

จากผลในการหาพารามิเตอร์ที่ดีที่สุดของไลบรารี่ OptunaSearchCV ค่าพารามิเตอร์ที่ใช้ในการเพิ่มประสิทธิภาพแบบจำลองซัพพอร์ตเวกเตอร์แมชชีน คือ C = 2.2204151590602295, gamma = ‘scale’ และ Kernel = ‘rbf’ โดยหลังจากปรับค่าพารามิเตอร์แล้วได้ผลการทดลองประสิทธิภาพของซัพพอร์ตเวกเตอร์แมชชีน ด้วยข้อมูลทดสอบสามารถแสดงได้ดังตารางที่ 4.9 พบว่าจากความคิดเห็นต่อโครงการบ้านเพื่อคนไทยสำหรับทดสอบ 671 ข้อความ ดังนี้

จำนวนข้อความที่เป็น Positive จริงและทำนายว่าเป็น Positive = 190 ครั้ง
จำนวนข้อความที่เป็น Positive จริงและทำนายว่าเป็นอย่างอื่น = 124 ครั้ง
จำนวนข้อความที่เป็น Negative จริงและทำนายว่าเป็น Negative = 112 ครั้ง
จำนวนข้อความที่เป็น Negative จริงและทำนายว่าเป็นอย่างอื่น = 103 ครั้ง
จำนวนข้อความที่เป็น Neutral จริงและทำนายว่าเป็น Neutral = 101 ครั้ง
จำนวนข้อความที่เป็น Neutral จริงและทำนายว่าเป็นอย่างอื่น = 41 ครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.9 ผลการทดสอบแบบจำลองซัพพอร์ตเวกเตอร์แมชชีนด้วยข้อมูลทดสอบ โดยใช้ตัวตัดคำ PyThaiNLP

ผลทำนาย / ผลจริง	Positive	Negative	Neutral	รวม
Positive	190	66	17	273
Negative	90	112	24	226
Neutral	34	37	101	172
รวม	314	215	142	671

จากตารางที่ 4.9 สามารถคำนวณตัววัดประสิทธิภาพทั้ง 4 ตัววัด ดังนี้

1. ค่าความถูกต้อง (Accuracy) = 60.06%
2. ค่าความแม่นยำ (Precision) = 60.40%
3. ค่าความระลึก (Recall) = 60.06%
4. ค่าความถ่วงดุล (F1-Score) = 59.94%

4.5.4.2 ผลจากแบบจำลองซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machine) โดยใช้ตัวตัดคำ DeepCut

จากผลในการหาพารามิเตอร์ที่ดีที่สุดของไลบรารี่ OptunaSearchCV ค่าพารามิเตอร์ที่ใช้ในการเพิ่มประสิทธิภาพแบบจำลองซัพพอร์ตเวกเตอร์แมชชีน คือ C = 1, gamma = 'scale' และ Kernel = 'rbf' โดยหลังจากปรับค่าพารามิเตอร์แล้วได้ผลการทดลอบประสิทธิภาพของซัพพอร์ตเวกเตอร์แมชชีน ด้วยข้อมูลทดสอบสามารถแสดงได้ดังตารางที่ 4.10 พบว่าจากความคิดเห็นต่อโครงการบ้านเพื่อคนไทยสำหรับทดสอบ 671 ข้อความ ดังนี้

จำนวนข้อความที่เป็น Positive จริงและทำนายว่าเป็น Positive = 207 ครั้ง
จำนวนข้อความที่เป็น Positive จริงและทำนายว่าเป็นอย่างอื่น = 142 ครั้ง
จำนวนข้อความที่เป็น Negative จริงและทำนายว่าเป็น Negative = 109 ครั้ง
จำนวนข้อความที่เป็น Negative จริงและทำนายว่าเป็นอย่างอื่น = 95 ครั้ง
จำนวนข้อความที่เป็น Neutral จริงและทำนายว่าเป็น Neutral = 80 ครั้ง
จำนวนข้อความที่เป็น Neutral จริงและทำนายว่าเป็นอย่างอื่น = 38 ครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.10 ผลการทดสอบแบบจำลองซัพพอร์ตเวกเตอร์แมชชีนด้วยข้อมูลทดสอบ โดยใช้ตัวตัดคำ DeepCut

ผลทำนาย / ผลจริง	Positive	Negative	Neutral	รวม
Positive	207	49	17	273
Negative	108	109	21	238
Neutral	34	46	80	160
รวม	354	204	118	671

จากตารางที่ 4.10 สามารถคำนวณตัววัดประสิทธิภาพทั้ง 4 ตัววัด ดังนี้

1. ค่าความถูกต้อง (Accuracy) = 59.02%
2. ค่าความแม่นยำ (Precision) = 59.25%
3. ค่าความระลึก (Recall) = 59.02%
4. ค่าความถ่วงดุล (F1-Score) = 58.30%

4.5.5 ผลจากแบบจำลองป่าสุ่ม (Random Forest)

4.5.5.1 ผลจากแบบจำลองป่าสุ่ม (Random Forest) โดยใช้ตัวตัดคำ PyThaiNLP

จากผลในการหาพารามิเตอร์ที่ดีที่สุดของไลบรารี OptunaSearchCV ค่าพารามิเตอร์ที่ใช้ในการเพิ่มประสิทธิภาพแบบจำลองป่าสุ่ม คือ n_estimators = 179, max-features = ‘sqrt’ และ max_depth = 27 โดยหลังจากปรับค่าพารามิเตอร์แล้วได้ผลการทดลอบประสิทธิภาพของป่าสุ่มด้วยข้อมูลทดสอบสามารถแสดงได้ดังตารางที่ 4.11 พบว่าจากความคิดเห็นต่อโครงการบ้านเพื่อคนไทยสำหรับทดสอบ 671 ข้อความ ดังนี้

จำนวนข้อความที่เป็น Positive จริงและทำนายว่าเป็น Positive = 232 ครั้ง
จำนวนข้อความที่เป็น Positive จริงและทำนายว่าเป็นอย่างอื่น = 230 ครั้ง
จำนวนข้อความที่เป็น Negative จริงและทำนายว่าเป็น Negative = 71 ครั้ง
จำนวนข้อความที่เป็น Negative จริงและทำนายว่าเป็นอย่างอื่น = 65 ครั้ง
จำนวนข้อความที่เป็น Neutral จริงและทำนายว่าเป็น Neutral = 53 ครั้ง
จำนวนข้อความที่เป็น Neutral จริงและทำนายว่าเป็นอย่างอื่น = 20 ครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.11 เมทริกซ์ความสับสนแบบจำลองป่าสุ่มด้วยข้อมูลทดสอบ โดยใช้ตัวตัดคำ PyThaiNLP

ผลทำนาย / ผลจริง	Positive	Negative	Neutral	รวม
Positive	232	35	6	273
Negative	141	71	14	226
Neutral	89	30	53	172
รวม	431	165	75	671

จากตารางที่ 4.11 สามารถคำนวณตัววัดประสิทธิภาพทั้ง 4 ตัววัด ดังนี้

1. ค่าความถูกต้อง (Accuracy) = 53.06%
2. ค่าความแม่นยำ (Precision) = 56.62%
3. ค่าความระลึก (Recall) = 53.06%
4. ค่าความถ่วงดุล (F1-Score) = 49.99%

4.5.5.2 ผลจากแบบจำลองป่าสุ่ม (Random Forest) โดยใช้ตัวตัดคำ DeepCut

จากผลในการหาพารามิเตอร์ที่ดีที่สุดของไลบรารี OptunaSearchCV ค่าพารามิเตอร์ที่ใช้ในการเพิ่มประสิทธิภาพแบบจำลองป่าสุ่ม คือ n_estimators = 191, max_features = 'sqrt' และ max_depth = 30 โดยหลังจากปรับค่าพารามิเตอร์แล้วได้ผลการทดลอบประสิทธิภาพของป่าสุ่มด้วยข้อมูลทดสอบสามารถแสดงได้ดังตารางที่ 4.12 พบว่าจากความคิดเห็นต่อโครงการบ้านเพื่อคนไทยสำหรับทดสอบ 671 ข้อความ ดังนี้

จำนวนข้อความที่เป็น Positive จริงและทำนายว่าเป็น Positive	=	231	ครั้ง
จำนวนข้อความที่เป็น Positive จริงและทำนายว่าเป็นอย่างอื่น	=	200	ครั้ง
จำนวนข้อความที่เป็น Negative จริงและทำนายว่าเป็น Negative	=	92	ครั้ง
จำนวนข้อความที่เป็น Negative จริงและทำนายว่าเป็นอย่างอื่น	=	73	ครั้ง
จำนวนข้อความที่เป็น Neutral จริงและทำนายว่าเป็น Neutral	=	49	ครั้ง
จำนวนข้อความที่เป็น Neutral จริงและทำนายว่าเป็นอย่างอื่น	=	26	ครั้ง

ตารางที่ 4.12 เมทริกซ์ความสับสนแบบจำลองป่าสุ่มด้วยข้อมูลทดสอบ โดยใช้ตัวตัดคำ DeepCut

ผลทำนาย / ผลจริง	Positive	Negative	Neutral	รวม
Positive	231	32	10	273
Negative	130	92	16	238

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.12 เมทริกซ์ความสับสนแบบจำลองป่าสุ่มด้วยข้อมูลทดสอบ โดยใช้ตัวตัดคำ DeepCut (ต่อ)

ผลทำนาย / ผลจริง	Positive	Negative	Neutral	รวม
Neutral	70	41	49	160
รวม	431	165	75	671

จากตารางที่ 4.12 สามารถคำนวณตัววัดประสิทธิภาพทั้ง 4 ตัววัด ดังนี้

1. ค่าความถูกต้อง (Accuracy) = 55.44%
2. ค่าความแม่นยำ (Precision) = 57.16%
3. ค่าความระลึก (Recall) = 55.44%
4. ค่าความถ่วงดุล (F1-Score) = 52.84%

4.5.6 ผลจากแบบจำลองต้นไม้ตัดสินใจ (Decision Tree)

4.5.6.1 ผลจากแบบจำลองต้นไม้ตัดสินใจ (Decision Tree) โดยใช้ตัวตัดคำ PyThaiNLP

จากผลในการหาพารามิเตอร์ที่ดีที่สุดของไลบรารี่ OptunaSearchCV ค่าพารามิเตอร์ที่ใช้ในการเพิ่มประสิทธิภาพแบบจำลองต้นไม้ตัดสินใจ คือ max_depth = 30 โดยหลังจากปรับค่าพารามิเตอร์แล้วได้ผลการทดลอบประสิทธิภาพของต้นไม้ตัดสินใจ ด้วยข้อมูลทดสอบสามารถแสดงได้ดังตารางที่ 4.13 พบว่าจากความคิดเห็นต่อโครงการบ้านเพื่อคนไทยสำหรับทดสอบ 671 ข้อความ ดังนี้

จำนวนข้อความที่เป็น Positive จริงและทำนายว่าเป็น Positive = 111 ครั้ง
จำนวนข้อความที่เป็น Positive จริงและทำนายว่าเป็นอย่างอื่น = 80 ครั้ง
จำนวนข้อความที่เป็น Negative จริงและทำนายว่าเป็น Negative = 152 ครั้ง
จำนวนข้อความที่เป็น Negative จริงและทำนายว่าเป็นอย่างอื่น = 244 ครั้ง
จำนวนข้อความที่เป็น Neutral จริงและทำนายว่าเป็น Neutral = 49 ครั้ง
จำนวนข้อความที่เป็น Neutral จริงและทำนายว่าเป็นอย่างอื่น = 35 ครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.13 เมทริกซ์ความสับสนแบบจำลองต้นไม้ตัดสินใจด้วยข้อมูลทดสอบ โดยใช้ตัวตัดคำ PyThaiNLP

ผลทำนาย / ผลจริง	Positive	Negative	Neutral	รวม
Positive	111	145	17	273
Negative	56	152	18	226
Neutral	24	99	49	172
รวม	191	396	84	671

จากตารางที่ 4.13 สามารถคำนวณตัววัดประสิทธิภาพทั้ง 4 ตัววัด ดังนี้

1. ค่าความถูกต้อง (Accuracy) = 46.50%
2. ค่าความแม่นยำ (Precision) = 51.53%
3. ค่าความระลึก (Recall) = 46.50%
4. ค่าความถ่วงดุล (F1-Score) = 45.74%

4.5.6.2 ผลจากแบบจำลองต้นไม้ตัดสินใจ (Decision Tree) โดยใช้ตัวตัดคำ DeepCut

จากผลในการหาพารามิเตอร์ที่ดีที่สุดของไลบรารี่ OptunaSearchCV ค่าพารามิเตอร์ที่ใช้ในการเพิ่มประสิทธิภาพแบบจำลองต้นไม้ตัดสินใจ คือ max_depth = 30 โดยหลังจากปรับค่าพารามิเตอร์แล้วได้ผลการทดลอบประสิทธิภาพของต้นไม้ตัดสินใจ ด้วยข้อมูลทดสอบสามารถแสดงได้ดังตารางที่ 4.14 พบว่าจากความคิดเห็นต่อโครงการบ้านเพื่อคนไทยสำหรับทดสอบ 671 ข้อความ ดังนี้

จำนวนข้อความที่เป็น Positive จริงและทำนายว่าเป็น Positive = 141 ครั้ง
จำนวนข้อความที่เป็น Positive จริงและทำนายว่าเป็นอย่างอื่น = 76 ครั้ง
จำนวนข้อความที่เป็น Negative จริงและทำนายว่าเป็น Negative = 167 ครั้ง
จำนวนข้อความที่เป็น Negative จริงและทำนายว่าเป็นอย่างอื่น = 226 ครั้ง
จำนวนข้อความที่เป็น Neutral จริงและทำนายว่าเป็น Neutral = 30 ครั้ง
จำนวนข้อความที่เป็น Neutral จริงและทำนายว่าเป็นอย่างอื่น = 31 ครั้ง

ตารางที่ 4.14 เมทริกซ์ความสับสนแบบจำลองต้นไม้ตัดสินใจด้วยข้อมูลทดสอบ โดยใช้ตัวตัดคำ DeepCut

ผลทำนาย / ผลจริง	Positive	Negative	Neutral	รวม
Positive	141	119	13	273

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.14 เมทริกซ์ความสับสนแบบจำลองต้นไม้ตัดสินใจด้วยข้อมูลทดสอบ โดยใช้ตัวตัดคำ DeepCut (ต่อ)

ผลทำนาย / ผลจริง	Positive	Negative	Neutral	รวม
Negative	53	167	18	238
Neutral	23	107	30	160
รวม	217	393	61	671

จากตารางที่ 4.14 สามารถคำนวณตัววัดประสิทธิภาพทั้ง 4 ตัววัด ดังนี้

1. ค่าความถูกต้อง (Accuracy) = 50.37%
2. ค่าความแม่นยำ (Precision) = 53.24%
3. ค่าความระลึก (Recall) = 50.37%
4. ค่าความถ่วงดุล (F1-Score) = 48.66%

4.5.7 ผลจากแบบจำลองโครงข่ายประสาทเทียมหลายชั้น (Multi-Layer Perceptron)

4.5.7.1 ผลจากแบบจำลองโครงข่ายประสาทเทียมหลายชั้น (Multi-Layer Perceptron) โดยใช้ตัวตัดคำ PyThaiNLP

จากผลในการหาพารามิเตอร์ที่ดีที่สุดของไลบรารี่ OptunaSearchCV ค่าพารามิเตอร์ที่ใช้ในการเพิ่มประสิทธิภาพแบบจำลองโครงข่ายประสาทเทียมหลายชั้น คือ hidden_layer_sizes = (100,), activation = ‘relu’ และ alpha = 0.006076304590109201 โดยหลังจากปรับค่าพารามิเตอร์แล้วได้ผลการทดลอบประสิทธิภาพของโครงข่ายประสาทเทียมหลายชั้น ด้วยข้อมูลทดสอบสามารถแสดงได้ดังตารางที่ 4.15 พบว่าจากความคิดเห็นต่อโครงการบ้านเพื่อคนไทยสำหรับทดสอบ 671 ข้อความ ดังนี้

จำนวนข้อความที่เป็น Positive จริงและทำนายว่าเป็น Positive = 166 ครั้ง
จำนวนข้อความที่เป็น Positive จริงและทำนายว่าเป็นอย่างอื่น = 107 ครั้ง
จำนวนข้อความที่เป็น Negative จริงและทำนายว่าเป็น Negative = 103 ครั้ง
จำนวนข้อความที่เป็น Negative จริงและทำนายว่าเป็นอย่างอื่น = 109 ครั้ง
จำนวนข้อความที่เป็น Neutral จริงและทำนายว่าเป็น Neutral = 111 ครั้ง
จำนวนข้อความที่เป็น Neutral จริงและทำนายว่าเป็นอย่างอื่น = 75 ครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.15 เมทริกซ์ความสับสนแบบจำลองโครงข่ายประสาทเทียมหลายชั้นด้วยข้อมูลทดสอบ โดยใช้ตัวตัดคำ PyThaiNLP

ผลทำนาย / ผลจริง	Positive	Negative	Neutral	รวม
Positive	166	75	32	273
Negative	80	103	43	226
Neutral	27	34	111	172
รวม	273	212	186	671

จากตารางที่ 4.15 สามารถคำนวณตัววัดประสิทธิภาพทั้ง 4 ตัววัด ดังนี้

1. ค่าความถูกต้อง (Accuracy) = 56.63%
2. ค่าความแม่นยำ (Precision) = 56.40%
3. ค่าความระลึก (Recall) = 56.63%
4. ค่าความถ่วงดุล (F1-Score) = 56.48%

4.5.7.2 ผลจากแบบจำลองโครงข่ายประสาทเทียมหลายชั้น (Multi-Layer Perceptron) โดยใช้ตัวตัดคำ DeepCut

จากผลในการหาพารามิเตอร์ที่ดีที่สุดของไลบรารี่ OptunaSearchCV ค่าพารามิเตอร์ที่ใช้ในการเพิ่มประสิทธิภาพแบบจำลองโครงข่ายประสาทเทียมหลายชั้น คือ hidden_layer_sizes = (100,), activation = 'relu' และ alpha = 0.003918127210188846 โดยหลังจากปรับค่าพารามิเตอร์แล้วได้ผลการทดลอบประสิทธิภาพของโครงข่ายประสาทเทียมหลายชั้น ด้วยข้อมูลทดสอบสามารถแสดงได้ดังตารางที่ 4.16 พบว่าจากความคิดเห็นต่อโครงการบ้านเพื่อคนไทยสำหรับทดสอบ 671 ข้อความ ดังนี้

จำนวนข้อความที่เป็น Positive จริงและทำนายว่าเป็น Positive = 161 ครั้ง
จำนวนข้อความที่เป็น Positive จริงและทำนายว่าเป็นอย่างอื่น = 105 ครั้ง
จำนวนข้อความที่เป็น Negative จริงและทำนายว่าเป็น Negative = 107 ครั้ง
จำนวนข้อความที่เป็น Negative จริงและทำนายว่าเป็นอย่างอื่น = 105 ครั้ง
จำนวนข้อความที่เป็น Neutral จริงและทำนายว่าเป็น Neutral = 95 ครั้ง
จำนวนข้อความที่เป็น Neutral จริงและทำนายว่าเป็นอย่างอื่น = 90 ครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.16 เมทริกซ์ความสับสนแบบจำลองโครงข่ายประสาทเทียมหลายชั้นด้วยข้อมูลทดสอบ โดยใช้ตัวตัดคำ DeepCut

ผลทำนาย / ผลจริง	Positive	Negative	Neutral	รวม
Positive	161	63	49	273
Negative	75	107	56	238
Neutral	29	36	95	160
รวม	265	206	200	671

จากตารางที่ 4.16 สามารถคำนวณตัววัดประสิทธิภาพทั้ง 4 ตัววัด ดังนี้

1. ค่าความถูกต้อง (Accuracy) = 54.10%
2. ค่าความแม่นยำ (Precision) = 54.47%
3. ค่าความระลึก (Recall) = 54.10%
4. ค่าความถ่วงดุล (F1-Score) = 54.03%

4.5.8 ผลจากแบบจำลองเพอร์เซปตรอน (Perceptron)

4.5.8.1 ผลจากแบบจำลองเพอร์เซปตรอน (Perceptron) โดยใช้ตัวตัดคำ PyThaiNLP

จากผลในการหาพารามิเตอร์ที่ดีที่สุดของไลบรารี่ OptunaSearchCV ค่าพารามิเตอร์ที่ใช้ในการเพิ่มประสิทธิภาพแบบจำลองเพอร์เซปตรอน คือ alpha = 0.0002851505094955989 และ penalty = ‘l2’ โดยหลังจากปรับค่าพารามิเตอร์แล้วได้ผลการทดลอบประสิทธิภาพของเพอร์เซปตรอน ด้วยข้อมูลทดสอบสามารถแสดงได้ดังตารางที่ 4.17 พบว่าจากความคิดเห็นต่อโครงการบ้านเพื่อคนไทยสำหรับทดสอบ 671 ข้อความ ดังนี้

จำนวนข้อความที่เป็น Positive จริงและทำนายว่าเป็น Positive = 164 ครั้ง
จำนวนข้อความที่เป็น Positive จริงและทำนายว่าเป็นอย่างอื่น = 171 ครั้ง
จำนวนข้อความที่เป็น Negative จริงและทำนายว่าเป็น Negative = 109 ครั้ง
จำนวนข้อความที่เป็น Negative จริงและทำนายว่าเป็นอย่างอื่น = 120 ครั้ง
จำนวนข้อความที่เป็น Neutral จริงและทำนายว่าเป็น Neutral = 52 ครั้ง
จำนวนข้อความที่เป็น Neutral จริงและทำนายว่าเป็นอย่างอื่น = 55 ครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.17 เมทริกซ์ความสับสนแบบจำลองเพอร์เซปตรอนด้วยข้อมูลทดสอบ โดยใช้ตัวตัดคำ PyThaiNLP

ผลทำนาย / ผลจริง	Positive	Negative	Neutral	รวม
Positive	164	83	26	273
Negative	88	109	29	226
Neutral	83	37	52	172
รวม	335	229	107	671

จากตารางที่ 4.17 สามารถคำนวณตัววัดประสิทธิภาพทั้ง 4 ตัววัด ดังนี้

1. ค่าความถูกต้อง (Accuracy) = 48.44%
2. ค่าความแม่นยำ (Precision) = 48.41%
3. ค่าความระลึก (Recall) = 48.44%
4. ค่าความถ่วงดุล (F1-Score) = 47.64%

4.5.8.2 ผลจากแบบจำลองเพอร์เซปตรอน (Perceptron) โดยใช้ตัวตัดคำ DeepCut

จากผลในการหาพารามิเตอร์ที่ดีที่สุดของไลบรารี OptunaSearchCV ค่าพารามิเตอร์ที่ใช้ในการเพิ่มประสิทธิภาพแบบจำลองเพอร์เซปตรอน คือ alpha = 0.00043977073432509593 และค่า penalty = None โดยหลังจากปรับค่าพารามิเตอร์แล้วได้ผลการทดลอบประสิทธิภาพของเพอร์เซปตรอน ด้วยข้อมูลทดสอบสามารถแสดงได้ดังตารางที่ 4.18 พบว่าจากความคิดเห็นต่อโครงการบ้านเพื่อคนไทยสำหรับทดสอบ 671 ข้อความ ดังนี้

จำนวนข้อความที่เป็น Positive จริงและทำนายว่าเป็น Positive = 153 ครั้ง
จำนวนข้อความที่เป็น Positive จริงและทำนายว่าเป็นอย่างอื่น = 112 ครั้ง
จำนวนข้อความที่เป็น Negative จริงและทำนายว่าเป็น Negative = 98 ครั้ง
จำนวนข้อความที่เป็น Negative จริงและทำนายว่าเป็นอย่างอื่น = 86 ครั้ง
จำนวนข้อความที่เป็น Neutral จริงและทำนายว่าเป็น Neutral = 107 ครั้ง
จำนวนข้อความที่เป็น Neutral จริงและทำนายว่าเป็นอย่างอื่น = 115 ครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.18 เมทริกซ์ความสับสนแบบจำลองเพอร์เซปตรอนด้วยข้อมูลทดสอบ โดยใช้ตัวตัดคำ DeepCut

ผลทำนาย / ผลจริง	Positive	Negative	Neutral	รวม
Positive	153	63	57	273
Negative	82	98	58	238
Neutral	30	23	107	160
รวม	265	184	222	671

จากตารางที่ 4.18 สามารถคำนวณตัววัดประสิทธิภาพทั้ง 4 ตัววัด ดังนี้

1. ค่าความถูกต้อง (Accuracy) = 53.35%
2. ค่าความแม่นยำ (Precision) = 53.87%
3. ค่าความระลึก (Recall) = 53.35%
4. ค่าความถ่วงดุล (F1-Score) = 52.97%

4.5.9 ผลจากแบบจำลองสโตแคสติกการ์เดียนดิเซนท์ (Stochastic Gradient Descent)

4.5.9.1 ผลจากแบบจำลองสโตแคสติกการ์เดียนดิเซนท์ (Stochastic Gradient Descent) โดยใช้ตัวตัดคำ PyThaiNLP

จากผลในการหาพารามิเตอร์ที่ดีที่สุดของไลบรารี OptunaSearchCV ค่าพารามิเตอร์ที่ใช้ในการเพิ่มประสิทธิภาพแบบจำลองสโตแคสติกการ์เดียนดิเซนท์ คือ alpha = 0.00011281916225850828, loss = 'log_loss' และ penalty = 'l1' โดยหลังจากปรับค่าพารามิเตอร์แล้วได้ผลการทดลอบประสิทธิภาพของแบบจำลองสโตแคสติกการ์เดียนดิเซนท์ ด้วยข้อมูลทดสอบสามารถแสดงได้ดังตารางที่ 4.19 พบว่าจากความคิดเห็นต่อโครงการบ้านเพื่อคนไทยสำหรับทดสอบ 671 ข้อความ ดังนี้

จำนวนข้อความที่เป็น Positive จริงและทำนายว่าเป็น Positive	=	177	ครั้ง
จำนวนข้อความที่เป็น Positive จริงและทำนายว่าเป็นอย่างอื่น	=	115	ครั้ง
จำนวนข้อความที่เป็น Negative จริงและทำนายว่าเป็น Negative	=	102	ครั้ง
จำนวนข้อความที่เป็น Negative จริงและทำนายว่าเป็นอย่างอื่น	=	103	ครั้ง
จำนวนข้อความที่เป็น Neutral จริงและทำนายว่าเป็น Neutral	=	110	ครั้ง
จำนวนข้อความที่เป็น Neutral จริงและทำนายว่าเป็นอย่างอื่น	=	64	ครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.19 เมทริกซ์ความสับสนแบบจำลองสโตแคสติกการ์เดียนดิเซนท์ด้วยข้อมูลทดสอบ โดยใช้ตัวตัดคำ PyThaiNLP

ผลทำนาย / ผลจริง	Positive	Negative	Neutral	รวม
Positive	177	68	28	273
Negative	88	102	36	226
Neutral	27	35	110	172
รวม	292	205	174	671

จากตารางที่ 4.19 สามารถคำนวณตัววัดประสิทธิภาพทั้ง 4 ตัววัด ดังนี้

1. ค่าความถูกต้อง (Accuracy) = 57.97%
2. ค่าความแม่นยำ (Precision) = 57.63%
3. ค่าความระลึก (Recall) = 57.97%
4. ค่าความถ่วงดุล (F1-Score) = 57.73%

4.5.9.2 ผลจากแบบจำลองสโตแคสติกการ์เดียนดิเซนท์ (Stochastic Gradient Descent) โดยใช้ตัวตัดคำ DeepCut

จากผลในการหาพารามิเตอร์ที่ดีที่สุดของไลบรารี่ OptunaSearchCV ค่าพารามิเตอร์ที่ใช้ในการเพิ่มประสิทธิภาพแบบจำลองสโตแคสติกการ์เดียนดิเซนท์ คือ alpha = 0.0010838329977700823, loss = 'hinge' และ penalty = 'elasticnet' โดยหลังจากปรับค่าพารามิเตอร์แล้วได้ผลการทดลอบประสิทธิภาพของแบบจำลองสโตแคสติกการ์เดียนดิเซนท์ ด้วยข้อมูลทดสอบสามารถแสดงได้ดังตารางที่ 4.20 พบว่าจากความคิดเห็นต่อโครงการบ้านเพื่อคนไทยสำหรับทดสอบ 671 ข้อความ ดังนี้

จำนวนข้อความที่เป็น Positive จริงและทำนายว่าเป็น Positive = 210 ครั้ง
จำนวนข้อความที่เป็น Positive จริงและทำนายว่าเป็นอย่างอื่น = 141 ครั้ง
จำนวนข้อความที่เป็น Negative จริงและทำนายว่าเป็น Negative = 85 ครั้ง
จำนวนข้อความที่เป็น Negative จริงและทำนายว่าเป็นอย่างอื่น = 67 ครั้ง
จำนวนข้อความที่เป็น Neutral จริงและทำนายว่าเป็น Neutral = 96 ครั้ง
จำนวนข้อความที่เป็น Neutral จริงและทำนายว่าเป็นอย่างอื่น = 72 ครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.20 เมทริกซ์ความสับสนแบบจำลองสโตแคสติกการ์เดียนดิเซนท์ด้วยข้อมูลทดสอบ โดยใช้ตัวตัดคำ DeepCut

ผลทำนาย / ผลจริง	Positive	Negative	Neutral	รวม
Positive	210	35	28	273
Negative	109	85	44	238
Neutral	32	32	96	160
รวม	351	67	72	671

จากตารางที่ 4.20 สามารถคำนวณตัวชี้วัดประสิทธิภาพทั้ง 4 ตัววัด ดังนี้

1. ค่าความถูกต้อง (Accuracy) = 58.27%
2. ค่าความแม่นยำ (Precision) = 57.80%
3. ค่าความระลึก (Recall) = 58.27%
4. ค่าความถ่วงดุล (F1-Score) = 56.80%

4.5.10 ผลจากแบบจำลองพาสซีฟ อากัสซีฟ (Passive Aggressive)

4.5.10.1 ผลจากแบบจำลองพาสซีฟ อากัสซีฟ (Passive Aggressive) โดยใช้ตัวตัดคำ PyThaiNLP

จากผลในการหาพารามิเตอร์ที่ดีที่สุดของไลบรารี่ OptunaSearchCV ค่าพารามิเตอร์ที่ใช้ในการเพิ่มประสิทธิภาพแบบจำลองพาสซีฟ อากัสซีฟ คือ C = 0.011290944461967474 และ loss = ‘hinge’ โดยหลังจากปรับค่าพารามิเตอร์แล้วได้ผลการทดลอบประสิทธิภาพของพาสซีฟ อากัสซีฟด้วยข้อมูลทดสอบสามารถแสดงได้ดังตารางที่ 4.21 พบว่าจากความคิดเห็นต่อโครงการบ้านเพื่อคนไทยสำหรับทดสอบ 671 ข้อความ ดังนี้

จำนวนข้อความที่เป็น Positive จริงและทำนายว่าเป็น Positive = 191 ครั้ง
จำนวนข้อความที่เป็น Positive จริงและทำนายว่าเป็นอย่างอื่น = 84 ครั้ง
จำนวนข้อความที่เป็น Negative จริงและทำนายว่าเป็น Negative = 109 ครั้ง
จำนวนข้อความที่เป็น Negative จริงและทำนายว่าเป็นอย่างอื่น = 126 ครั้ง
จำนวนข้อความที่เป็น Neutral จริงและทำนายว่าเป็น Neutral = 81 ครั้ง
จำนวนข้อความที่เป็น Neutral จริงและทำนายว่าเป็นอย่างอื่น = 50 ครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.21 เมทริกซ์ความสับสนแบบจำลองพาสซีฟ อากัสซีฟด้วยข้อมูลทดสอบ โดยใช้ตัวตัดคำ PyThaiNLP

ผลทำนาย / ผลจริง	Positive	Negative	Neutral	รวม
Positive	191	65	17	273
Negative	84	109	33	226
Neutral	30	61	81	172
รวม	314	230	127	671

จากตารางที่ 4.21 สามารถคำนวณตัววัดประสิทธิภาพทั้ง 4 ตัววัด ดังนี้

1. ค่าความถูกต้อง (Accuracy) = 56.63%
2. ค่าความแม่นยำ (Precision) = 56.81%
3. ค่าความระลึก (Recall) = 56.63%
4. ค่าความถ่วงดุล (F1-Score) = 56.27%

4.5.10.2 ผลจากแบบจำลองพาสซีฟ อากัสซีฟ (Passive Aggressive) โดยใช้ตัวตัดคำ DeepCut

จากผลในการหาพารามิเตอร์ที่ดีที่สุดของไลบรารี่ OptunaSearchCV ค่าพารามิเตอร์ที่ใช้ในการเพิ่มประสิทธิภาพแบบจำลองพาสซีฟ อากัสซีฟ คือ C = 0.010354276577543788 และ loss = 'squared_hinge' โดยหลังจากปรับค่าพารามิเตอร์แล้วได้ผลการทดลอบประสิทธิภาพของพาสซีฟ อากัสซีฟ ด้วยข้อมูลทดสอบสามารถแสดงได้ดังตารางที่ 4.22 พบว่าจากความคิดเห็นต่อโครงการบ้านเพื่อคนไทยสำหรับทดสอบ 671 ข้อความ ดังนี้

จำนวนข้อความที่เป็น Positive จริงและทำนายว่าเป็น Positive = 169 ครั้ง
จำนวนข้อความที่เป็น Positive จริงและทำนายว่าเป็นอย่างอื่น = 101 ครั้ง
จำนวนข้อความที่เป็น Negative จริงและทำนายว่าเป็น Negative = 111 ครั้ง
จำนวนข้อความที่เป็น Negative จริงและทำนายว่าเป็นอย่างอื่น = 102 ครั้ง
จำนวนข้อความที่เป็น Neutral จริงและทำนายว่าเป็น Neutral = 93 ครั้ง
จำนวนข้อความที่เป็น Neutral จริงและทำนายว่าเป็นอย่างอื่น = 95 ครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.22 เมทริกซ์ความสับสนแบบจำลองพาสซีฟ อากัสซีฟด้วยข้อมูลทดสอบ โดยใช้ตัวตัดคำ DeepCut

ผลทำนาย / ผลจริง	Positive	Negative	Neutral	รวม
Positive	169	64	40	273
Negative	72	111	55	238
Neutral	29	38	93	160
รวม	270	213	188	671

จากตารางที่ 4.21 สามารถคำนวณตัววัดประสิทธิภาพทั้ง 4 ตัววัด ดังนี้

1. ค่าความถูกต้อง (Accuracy) = 55.59%
2. ค่าความแม่นยำ (Precision) = 55.75%
3. ค่าความระลึก (Recall) = 55.59%
4. ค่าความถ่วงดุล (F1-Score) = 55.53%

4.5.11 ผลจากแบบจำลองโครงข่ายประสาทหน่วยความจำระยะสั้นแบบยาว (Long Short-Term Memory)

4.5.11.1 ผลจากแบบจำลองโครงข่ายประสาทหน่วยความจำระยะสั้นแบบยาว (Long Short-Term Memory) โดยใช้ตัวตัดคำ PyThaiNLP

จากผลในการหาพารามิเตอร์ที่ดีที่สุดของไลบรารี่ OptunaSearchCV ค่าพารามิเตอร์ที่ใช้ในการเพิ่มประสิทธิภาพแบบจำลองโครงข่ายประสาทหน่วยความจำระยะสั้นแบบยาว คือ units = 128, learning_rate = 0.00012948370133259137, dropout_rate = 0.12855298825056743, epochs = 50 และ batch_size = 32 โดยหลังจากปรับค่าพารามิเตอร์แล้วได้ผลการทดลองประสิทธิภาพของโครงข่ายประสาทหน่วยความจำระยะสั้นแบบยาวด้วยข้อมูลทดสอบสามารถแสดงได้ดังตารางที่ 4.23 พบว่าจากความคิดเห็นต่อโครงการบ้านเพื่อคนไทยสำหรับทดสอบ 671 ข้อความ ดังนี้

จำนวนข้อความที่เป็น Positive จริงและทำนายว่าเป็น Positive = 192 ครั้ง
จำนวนข้อความที่เป็น Positive จริงและทำนายว่าเป็นอย่างอื่น = 123 ครั้ง
จำนวนข้อความที่เป็น Negative จริงและทำนายว่าเป็น Negative = 116 ครั้ง
จำนวนข้อความที่เป็น Negative จริงและทำนายว่าเป็นอย่างอื่น = 136 ครั้ง
จำนวนข้อความที่เป็น Neutral จริงและทำนายว่าเป็น Neutral = 74 ครั้ง
จำนวนข้อความที่เป็น Neutral จริงและทำนายว่าเป็นอย่างอื่น = 30 ครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.23 เมทริกซ์ความสับสนแบบจำลองโครงข่ายประสาทหน่วยความจำระยะสั้นแบบยาวด้วยข้อมูลทดสอบ โดยใช้ตัวตัดคำ PyThaiNLP

ผลทำนาย / ผลจริง	Positive	Negative	Neutral	รวม
Positive	191	65	17	273
Negative	84	109	33	226
Neutral	30	61	81	172
รวม	315	252	104	671

จากตารางที่ 4.23 สามารถคำนวณตัววัดประสิทธิภาพทั้ง 4 ตัววัด ดังนี้

1. ค่าความถูกต้อง (Accuracy) = 56.93%
2. ค่าความแม่นยำ (Precision) = 58.54%
3. ค่าความระลึก (Recall) = 56.93%
4. ค่าความถ่วงดุล (F1-Score) = 56.66%

4.5.11.2 ผลจากแบบจำลองโครงข่ายประสาทหน่วยความจำระยะสั้นแบบยาว (Long Short-Term Memory) โดยใช้ตัวตัดคำ DeepCut

จากผลในการหาพารามิเตอร์ที่ดีที่สุดของไลบรารี OptunaSearchCV ค่าพารามิเตอร์ที่ใช้ในการเพิ่มประสิทธิภาพแบบจำลองโครงข่ายประสาทหน่วยความจำระยะสั้นแบบยาว คือ units = 128, learning_rate = 0.0022833953700037565, dropout_rate = 0.12935980043289855, epochs = 5 และ batch_size = 32 โดยหลังจากปรับค่าพารามิเตอร์แล้วได้ผลการทดลอบประสิทธิภาพของโครงข่ายประสาทหน่วยความจำระยะสั้นแบบยาวด้วยข้อมูลทดสอบสามารถแสดงได้ดังตารางที่ 4.24 พบว่าจากความคิดเห็นต่อโครงการบ้านเพื่อคนไทยสำหรับทดสอบ 617 ข้อความ ดังนี้

จำนวนข้อความที่เป็น Positive จริงและทำนายว่าเป็น Positive = 189 ครั้ง
จำนวนข้อความที่เป็น Positive จริงและทำนายว่าเป็นอย่างอื่น = 114 ครั้ง
จำนวนข้อความที่เป็น Negative จริงและทำนายว่าเป็น Negative = 94 ครั้ง
จำนวนข้อความที่เป็น Negative จริงและทำนายว่าเป็นอย่างอื่น = 81 ครั้ง
จำนวนข้อความที่เป็น Neutral จริงและทำนายว่าเป็น Neutral = 100 ครั้ง
จำนวนข้อความที่เป็น Neutral จริงและทำนายว่าเป็นอย่างอื่น = 93 ครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.24 เมทริกซ์ความสับสนแบบจำลองโครงข่ายประสาทหน่วยความจำระยะสั้นแบบยาวด้วยข้อมูลทดสอบ โดยใช้ตัวตัดคำ DeepCut

ผลทำนาย / ผลจริง	Positive	Negative	Neutral	รวม
Positive	189	51	33	273
Negative	84	94	60	238
Neutral	30	30	100	160
รวม	303	175	193	671

จากตารางที่ 4.24 สามารถคำนวณตัววัดประสิทธิภาพทั้ง 4 ตัววัด ดังนี้

1. ค่าความถูกต้อง (Accuracy) = 57.08%
2. ค่าความแม่นยำ (Precision) = 56.79%
3. ค่าความระลึก (Recall) = 57.08%
4. ค่าความถ่วงดุล (F1-Score) = 56.36%

4.5.12 ผลจากแบบจำลองโครงข่ายประสาทหน่วยความจำระยะสั้นแบบยาวแบบสองทิศทาง (Bidirectional LSTM)

4.5.12.1 ผลจากแบบจำลองโครงข่ายประสาทหน่วยความจำระยะสั้นแบบยาวแบบสองทิศทาง (Bidirectional LSTM) โดยใช้ตัวตัดคำ PyThaiNLP

จากผลในการหาพารามิเตอร์ที่ดีที่สุดของไลบรารี่ OptunaSearchCV ค่าพารามิเตอร์ที่ใช้ในการเพิ่มประสิทธิภาพแบบจำลองโครงข่ายประสาทหน่วยความจำระยะสั้นแบบยาวแบบสองทิศทาง คือ units = 128, learning_rate = 0.00017301884422386325, dropout_rate = 0.19908126865254286 epochs = 50 และ batch_size = 32 โดยหลังจากปรับค่าพารามิเตอร์แล้วได้ผลการทดลอบประสิทธิภาพของโครงข่ายประสาทหน่วยความจำระยะสั้นแบบยาวแบบสองทิศทางด้วยข้อมูลทดสอบสามารถแสดงได้ดังตารางที่ 4.25 พบว่าจากความคิดเห็นต่อโครงการบ้านเพื่อคนไทยสำหรับทดสอบ 617 ข้อความ ดังนี้

จำนวนข้อความที่เป็น Positive จริงและทำนายว่าเป็น Positive	=	179	ครั้ง
จำนวนข้อความที่เป็น Positive จริงและทำนายว่าเป็นอย่างอื่น	=	104	ครั้ง
จำนวนข้อความที่เป็น Negative จริงและทำนายว่าเป็น Negative	=	110	ครั้ง
จำนวนข้อความที่เป็น Negative จริงและทำนายว่าเป็นอย่างอื่น	=	109	ครั้ง
จำนวนข้อความที่เป็น Neutral จริงและทำนายว่าเป็น Neutral	=	112	ครั้ง
จำนวนข้อความที่เป็น Neutral จริงและทำนายว่าเป็นอย่างอื่น	=	57	ครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.25 เมทริกซ์ความสับสนแบบจำลองโครงข่ายประสาทหน่วยความจำระยะสั้นแบบยาวแบบสองทิศทางด้วยข้อมูลทดสอบ โดยใช้ตัวตัดคำ PyThaiNLP

ผลทำนาย / ผลจริง	Positive	Negative	Neutral	รวม
Positive	179	70	24	273
Negative	83	110	33	226
Neutral	21	39	112	172
รวม	283	219	169	671

จากตารางที่ 4.25 สามารถคำนวณตัววัดประสิทธิภาพทั้ง 4 ตัววัด ดังนี้

1. ค่าความถูกต้อง (Accuracy) = 59.76%
2. ค่าความแม่นยำ (Precision) = 59.64%
3. ค่าความระลึก (Recall) = 59.76%
4. ค่าความถ่วงดุล (F1-Score) = 59.69%

4.5.12.2 ผลจากแบบจำลองโครงข่ายประสาทหน่วยความจำระยะสั้นแบบยาวแบบสองทิศทาง (Bidirectional LSTM) โดยใช้ตัวตัดคำ DeepCut

จากผลในการหาพารามิเตอร์ที่ดีที่สุดของไลบรารี่ OptunaSearchCV ค่าพารามิเตอร์ที่ใช้ในการเพิ่มประสิทธิภาพแบบจำลองโครงข่ายประสาทหน่วยความจำระยะสั้นแบบยาวแบบสองทิศทาง คือ units = 128, learning_rate = 0.0033656720953600503, dropout_rate = 0.18542877799017332, epochs = 5 และ batch_size = 128 โดยหลังจากปรับค่าพารามิเตอร์แล้วได้ผลการทดลอบประสิทธิภาพของโครงข่ายประสาทหน่วยความจำระยะสั้นแบบยาวแบบสองทิศทางด้วยข้อมูลทดสอบสามารถแสดงได้ดังตารางที่ 4.26 พบว่าจากความคิดเห็นต่อโครงการบ้านเพื่อคนไทยสำหรับทดสอบ 617 ข้อความ ดังนี้

จำนวนข้อความที่เป็น Positive จริงและทำนายว่าเป็น Positive = 190 ครั้ง
จำนวนข้อความที่เป็น Positive จริงและทำนายว่าเป็นอย่างอื่น = 120 ครั้ง
จำนวนข้อความที่เป็น Negative จริงและทำนายว่าเป็น Negative = 99 ครั้ง
จำนวนข้อความที่เป็น Negative จริงและทำนายว่าเป็นอย่างอื่น = 92 ครั้ง
จำนวนข้อความที่เป็น Neutral จริงและทำนายว่าเป็น Neutral = 94 ครั้ง
จำนวนข้อความที่เป็น Neutral จริงและทำนายว่าเป็นอย่างอื่น = 76 ครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.26 เมทริกซ์ความสับสนแบบจำลองโครงข่ายประสาทหน่วยความจำระยะสั้นแบบยาวแบบสองทิศทางด้วยข้อมูลทดสอบ โดยใช้ตัวตัดคำ DeepCut

ผลทำนาย / ผลจริง	Positive	Negative	Neutral	รวม
Positive	190	54	29	273
Negative	92	99	47	238
Neutral	28	38	94	160
รวม	310	191	170	671

จากตารางที่ 4.26 สามารถคำนวณตัววัดประสิทธิภาพทั้ง 4 ตัววัด ดังนี้

1. ค่าความถูกต้อง (Accuracy) = 57.08%
2. ค่าความแม่นยำ (Precision) = 56.51%
3. ค่าความระลึก (Recall) = 57.08%
4. ค่าความถ่วงดุล (F1-Score) = 56.47%

4.5.13 ผลจากแบบจำลองโครงข่ายประสาทเทียมแบบคอนโวลูชัน (Convolutional Neural Network)

4.5.13.1 ผลจากแบบจำลองโครงข่ายประสาทเทียมแบบคอนโวลูชัน (Convolutional Neural Network) โดยใช้ตัวตัดคำ PyThaiNLP

จากผลในการหาพารามิเตอร์ที่ดีที่สุดของไลบรารี่ OptunaSearchCV ค่าพารามิเตอร์ที่ใช้ในการเพิ่มประสิทธิภาพแบบจำลองโครงข่ายประสาทเทียมแบบคอนโวลูชัน คือ filters = 64, kernel_size = 3, learning_rate = 0.00020325945152856084, dropout_rate = 0.11946224922115506, epochs = 5 และbatch_size = 64 โดยหลังจากปรับค่าพารามิเตอร์แล้ว ได้ผลการทดลอบประสิทธิภาพของโครงข่ายประสาทเทียมแบบคอนโวลูชันด้วยข้อมูลทดสอบสามารถแสดงได้ดังตารางที่ 4.27 พบว่าจากความคิดเห็นต่อโครงการบ้านเพื่อคนไทยสำหรับทดสอบ 617 ข้อความ ดังนี้

จำนวนข้อความที่เป็น Positive จริงและทำนายว่าเป็น Positive = 215 ครั้ง
จำนวนข้อความที่เป็น Positive จริงและทำนายว่าเป็นอย่างอื่น = 149 ครั้ง
จำนวนข้อความที่เป็น Negative จริงและทำนายว่าเป็น Negative = 113 ครั้ง
จำนวนข้อความที่เป็น Negative จริงและทำนายว่าเป็นอย่างอื่น = 133 ครั้ง
จำนวนข้อความที่เป็น Neutral จริงและทำนายว่าเป็น Neutral = 45 ครั้ง
จำนวนข้อความที่เป็น Neutral จริงและทำนายว่าเป็นอย่างอื่น = 16 ครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.27 เมทริกซ์ความสับสนแบบจำลองโครงข่ายประสาทเทียมแบบคอนโวลูชันด้วยข้อมูลทดสอบ โดยใช้ตัวตัดคำ PyThaiNLP

ผลทำนาย / ผลจริง	Positive	Negative	Neutral	รวม
Positive	215	53	5	273
Negative	102	113	11	226
Neutral	47	80	45	172
รวม	364	246	61	671

จากตารางที่ 4.27 สามารถคำนวณตัววัดประสิทธิภาพทั้ง 4 ตัววัด ดังนี้

1. ค่าความถูกต้อง (Accuracy) = 55.59%
2. ค่าความแม่นยำ (Precision) = 58.41%
3. ค่าความระลึก (Recall) = 55.59%
4. ค่าความถ่วงดุล (F1-Score) = 53.49%

4.5.13.2 ผลจากแบบจำลองโครงข่ายประสาทเทียมแบบคอนโวลูชัน (Convolutional Neural Network) โดยใช้ตัวตัดคำ DeepCut

จากผลในการหาพารามิเตอร์ที่ดีที่สุดของไลบรารี่ OptunaSearchCV ค่าพารามิเตอร์ที่ใช้ในการเพิ่มประสิทธิภาพแบบจำลองโครงข่ายประสาทเทียมแบบคอนโวลูชัน คือ filters': 64, kernel_size = 3, learning_rate = 0.0008822714279961056, dropout_rate = 0.40426525673778807, epochs = 5 และ batch_size = 64 โดยหลังจากปรับค่าพารามิเตอร์แล้วได้ผลการทดลอบประสิทธิภาพขอโครงข่ายประสาทเทียมแบบคอนโวลูชันข้อมูลทดสอบสามารถแสดงได้ดังตารางที่ 4.28 พบว่าจากความคิดเห็นต่อโครงการบ้านเพื่อคนไทยสำหรับทดสอบ 617 ข้อความ ดังนี้

จำนวนข้อความที่เป็น Positive จริงและทำนายว่าเป็น Positive	=	152	ครั้ง
จำนวนข้อความที่เป็น Positive จริงและทำนายว่าเป็นอย่างอื่น	=	240	ครั้ง
จำนวนข้อความที่เป็น Negative จริงและทำนายว่าเป็น Negative	=	110	ครั้ง
จำนวนข้อความที่เป็น Negative จริงและทำนายว่าเป็นอย่างอื่น	=	129	ครั้ง
จำนวนข้อความที่เป็น Neutral จริงและทำนายว่าเป็น Neutral	=	99	ครั้ง
จำนวนข้อความที่เป็น Neutral จริงและทำนายว่าเป็นอย่างอื่น	=	93	ครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.28 ผลการทดสอบแบบจำลองโครงข่ายประสาทเทียมแบบคอนโวลูชันด้วยข้อมูลทดสอบ โดยใช้ตัวตัดคำ DeepCut

ผลทำนาย / ผลจริง	Positive	Negative	Neutral	รวม
Positive	152	86	35	273
Negative	70	110	58	238
Neutral	18	43	99	160
รวม	240	239	192	671

จากตารางที่ 4.28 สามารถคำนวณตัววัดประสิทธิภาพทั้ง 4 ตัววัด ดังนี้

1. ค่าความถูกต้อง (Accuracy) = 53.80%
2. ค่าความแม่นยำ (Precision) = 54.39%
3. ค่าความระลึก (Recall) = 53.80%
4. ค่าความถ่วงดุล (F1-Score) = 53.11%

4.6 ผลการเปรียบเทียบประสิทธิภาพของแบบจำลอง

จากผลการคำนวณตัววัดประสิทธิภาพของแบบจำลองในหัวข้อ 4.5 นำผลที่ได้มาเปรียบเทียบกันโดยดูจากตัววัดประสิทธิภาพทั้ง 4 ตัว คือ ค่าความถูกต้อง (Accuracy) ค่าความแม่นยำ (Precision) ค่าความระลึก (Recall) และค่าความถ่วงดุล (F1-Score) ได้สรุปผลว่าการใช้ตัวตัดคำ PyThaiNLP กับแบบจำลองซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machine) มีค่าประสิทธิภาพสูงที่สุด อยู่ที่ค่าความถูกต้องร้อยละ 60.06 ค่าความแม่นยำร้อยละ 60.40 ค่าความระลึกร้อยละ 60.06 และค่าความถ่วงดุลร้อยละ 59.94 ตามตารางที่ 4.29 และตารางที่ 4.30

ตารางที่ 4.29 ค่าประสิทธิภาพของแบบจำลองจากการใช้ตัวตัดคำ PyThaiNLP

ลำดับ	แบบจำลอง	Accuracy	Precision	Recall	F1-Score
1	Logistic Regression	59.76	59.50	59.76	59.55
2	Naïve Bayes	55.89	58.27	55.89	53.68
3	KNN	43.07	49.84	43.07	42.42
4	SVM	**60.06**	**60.40**	**60.06**	**59.94**
5	Random Forest	53.06	56.62	53.06	49.99

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.29 ค่าประสิทธิภาพของแบบจำลองจากการใช้ตัวตัดคำ PyThaiNLP (ต่อ)

ลำดับ	แบบจำลอง	Accuracy	Precision	Recall	F1-Score
6	Decision Tree	46.50	51.53	46.50	45.74
7	Multi-Layer Perceptron	56.63	56.40	56.63	56.48
8	Perceptron	48.44	48.41	48.44	47.64
9	Stochastic Gradient Descent	57.97	57.63	57.97	57.73
10	Passive Aggressive	56.63	56.81	56.63	56.27
11	LSTM	56.93	58.54	56.93	56.66
12	BiLSTM	59.76	59.64	59.76	59.69
13	CNN	55.59	58.41	55.59	53.49

ตารางที่ 4.30 ค่าประสิทธิภาพของแบบจำลองจากการใช้ตัวตัดคำ DeepCut

ลำดับ	แบบจำลอง	Accuracy	Precision	Recall	F1-Score
1	Logistic Regression	57.23	56.69	57.23	56.72
2	Naïve Bayes	54.40	56.60	54.40	51.95
3	KNN	45.16	52.92	45.16	45.18
4	SVM	**59.02**	**59.25**	**59.02**	**58.30**
5	Random Forest	55.44	57.16	55.44	52.84
6	Decision Tree	50.37	53.24	50.37	48.66
7	Multi-Layer Perceptron	54.10	54.47	54.10	54.03
8	Perceptron	53.35	53.87	53.35	52.97
9	Stochastic Gradient Descent	58.27	57.80	58.27	56.80
10	Passive Aggressive	55.59	55.75	55.59	55.53
11	LSTM	57.08	56.79	57.08	56.36
12	BiLSTM	57.08	56.51	57.08	56.47
13	CNN	53.80	54.39	53.80	53.11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.7 ผลการนำแบบจำลองไปใช้งาน

จากการนำแบบจำลองที่พัฒนาขึ้นไปทดลองใช้งานอย่างง่ายบน Google Colab

4.7.1 การตรวจสอบข้อความความคิดเห็นจากผู้ใช้งาน

การตรวจสอบข้อความความคิดเห็นของผู้ใช้งาน เป็นการรับค่าข้อความจากผู้ใช้งานแล้วนำข้อความไปจำแนกความรู้สึกต่อโครงการบ้านเพื่อคนไทย

ตัวอย่างของข้อความเชิงบวกที่ป้อนเข้าไป “น่าสนใจ” แบบจำลองจำแนกว่า “Pos” ดังรูปที่ 4.8

```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.svm import SVC
from sklearn.metrics import classification_report
# ✅ ใช้เฉพาะ train set
tfidf_PythaiNL = TfidfVectorizer(tokenizer=lambda x: x, lowercase=False)
X_train_bow = tfidf_PythaiNL.fit_transform(X_train['stopwords_PythaiNLP'])
X_test_bow = tfidf_PythaiNL.transform(X_test['stopwords_PythaiNLP'])
# ✅ สร้างและฝึกโมเดล SVM
model = SVC(C=2.220415159602295, kernel='rbf', gamma='scale')
model.fit(X_train_bow, y_train)
# ✅ ทำนายผลจากชุดทดสอบ
y_pred = model.predict(X_test_bow)
# ✅ ทำนายข้อความใหม่
my_text = input("ข้อความ : ")
my_text = clean_text(my_text)
my_text = punctuation_removal(my_text)
my_text = clean_and_tokenize(my_text)
my_tokens = clean_tokenize_remove_stopwords(my_text)
# ✅ แปลงข้อความใหม่เป็นเวกเตอร์ด้วย vectorizer เดิม
my_bow = tfidf_PythaiNL.transform([" ".join(my_tokens)])
my_prediction = model.predict(my_bow)
# ✅ แสดงผลลัพธ์
print("ผลการทำนาย:", my_prediction[0])
```

```
/usr/local/lib/python3.11/dist-packages/sklearn/feature_extraction/text.py:517: UserWar
  warnings.warn(
ข้อความ : น่าสนใจ
ผลการทำนาย: Pos
```

รูปที่ 4.8 การตรวจสอบข้อความเชิงบวกจากผู้ใช้งาน

ตัวอย่างของข้อความเชิงกลางที่ป้อนเข้าไป “จังหวัดเชียงใหม่มีมั้ย” แบบจำลองจำแนกว่า “Neu” ดังรูปที่ 4.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```python
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.svm import SVC
from sklearn.metrics import classification_report
# ✅ ใช้เฉพาะ train set
tfidf_PythaiNL = TfidfVectorizer(tokenizer=lambda x: x, lowercase=False)
X_train_bow = tfidf_PythaiNL.fit_transform(X_train['stopwords_PythaiNLP'])
X_test_bow = tfidf_PythaiNL.transform(X_test['stopwords_PythaiNLP'])
# ✅ สร้างและฝึกโมเดล SVM
model = SVC(C=2.2204151590602295, kernel='rbf', gamma='scale')
model.fit(X_train_bow, y_train)
# ✅ ทำนายผลจากชุดทดสอบ
y_pred = model.predict(X_test_bow)
# ✅ ทำนายข้อความใหม่
my_text = input("ข้อความ : ")
my_text = clean_text(my_text)
my_text = punctuation_removal(my_text)
my_text = clean_and_tokenize(my_text)
my_tokens = clean_tokenize_remove_stopwords(my_text)
# ✅ แปลงข้อความใหม่เป็นเวกเตอร์ด้วย vectorizer เดิม
my_bow = tfidf_PythaiNL.transform([" ".join(my_tokens)])
my_prediction = model.predict(my_bow)
# ✅ แสดงผลลัพธ์
print("ผลการทำนาย:", my_prediction[0])
```

```
/usr/local/lib/python3.11/dist-packages/sklearn/feature_extraction/text.py:517: UserWar
  warnings.warn(
ข้อความ : จังหวัดเชียงใหม่มีมั้ย
ผลการทำนาย: Neu
```

รูปที่ 4.9 การตรวจสอบข้อความเชิงกลางจากผู้ใช้งาน

ตัวอย่างของข้อความเชิงลบที่ป้อนเข้าไป “โกง” แบบจำลองจำแนกว่า “Neg” ดังรูปที่ 4.10

```python
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.svm import SVC
from sklearn.metrics import classification_report
# ✅ ใช้เฉพาะ train set
tfidf_PythaiNL = TfidfVectorizer(tokenizer=lambda x: x, lowercase=False)
X_train_bow = tfidf_PythaiNL.fit_transform(X_train['stopwords_PythaiNLP'])
X_test_bow = tfidf_PythaiNL.transform(X_test['stopwords_PythaiNLP'])
# ✅ สร้างและฝึกโมเดล SVM
model = SVC(C=2.2204151590602295, kernel='rbf', gamma='scale')
model.fit(X_train_bow, y_train)
# ✅ ทำนายผลจากชุดทดสอบ
y_pred = model.predict(X_test_bow)
# ✅ ทำนายข้อความใหม่
my_text = input("ข้อความ : ")
my_text = clean_text(my_text)
my_text = punctuation_removal(my_text)
my_text = clean_and_tokenize(my_text)
my_tokens = clean_tokenize_remove_stopwords(my_text)
# ✅ แปลงข้อความใหม่เป็นเวกเตอร์ด้วย vectorizer เดิม
my_bow = tfidf_PythaiNL.transform([" ".join(my_tokens)])
my_prediction = model.predict(my_bow)
# ✅ แสดงผลลัพธ์
print("ผลการทำนาย:", my_prediction[0])
```

```
/usr/local/lib/python3.11/dist-packages/sklearn/feature_extraction/text.py:517: UserWar
  warnings.warn(
ข้อความ : โกง
ผลการทำนาย: Neg
```

รูปที่ 4.10 การตรวจสอบข้อความเชิงลบจากผู้ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.8 อภิปรายผล

ผลการทดลองที่ได้ดำเนินการตามกระบวนการวิเคราะห์ความรู้สึกของความคิดเห็นจากยูทูปต่อโครงการบ้านเพื่อคนไทย โดยใช้ข้อมูลที่เก็บรวบรวมระหว่างวันที่ 1 ธันวาคม 2567 ถึง 30 เมษายน 2568 พบว่าแบบจำลองที่ให้ผลลัพธ์ที่ดีที่สุดคือ แบบจำลองซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machine) เมื่อใช้ร่วมกับการตัดคำโดย PyThaiNLP โดยให้ค่าความถูกต้องร้อยละ 60.06 ค่าความแม่นยำร้อยละ 60.40 ค่าความระลึกร้อยละ 60.06 และค่าความถ่วงดุลร้อยละ 59.94 ซึ่งสูงที่สุดในแบบจำลองทั้งหมด 13 แบบจำลอง

จากผลการทดลองพบว่าแบบจำลองซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machine) ให้ค่าความแม่นยำในการวิเคราะห์ความรู้สึกสูงกว่าแบบจำลอง Deep Learning เนื่องจากใช้ข้อมูลที่มีจำนวนจำกัดหรือมีขนาดเล็ก ซึ่งสามารถอธิบายได้จากเหตุผลเชิงทฤษฎีดังต่อไปนี้ ประการแรก แบบจำลอง Deep Learning มักมีโครงสร้างที่ซับซ้อนและมีจำนวนพารามิเตอร์จำนวนมาก ส่งผลให้เกิดปัญหา overfitting ได้ง่ายเมื่อใช้กับข้อมูลขนาดเล็ก กล่าวคือแบบจำลองอาจเรียนรู้ noise จากชุดข้อมูลมากเกินไป แทนที่จะเรียนรู้รูปแบบสำคัญในข้อมูล ทำให้ผลลัพธ์ไม่สามารถนำไปใช้กับข้อมูลใหม่ได้อย่างแม่นยำ (Salman and Liu, 2019) ในทางตรงกันข้ามแบบจำลองซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machine) มีจำนวนพารามิเตอร์น้อยและใช้แนวคิดในการแบ่งเส้นเขตแดนระหว่างคลาสแบบชัดเจน จึงมีความสามารถในการ generalize ได้ดีกว่าในบริบทของข้อมูลน้อย ประการที่สองตามทฤษฎี Bias–Variance Tradeoff แบบจำลองซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machine) มักมีค่า bias สูง แต่มี variance ต่ำ ซึ่งทำให้เสี่ยงต่อการ overfitting น้อยกว่าแบบจำลอง Deep Learning ที่มี variance สูง แม้แบบจำลอง Deep Learning จะสามารถเรียนรู้รูปแบบที่ซับซ้อนได้มากกว่า แต่จะต้องอาศัยข้อมูลจำนวนมากเพื่อสร้างแบบจำลองที่มีเสถียรภาพและแม่นยำ (James, 2011) นอกจากนี้ Machine Learning อย่างแบบจำลองซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machine) ยังสามารถทำงานร่วมกับฟีเจอร์ที่สกัดด้วยวิธีแบบดั้งเดิม เช่น TF-IDF หรือ Bag-of-Words ได้อย่างมีประสิทธิภาพ โดยไม่ต้องอาศัย representation learning เหมือนกับแบบจำลอง Deep Learning ที่ต้องใช้ข้อมูลจำนวนมากเพื่อสกัดฟีเจอร์จาก raw text อย่างมีคุณภาพ

จากปัจจัยทั้งหมดข้างต้นจึงสามารถสรุปได้ว่า ในบริบทของงานวิเคราะห์ความรู้สึกที่มีข้อมูลขนาดเล็ก (small dataset) การเลือกใช้แบบจำลองซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machine หรือแบบจำลอง Machine Learning แบบดั้งเดิมจะเหมาะสมกว่า Deep Learning และให้ผลลัพธ์ที่เสถียรและแม่นยำมากกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผลการดำเนินงานและข้อเสนอแนะ

ในบทนี้จำทำการสรุปผลการดำเนินงานและข้อเสนอแนะ เพื่อใช้สำหรับงานวิจัยในอนาคตแก่ผู้วิจัยหรือผู้ที่ต้องการศึกษาค้นคว้า โดยการสรุปผลอ้างอิงผลการดำเนินงานการค้นคว้าอิสระในบทที่ 4 โดยมีรายละเอียดดังนี้

5.1 สรุปผลการวิจัย

งานวิจัยนี้มีวัตถุประสงค์เพื่อศึกษาและพัฒนาแบบจำลองวิเคราะห์ความรู้สึกของความคิดเห็นต่อโครงการบ้านเพื่อคนไทย โดยใช้ความคิดเห็นจากเว็บไซต์ยูทูป ซึ่งผู้วิจัยทำการรวบรวมข้อมูลบน Google Colab ด้วยภาษาไพธอน ตั้งแต่วันที่ 1 ธันวาคม 2567 จนถึงวันที่ 30 เมษายน 2568 หลังจากได้ข้อมูลมาแล้วก็นำข้อมูลที่รวบรวมได้มาทำการเตรียมข้อมูล การติดฉลากข้อมูล การนอร์มอลไลซ์ตัวอักษร การลบเครื่องหมายวรรคตอนและสัญลักษณ์ การตัดคำ โดยในงานวิจัยนี้จะใช้ตัวตัดคำ 2 แบบ คือ แบบที่ 1 การตัดคำด้วย PyThaiNLP และแบบที่ 2 การตัดคำด้วย DeepCut หลังจากแบ่งการใช้ตัวตัดคำเป็น 2 แบบแล้วก็นำแต่ละแบบมากำจัดคำหยุดและการแปลงคำเป็นเวกเตอร์ตัวเลข เพื่อทำให้ข้อมูลยู่ในรูปแบบที่คอมพิวเตอร์สามารถประมวลผลได้ มีการแบ่งข้อมูลออกเป็น 2 ชุดข้อมุลเรียนรู้ 70% เพื่อใช้ในการพัฒนาแบบจำลองและเรียนรู้ และชุดข้อมูลทดสอบ 30% เพื่อใช้ในการทดสอบประสิทธิภาพของแบบจำลองที่พัฒนา

หลังจากได้ทำการแปลงข้อมูลเป็นเวกเตอร์ด้วยภาษาไพธอนและข้อมูลเรียบร้อยแล้ว ข้อมูลเหล่านี้จะถูกนำมาเข้ากระบวนการฝึกฝนข้อมูลด้วยอัลกอริทึมการจำแนกประเภททั้งหมด 13 แบบ คือ การถดถอยเชิงเส้นโลจิสติค (Logistic Regression) นาอีฟเบย์ (Naïve Bayes) การหาค่าด้วยเพื่อนบ้านใกล้ที่สุด k จุด (K-Nearest Neighbors) ซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machine) ป่าสุ่ม (Random Forest) ต้นไม้ตัดสินใจ (Decision Tree) ตัวจำแนกแบบโครงข่ายประสาทเทียมหลายชั้น (Multi-Layer Perceptron) เพอร์เซปตรอน (Perceptron) สโตแคสติกการ์เดียนดิเซนท์ (Stochastic Gradient Descent) พาสซีฟ อากัสซีฟ (Passive Aggressive) โครงข่ายประสาทหน่วยความจำระยะสั้นแบบยาว(Long Short-Term Memory) โครงข่ายประสาทหน่วยความจำระยะสั้นแบบยาวแบบสองทิศทาง (Bidirectional LSTM) และโครงข่ายประสาทเทียมแบบคอนโวลูชัน (Convolutional Neural Network) หลังจากทำการกระบวนการฝึกฝนข้อมูล แล้วนำมาวัดประสิทธิภาพของแบบจำลองโดยค่าความถูกต้อง (Accuracy) ค่าความแม่นยำ (Precision) ค่าความระลึก (Recall) และค่าความถ่วงดุล (F1-Score) ด้วยชุดข้อมูลทดสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.1 การวิเคราะห์คำในแต่ละประเภทความรู้สึก

การวิเคราะห์ข้อความความคิดเห็นต่อโครงการบ้านเพื่อคนไทย โดยแบ่งออกเป็น 3 กลุ่ม ได้แก่ ความคิดเห็นเชิงบวก ความคิดเห็นเชิงกลางและความคิดเห็นเชิงลบ เพื่อให้สามารถทำความเข้าใจเนื้อหาที่ความสำคัญในแต่ละมุมมองได้อย่างชัดเจน โดยนำข้อมูลความถี่ของคำในแต่ละกลุ่มมาสร้างเป็นภาพ Word Cloud ซึ่งเป็นเครื่องมือที่ช่วยแสดงผลคำที่พบบ่อยในรูปแบบที่เข้าใจง่าย

สรุปผลการวิเคราะห์ถ้อยคำของความคิดเห็นเชิงบวก มีคำปรากฏบ่อยและโด่นเด่นมักเกี่ยวกับคำชื่นชมต่อโครงการบ้านเพื่อคนไทย โดยส่วนใหญ่จะเป็นการชื่นชมเกี่ยวกับทำเลที่ดีของโครงการ มีการเดินทางโทรคมนาคมที่สะดวก ราคาที่ถูกในทำเลที่ที่ดินราคาแพง การชื่นชมและสนับสนุนต่อโครงการของภาครัฐ มีการแสดงถึงการอยากเข้าร่วมโครงการบ้านเพื่อคนไทย

ความคิดเห็นเชิงกลาง คำปรากฏบ่อยมักเกี่ยวกับเชิงพฤติกรรม เช่น “จอง” “ผ่อน” “ขาย” และ “ซื้อ” ซึ่งจากผลสำรวจจากข้อมูลเชิงกลางคำปรากฏบ่อยจะอยู่ในรูปแบบประโยคคำถาม ข้อสงสัย และการรอข้อมูลเพิ่มเติมโดยส่วนใหญ่ เช่น มีขั้นตอนการจองยังไง มีรูปแบบการผ่อนชำระยังไง เงื่อนไขในการซื้อ รูปแบบการขายต่อในอนาคต เนื่องจากทางหน่วยงานภาครัฐอาจให้ข้อมูลที่กำกวมไม่ชัดเจน ทำให้ประชาชนจำนวนหนึ่ง เกิดคำถามและข้อสงสัยต่อโครงการ และต้องการข้อมูลที่ชัดเจนเพิ่มเติม โดยข้อความเหล่านี้สามารถช่วยให้หน่วยงานภาครัฐหรือผู้พัฒนาโครงการ สามารถนำไปข้อความหรือคำถามเหล่านี้ไปสื่อสารกับประชาชนได้มากขึ้น ลดความสงสัยและความกังวนของประชาชนที่มีต่อโครงการบ้านเพื่อคนไทยได้มากขึ้น

ความคิดเห็นเชิงลบ โดยคำที่โดดเด่นจะเกี่ยวกับภาพลักษณ์เชิงลบที่ผู้คนมีต่อโครงการในอดีต นั้นก็คือคำว่า “เอื้ออาทร” ซึ่งจากผลสำรวจจากข้อมูลเชิงลบ คำนี้มักถูกใช้ในบริบทของการเปรียบเทียบกับโครงการในอดีต โดยแสดงความกังวลว่าจะเกิดปัญหาเดิมซ้ำรอย มักจะถูกใช้ในลักษณะเปรียบเปรยถึง ความล้มเหลวของโครงการในอดีต โดยเน้นความทรงจำเชิงลบ เช่น เช่น บ้านร้าง การทุจริต การบริหารผิดพลาด และการไม่ได้ประโยชน์จริงจากโครงการ ประสบการณ์หรือมีความรับรู้ในด้านลบมาก่อน ปัญหาสังคมที่เกิดขึ้นภายหลังการเข้าอยู่อาศัย จึงอาจกล่าวได้ว่าการกล่าวถึง “บ้านเอื้ออาทร” ในที่นี้ สะท้อนถึงความกังวล ความไม่มั่นใจ และการตั้งข้อสังเกตเกี่ยวกับความยั่งยืนของโครงการใหม่ในลักษณะเดียวกัน และขณะเดียวกันประเด็นเรื่องการผ่อนชำระ โดยไม่มีความชัดเจนในกรรมสิทธิ์ เป็นประเด็นที่ถูกวิพากษ์อย่างหนักในเชิงความรู้สึกว่าถูกทำให้เข้าใจผิดและไม่คุ้มค่ากับภาระทางการเงินที่ประชาชนต้องแบกรับในระยะยาว โดยพบคำว่า “ผ่อน” มีความถี่สูงสุดอันดับ 2 จากผลสำรวจจากข้อมูลเชิงลบ และคำนี้มักถูกใช้ในบริบทของการสงสัยสิทธิในการอยู่อาศัยที่กำหนดระยะเวลาการอยู่อาศัย การเป็นเจ้าของ และผลประโยชน์ที่แท้จริงหลังการผ่อนจ่าย ทำให้ประชาชนรู้สึกไม่มั่นใจ ไม่พอใจต่อ ความไม่ชัดเจนของสิทธิในกรรมสิทธิ์ ผู้แสดงความคิดเห็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จำนวนมากรู้สึกว่าตนอาจต้องรับภาระการผ่อนจ่ายในระยะยาว แต่สุดท้ายก็ไม่ได้ครอบครองบ้านจริง ตัวอย่างประโยค เช่น ผ่อนแต่ไม่มีโฉนด เหมือนเช่าอยู่ ผ่อนจบไม่ได้เป็นเจ้าของ จากข้อมูลสะท้อนว่าประชาชนมีทั้งความคาดหวัง ความลังเล และความไม่ไว้วางใจต่อโครงการบ้านเพื่อคนไทย ซึ่งข้อมูลเหล่านี้สามารถช่วยให้หน่วยงานภาครัฐหรือผู้พัฒนาโครงการ สามารถนำไปข้อมูลไปวางแผน และแก้ไขความกังวน ความไม่ชัดเจนต่อข้อมูลโครงการ ความไม่เชื่อมั่นต่อโครงการ และลบภาพจำที่ไม่ดีในอดีตต่อโครงการที่เกี่ยวกับที่อยู่อาศัยของภาครัฐในอนาคตได้มากขึ้น

5.1.2 การเปรียบเทียบผลการจำแนกความคิดเห็นโครงการบ้านเพื่อคนไทยตามตัวตัดคำ

งานวิจัยนี้มีแยกการเปรียบเทียบออกเป็น 2 การเปรียบเทียบ คือ การเปรียบเทียบตัวตัดคำภาษาไทยที่เหมาะสมกับวิธีการจำแนกความคิดเห็นต่อโครงการบ้านเพื่อคนไทยระหว่าง PyThaiNLP กับ DeepCut จากผลค่าความถูกต้อง (Accuracy) ค่าความแม่นยำ (Precision) ค่าความระลึก (Recall) และค่าความถ่วงดุล (F1-Score) โดยสรุปผลการวัดประสิทธิภาพตัวตัดคำจากแบบจำลองทั้ง 13 แบบ ได้ว่า การตัวตัดคำแบบ PyThaiNLP มีประสิทธิภาพดีกว่าการตัวตัดคำแบบ DeepCut โดยแบบจำลองที่ใช้การตัวตัดคำแบบ PyThaiNLP มีค่าความถูกต้องมากกว่าแบบจำลองที่ใช้การตัวตัดคำแบบ DeepCut อยู่ 9 แบบจำลอง คือ นาอีฟเบย์ (Naïve Bayes) การหาค่าด้วยเพื่อนบ้านใกล้ที่สุด k จุด (K-Nearest Neighbors) ซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machine) ป่าสุ่ม (Random Forest) ตัวจำแนกแบบโครงข่ายประสาทเทียมหลายชั้น (Multi-Layer Perceptron) เพอร์เซปตรอน (Perceptron) โครงข่ายประสาทหน่วยความจำระยะสั้นแบบยาว(Long Short-Term Memory) โครงข่ายประสาทหน่วยความจำระยะสั้นแบบยาวแบบสองทิศทาง (Bidirectional LSTM) และโครงข่ายประสาทเทียมแบบคอนโวลูชัน (Convolutional Neural Network) และแบบจำลองที่ใช้การตัวตัดคำ DeepCut มีค่าความถูกต้องมากกว่าแบบจำลองที่ใช้การตัวตัดคำ PyThaiNLP อยู่ 4 แบบจำลอง คือ การถดถอยเชิงเส้นโลจิสติค (Logistic Regression) ต้นไม้ตัดสินใจ (Decision Tree) สโตแคสติกการ์เดียนดิเซนท์ (Stochastic Gradient Descent) พาสซีฟ อากัสซีฟ (Passive Aggressive)

5.1.3 การเปรียบเทียบแบบจำลองการจำแนกความคิดเห็นต่อโครงการบ้านเพื่อคนไทย

การเปรียบเทียบวิธีการจำแนกความคิดเห็นต่อโครงการบ้านเพื่อคนไทยด้วยแบบจำลอง 13 แบบ โดยมีการปรับจูนไฮเปอร์พารามิเตอร์โดยใช้ OptunaSearchCV ค่าที่ดีที่สุดของพารามิเตอร์สรุปผลออกมาได้ว่าแบบจำลองซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machine) โดยใช้ตัวตัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำ PyThaiNLP ให้ผลลัพธ์การจำแนกที่ดีที่สุด โดยมีค่าความถูกต้องร้อยละ 60.06 ค่าความแม่นยำร้อยละ 60.40 ค่าความระลึกร้อยละ 60.06 และค่าความถ่วงดุลร้อยละ 59.94

5.2 ข้อเสนอแนะ

จากผลการดำเนินการค้นคว้าอิสระครั้งนี้ สามารถนำไปเพิ่มประสิทธิภาพแบบจำลองเพื่อการจำแนกความคิดเห็นต่อโครงการบ้านเพื่อคนไทย ดังนี้

1) การขยายขอบเขตของข้อมูลความคิดเห็นต่อโครงการบ้านเพื่อคนไทยเพื่อเพิ่มความน่าเชื่อถือของผลลัพธ์ อาจเพิ่มข้อมูลความคิดเห็นจากแหล่งข้อมูลที่หลากหลายมากขึ้น เช่น Facebook Tiktok
2) การปรับจูนพารามิเตอร์ของแบบจำลองในการหาค่าพารามิเตอร์ที่ดีที่สุด เพื่อช่วยให้แบบจำลองมีประสิทธิภาพที่ดีที่สุด
3) การเปรียบเทียบระหว่างวิธีการตัดคำหลายแบบ เช่น DeepCut, PyThaiNLP และ Wordpiece Tokenizer เพื่อศึกษาเครื่องมือที่เหมาะสมกับแบบจำลองและประสิทธิภาพของแบบจำลอง
4) การใช้การวิเคราะห์ความคิดเห็นต่อโครงการหรือนโยบายต่าง ๆ ของภาครัฐ เพื่อช่วยให้หน่วยงานรัฐของเข้าถึงปัญหาและความต้องการของประชาชนได้ถูกจุดมากขึ้น และสามารถช่วยในการวางแผนการทำงานของหน่วยงานภาครัฐในอนาคตได้
5) ใช้ความคิดเห็นเชิงบวกในการพัฒนาโครงการ หน่วยงานควรใช้ข้อมูลจากคำชื่นชม เช่น ทำเลดี ราคาถูก เป็นจุดแข็งในการประชาสัมพันธ์ และพัฒนาโครงการในพื้นที่ที่มีศักยภาพต่อไป
6) การปรับปรุงการสื่อสารข้อมูล ควรจัดทำข้อมูลเกี่ยวกับขั้นตอน เงื่อนไข และสิทธิให้ชัดเจน เข้าใจง่าย เพื่อแก้ปัญหาความสับสนที่พบในความคิดเห็นเชิงกลาง
7) สร้างความเชื่อมั่นเพื่อลดความกังวล ควรอธิบายความแตกต่างจากโครงการในอดีต และให้ความชัดเจนเรื่องกรรมสิทธิ์ เพื่อลดความไม่ไว้วางใจจากความคิดเห็นเชิงลบ
8) เปิดรับฟังความคิดเห็นต่อเนื่อง ระบบติดตาม ประเมินผลอย่างโปร่งใส และเปิดเผยความคืบหน้าต่อสาธารณะเพื่อสร้างความมั่นใจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

ณัฐิมา อภินาทเมธี. 2567. **การจำแนกแนวโน้มของราคาหุ้นไทยรายวัน โดยใช้การวิเคราะห์ความรู้สึกจากข่าว.** : สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.

บ้านเพื่อคนไทย. 2024. **โครงการบ้านเพื่อคนไทย.** [Online]. Available: https://www.xn--42c2bebcz5a3fxb0cwf7b7bi.th/

บ้านเพื่อคนไทย. 2024. **พื้นที่นำร่องและแผนการดำเนินงานปี 2568–2569.** [Online]. Available: https://www.xn--42c2bebcz5a3fxb0cwf7b7bi.th/

บ้านเพื่อคนไทย. 2024. **แนวคิด TOD และการพัฒนาระบบขนส่ง.** [Online]. Available: https://www.xn--42c2bebcz5a3fxb0cwf7b7bi.th/

บ้านเพื่อคนไทย. 2024. **คุณสมบัติผู้เข้าร่วมโครงการและประเภทที่อยู่อาศัย.** [Online]. Available: https://www.xn--42c2bebcz5a3fxb0cwf7b7bi.th/

พรนิภา ตั้งพิทักษ์เสมอ. 2567. **การทำนายราคาขึ้นลงของหุ้นโดยการวิเคราะห์จากข้อมูลข่าว กรณีศึกษาหุ้นกลุ่มอุตสาหกรรมธุรกิจการเงินของไทย**. : สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.

ภานุมาศ ทุกข์จาก. 2565. **การขุดเหมืองข้อความและการวิเคราะห์ความรู้สึกสำหรับการลงทุน ในสกุลเงินดิจิทัลโดยใช้ความคิดเห็นจากเว็บไซต์คลังคลิปวิดีโอ.** : สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.

Alonso, O. Rose, D. E. and Stewart, B. 2008. **Crowdsourcing for relevance evaluation**. ACM SIGIR Forum, vol. 42, no. 2, pp. 9–15.

Bishop, C. M. 2006. **Pattern Recognition and Machine Learning,** Springer.

Boonkwan, A. 2016. **Comparative evaluation of Thai word segmentation methods for social media texts**. in Proc. 2016 International Conference on Asian Language Processing (IALP), pp. 206–209.

Bottou, L. 2012. **Stochastic gradient descent tricks**. in Neural Networks: Tricks of the Trade, 2nd ed., Springer, pp. 421–436.

Breiman, L. 2001. **Random forests**. Machine Learning, vol. 45, no. 1, pp. 5–32.

Breiman, L. Friedman, J. Olshen, R. and Stone, C. 1984. **Classification and Regression Trees.** Wadsworth International Group.

Brownlee, J. 2020. **Imbalanced Classification with Python: Better Metrics, Balance Skewed Classes, Cost-Sensitive Learning.** Machine Learning Mastery.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Cambria, A. Schuller, B. Xia, Y. and Havasi, C. 2013. **New avenues in opinion mining and sentiment analysis**. IEEE Intelligent Systems, vol. 28, no. 2, pp. 15–21.

Cesa-Bianchi, N. and Lugosi, 2006. **Learning, and Games**, Cambridge University Press.

Cortes, C. and Vapnik, V. 1995. **Support-vector networks**. *Machine Learning*, vol. 20, no. 3, pp. 273–297.

Cover, T. and Hart, P. 1967. **Nearest neighbor pattern classification**. IEEE Transactions on Information Theory, vol. 13, no. 1, pp. 21–27.

Crammer, K. Dekel, O. Keshet J. Shalev-Shwartz, S. andSinger, Y. 2006. **Online passive-aggressive algorithms**. Journal of Machine Learning Research, vol. 7, pp. 551–585.

Dasarathy, S. 1991. **Nearest neighbor (NN) norms: NN pattern classification techniques**. IEEE Computer Society Press.

Devlin, J. Chang, M. Lee, K. and Toutanova, K. 2019**. BERT: Pre-training of deep bidirectional transformers for language understanding**. Conference of the North American Chapter of the Association for Computational Linguistics (NAACL), 2019, pp. 4171–4186.

Fayyad, U. Piatetsky-Shapiro, G. and Smyth, P. 1996. **From data mining to knowledge discovery in databases**. AI Magazine, vol. 17, no. 3, pp. 37–54.

Fawcett, T. 2006. **An introduction to ROC analysis.** Pattern Recognition Letters, vol. 27, no. 8, pp. 861–874.

Feldman, S. and Sanger, D. 2007. **The text mining handbook: Advanced approaches in analyzing unstructured data**. Cambridge University Press.

Goodfellow, I. Bengio, Y. and Courville, A. 2016. **Deep Learning.** MIT Press.

Goldberg , Y. 2016. **A primer on neural network models for natural language processing**. Journal of Artificial Intelligence Research, vol. 57, pp. 345–420.

Graves, A. 2008. **Supervised sequence labelling with recurrent neural networks**. Ph.D. dissertation, Technical University of Munich.

Graves, A. and Schmidhuber, J. 2005. **Framewise phoneme classification with bidirectional LSTM and other neural network architectures**. Neural Networks, vol. 18, no. 5–6, pp. 602–610.

Hagan, M. T. Demuth, H. B. and Beale, M. H. 1996. **Neural Network Design**. PWS Publishing.

Haykin, S. 2009. ***Neural Networks and Learning Machines.*** 3rd ed., Pearson.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Han, J. Kamber, M. and Pei, J. 2011. **Data Mining: Concepts and Techniques**, 3rd ed., San Francisco, CA: Morgan Kaufmann.

Halevy, A. Norvig, P. and Pereira, F. 2009. **The unreasonable effectiveness of data**. IEEE Intelligent Systems, vol. 24, no. 2, pp. 8–12.

Ho, T. K. 1998. **The random subspace method for constructing decision forests.** IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, no. 8, pp. 832–844

Hochreiter, S. and Schmidhuber, J. **Long short-term memory**. Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.

Hosmer, D. W. Lemeshow, S. and Sturdivant, R. X. 2013. **Applied Logistic Regression**. 3rd ed., Wiley.

Jain, A. K. Duin, R. P. W. and Mao, J. 2000. **Statistical pattern recognition: A review**. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 1, pp. 4–37

James, G. Witten, D. Hastie, T. and Tibshirani, R. 2011. **Bias–variance tradeof**f. in *Encyclopedia of Machine Learning*, Springer.

Joachims, T. 1998. **Text categorization with support vector machines: Learning with many relevant features**. in *Proc. 10th European Conference on Machine Learning (ECML)*, pp. 137–142.

Joachims, T. 2006. **Training linear SVMs in linear time**. in Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), pp. 217–226.

Jurafsky, D. and Martin, J. H. 2023. **Speech and Language Processing**. [Online]. Available: https://web.stanford.edu/~jurafsky/slp3/

Karamat, A. and. Farooq, M. 2016. **Emerging role of social media in political activism: Perceptions and practices.** South Asian Studies, vol. 31, no. 1, pp. 381–396.

Kim, Y. 2024. **Convolutional neural networks for sentence classification**. pp. 1746–1751.

Kowsari, A. Jafari Meimandi, K. Heidarysafa, M. et al. 2019. **Text classification algorithms: A survey**. Information, vol. 10, no. 4, p. 150.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Krizhevsky, A. Sutskever, I. and Hinton, G. E. 2012. **ImageNet classification with deep convolutional neural networks**. in Advances in Neural Information Processing Systems (NeurIPS), pp. 1097–1105.

LeCunBottou, Y. L. Orr, G.,and K. Müller. 1998. **Efficient backprop**. in Neural Networks: Tricks of the Trade, Springer, pp. 9–50.

LeCun, Y. Bengio, Y. and Hinton, G. 2015. **Deep learning**. Nature, vol. 521, no. 7553, pp. 436–444.

Lertniphonphun, W. 2019. **A study of Thai sentiment classification using deep learning and PyThaiNLP toolkit**. in Proc. 17th International Conference on ICT and Knowledge Engineering (ICT&KE), pp. 1–6.

Liaw, A. and Wiener, M. 2002. **Classification and regression by randomForest**. R News, vol. 2, no. 3, pp. 18–22.

Liu, B. 2012. **Sentiment Analysis and Opinion Mining.** Morgan & Claypool Publishers.

Liu, B. 2015. **Sentiment Analysis: Mining Opinions, Sentiments, and Emotions**. Cambridge University Press.

Liu, Y. et al. 2016. **Bidirectional LSTM–CRF models for sequence tagging**. arXiv preprint arXiv:1603.01360.

Manning, C. D. Raghavan, P. and Schütze, H. 2008. **Introduction to Information Retrieval**. Cambridge University Press.

Maas, A. L. et al. 2011. **Learning word vectors for sentiment analysis**. in Proc. 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pp. 142–150.

Medhat, A. Hassan, A. and Korashy, H. 2014. **Sentiment analysis algorithms and applications: A survey**. Ain Shams Engineering Journal, vol. 5, no. 4, pp. 1093–1113.

Minsky, M. and Papert. S. 1969. **Perceptrons: An Introduction to Computational Geometry**, MIT Press.

Mikolov, T. Chen, K. Corrado, G. and Dean, J. 2013. **Efficient estimation of word representations in vector space.** *arXiv preprint arXiv:1301.3781*.

Mitchell, T. 1997. **Machine Learning**. McGraw-Hill

Mueller, A. 2015. **Word_cloud: A little word cloud generator in Python**. GitHub repository [Online]. Available: https://github.com/amueller/word_cloud

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Pang, B. and Lee, L. 2008. **Opinion mining and sentiment analysis**. Foundations and Trends in Information Retrieval, vol. 2, no. 1–2, pp. 1–135.

PyThaiNLP Developers. 2024. **DeepCut integration in PyThaiNLP**. GitHub, [Online]. Available: https://github.com/PyThaiNLP/pythainlp

PyThaiNLP Developers. 2024. **PyThaiNLP: Thai natural language processing in Python**. GitHub, [Online]. Available: https://github.com/PyThaiNLP/pythainlp

PyThaiNLP Developers. 2024. **Thai stopword list**. GitHub, [Online]. Available: https://github.com/PyThaiNLP/pythainlp

Quinlan, J. R. 1996. **Improved use of continuous attributes in C4.5**. Journal of Artificial Intelligence Research, vol. 4, pp. 77–90.

Quinlan, J. R. 1986. **Induction of decision trees**. Machine Learning, vol. 1, no. 1, pp. 81–106.

Rajaraman, A. and Ullman, J. D. 2020. **Mining of Massive Datasets**. Cambridge University Press.

Rakwong, C. and Sornlertlamvanich, T. 2019. **DeepCut: A deep learning-based Thai word segmentation**. arXiv preprint arXiv:1905.02023.

Raschka, S. and Mirjalili, V. 2019. **Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow 2.** 3rd ed., Packt Publishing.

Rosenblatt, F. 1958. **The perceptron: A probabilistic model for information storage and organization in the brain**. Psychological Review, vol. 65, no. 6, pp. 386–408.

Rumelhart, D. E. Hinton, G. E. and Williams, R. J. 1986. **Learning internal representations by error propagation.** in Parallel Distributed Processing: Explorations in the Microstructure of Cognition, vol. 1, MIT Press, pp. 318–362.

Sahami, M. Dumais, S. Heckerman, D. and E. Horvitz. 1998. **A Bayesian approach to filtering junk e-mail**. vol. 62, pp. 98–105, AAAI Technical Report.

Salman, S. and Liu, X. 2019.**Overfitting Mechanism and Avoidance in Deep Neural Networks.** *arXiv preprint arXiv:1901.06566*

Schuster, M. and Paliwal, K. K. 1997. **Bidirectional recurrent neural networks**. IEEE Trans. Signal Processing, vol. 45, no. 11, pp. 2673–2681.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Sornlertlamvanich, T. et al. 2019. **Attacut: Fast and accurate Thai word tokenizer**. *arXiv preprint arXiv:1904.09735*.

Sornlertlamvanich, T. et al. 2000. **ORCHID and BEST corpora: Benchmark datasets for Thai word segmentation**. NECTEC.

Sornlertlamvanich, T. Isahara, H. and Charoenporn, V. 1997. **Thai word segmentation based on the longest matching approach**. in Proc. Natural Language Processing Pacific Rim Symposium (NLPRS), pp. 145–150.

Sujiwo, B. Wibowo, A. Saputro, D. 2021. **Sentiment Analysis of Indonesian Government's Policy to Overcome COVID-19 Using SVM, Naive Bayes and LSTM.** International Conference on Data Science and Its Applications (ICoDSA).

Supnithi, T. Srisuk, C. and Sornlertlamvanich, T. 2013. **Lexical normalization for Thai social media text**. in Proc. 2013 10th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), pp. 1–6.

Thelwall, M. Buckley, K. and Paltoglou, G. 2012. **Sentiment strength detection for the social web**. Journal of the American Society for Information Science and Technology, vol. 63, no. 1, pp. 163–173.

Vapnik, V. N. 2000. **The Nature of Statistical Learning Theory.** 2nd ed., Springer.

Viégas, J. and Wattenberg, M. 2008. **Tag clouds and the case for vernacular visualization**. Interactions, vol. 15, no. 4, pp. 49–52.

We Are Social and Hootsuite. 2024. **Digital 2024: Thailand**. [Online]. Available: https://datareportal.com/reports/digital-2024-thailand

Witten, I. H. Frank, E. and Hall, M. A. 2011. **Data Mining: Practical Machine Learning Tools and Techniques**, 3rd ed., Burlington, MA: Morgan Kaufmann.

Ying, X. 2019. **An Overview of Overfitting and its Solutions**. *Journal of Physics: Conference Series*, vol. 1168, no. 2, pp. 022022

Zaidan, A. and Zaidan, C. 2022. **Survey on data labeling for supervised learning: Techniques, challenges and opportunities**. IEEE Access, vol. 10, pp. 57299–57319.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

ชื่อ	นางสาวเขมรัตน ทับแว่ว
วัน เดือน ปีเกิด	20 กันยายน 2540
ที่อยู่ปัจจุบัน	148/80 ซอยคู้บอน 27 แยก 52 ถนนคู้บอน แขวงท่าแร้ง เขตบางเขน 10220
ประวัติการศึกษา	(2562) สาขาวิศวกรรมสารสนเทศ เกรดเฉลี่ย 2.70 สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้