

ระบบคัดกรองผู้ป่วยภาวะซึมเศร้าโดยใช้แบบจำลองการเรียนรู้ของเครื่อง
และการเรียนรู้เชิงลึก

DEPRESSION SCREENING SYSTEM USING DATA-DRIVEN AUDIO BY
MACHINE LEARNING AND DEEP LEARNING MODEL



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิศวกรรมชีวการแพทย์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2566

KMITL-2023-EN-M- 317-164

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DEPRESSION SCREENING SYSTEM USING DATA-DRIVEN AUDIO BY
MACHINE LEARNING AND DEEP LEARNING MODEL



PHANOMKORN HOMSIANG

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
MASTER OF ENGINEERING IN BIOMEDICAL ENGINEERING
SCHOOL OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
2023

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



COPYRIGHT 2023

SCHOOL OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญาานิพนธ์	ระบบคัดกรองผู้ป่วยภาวะซึมเศร้าโดยใช้แบบจำลองการเรียนรู้ของเครื่องและการเรียนรู้เชิงลึก
ชื่อนักศึกษา	นายพนมกร หอมเสียง
รหัสประจำตัว	63601218
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมชีวการแพทย์
พ.ศ.	2566
อาจารย์ที่ปรึกษาปริญญาานิพนธ์	ผศ. ดร. ตรีสุคนธ์ ตรีบุปผชาติสกุล

บทคัดย่อ

งานวิจัยนี้จัดขึ้นเนื่องด้วยการแพร่ระบาดของไวรัสโควิด-19 ตั้งแต่ปี 2562 ทำให้คนไทยมีความเครียดสะสมเนื่องจากความกังวลด้านการติดเชื้อและเศรษฐกิจ จำนวนผู้ป่วยโรคซึมเศร้าในประเทศไทยเพิ่มขึ้น 1-2% และมีแนวโน้มเพิ่มขึ้นอย่างต่อเนื่อง การวินิจฉัยโรคซึมเศร้าสามารถประเมินได้จากการทำแบบประเมิน เช่น PHQ-2, PHQ-8, CES-D, PHQ-9 และ CES-D ที่ใช้ในการปฏิบัติทางคลินิก ซึ่งจะช่วยให้ผู้ทำแบบประเมินภาวะซึมเศร้าและช่วยให้พวกเขาตัดสินใจปรึกษาจิตแพทย์ได้ ปัจจุบันมีการพัฒนาและประยุกต์เทคโนโลยี Machine Learning เพื่อสร้างแบบจำลองการวินิจฉัยทางการแพทย์ การศึกษาวิจัยที่ได้รับการตีพิมพ์จำนวนมากได้แสดงให้เห็นถึงความสำเร็จของการเรียนรู้ของเครื่องและการสร้างการเรียนรู้เชิงลึกสำหรับการจดจำสัญญาณเสียง ดังนั้นผู้วิจัยจึงตัดสินใจศึกษาการสร้างโมเดลการเรียนรู้ของเครื่องและการเรียนรู้เชิงลึกสำหรับการจำแนกภาวะซึมเศร้าโดยใช้ชุดข้อมูลเสียง แบบจำลองเหล่านี้สามารถช่วยจิตแพทย์คัดกรองอาการซึมเศร้าของผู้ป่วยผ่านทางเสียงของพวกเขา ดังนั้น งานวิจัยนี้จึงมีจุดมุ่งหมายเพื่อสร้างโมเดลการเรียนรู้ของเครื่องและการเรียนรู้เชิงลึก ได้แก่ 1D CNN, 2D CNN, LSTM, GRU และ KNN, MLP, Random Forest, Decision Tree, SGD, Logistic Regression, SVM, AdaBoot และ Gaussian แบบจำลอง Naïve Bayes สำหรับแมชชีนเลิร์นนิง สำหรับการจำแนกชุดข้อมูล MFCC ที่แปลงจากคลื่นเสียง ชุดข้อมูลเสียงที่ใช้ในการฝึกอบรมและทดสอบแบบจำลองนี้รวบรวมจากอาสาสมัครทั้งหมด 100 คน ประกอบด้วยผู้ที่มีอาการปกติ 50 คน (ไม่มีความผิดปกติทางจิตภายใต้การดูแลของแพทย์) และผู้ป่วยภาวะซึมเศร้าอีก 50 คนที่อยู่ภายใต้การดูแลของแพทย์ ผู้วิจัยได้สร้างเว็บไซต์แพลตฟอร์มขึ้นมาเพื่อเป็นเครื่องมือในการเก็บรวบรวมชุดข้อมูลเสียงจากการประเมินอาสาสมัครตามการประเมิน PHQ-9

ชุดข้อมูลแบ่งออกเป็น 2 กรณี คือ กรณีที่ 1 สำหรับภาวะซึมเศร้าและไม่ซึมเศร้า ซึ่งเป็นแบบไบนารีคลาส และกรณีที่ 2 สำหรับ 5 ระดับ ได้แก่ ไม่มีอาการ น้อย ปานกลาง มาก และรุนแรง ขึ้นอยู่กับคะแนน PHQ-9 ดัชนีประสิทธิภาพที่ใช้ในการประเมินประสิทธิภาพของโมเดล ได้แก่ Accuracy, Precision, Recall, and F1-score ประสิทธิภาพที่ดีที่สุดของการเรียนรู้ของเครื่องถูกพบบน MLP และ 2D CNN ของโมเดลการเรียนรู้เชิงลึก ซึ่งแสดงความแม่นยำมากกว่า 90% สำหรับทั้งสองโมเดลด้วยความแม่นยำที่ยอมรับได้ Precision, Recall, F1-score มากกว่า 80% สำหรับ ทั้ง 2 คลาส และ 5 คลาสยิ่งไปกว่านั้น ผู้วิจัยยังประสบความสำเร็จในการพัฒนาส่วนติดต่อกับผู้ใช้งาน (User Interface : UI) เพื่ออำนวยความสะดวกให้กับผู้ใช้โดยการนำโมเดลการเรียนรู้เชิงลึกที่ได้รับการฝึกอบรมมาใช้



Thesis	Depression screening system using data-driven audio by machine learning and deep learning model
Student	Mr. Phanomkorn Homsiang
Student ID.	63601218
Degree	Master of Engineering
Program	Biomedical Engineering
Year	2023
Thesis Advisor	Asst. Prof. Dr. Treesukon Treebupachatsakul

ABSTRACT

The pandemic COVID-19 virus since 2019, caused Thai people have stress accumulation because of concerning infection and the economy. The number of depression patients in Thailand increases 1-2% and tends to continually increase. The depression diagnosis can be evaluated based on the questionnaire assessment such as PHQ-2, PHQ-8, PHQ-9, and CES-D that are used in clinical practice. However, the assessments are primarily diagnosed, which helps the assessor evaluate the depression condition and helps them make a decision to consult the psychiatrist. Nowadays, the technology of machine learning has been developed and applied to construct the model for medical diagnosis. Many published research studies have shown the success of machine learning and deep learning construction for voice signal recognition. Therefore, we decided to construct machine learning and deep learning models for depression classification by using an audio dataset. These models could help the psychiatrist primarily screen the condition of depression of patients via their voice. Therefore, this research aims to construct machine learning and deep learning models including implementing 1D CNN, 2D CNN, LSTM, GRU models of deep learning and KNN, MLP, Random Forest, Decision Tree, SGD, Logistic Regression, SVM, AdaBoot, and Gaussian Naïve Bayes models for machine learning, for classifying the depression sound wave audio converted to MFCC datasets. The audio datasets used to train and test the

models were collected from a total of 100 volunteers including 50 people in normal condition (no mental disorder under medical supervision) and another 50 depression patients who are under medical supervision. We created the platform website as a tool for collecting the audio dataset from assessing volunteers based on the PHQ-9 assessment. The dataset was divided into 2 cases; case 1 for the binary class of depression and non-depression and case 2 for 5 classes of non-minimal, mild, moderate, moderately severe, and severe, based on the PHQ-9 score. The performance indexes used to evaluate the model performance include accuracy, precision, recall, and F1-score. The best performance of machine learning was observed on MLP of machine learning and 2D CNN of deep learning model in which the accuracy was shown more than 90% for both models with acceptable precision, recall, and F1-score for almost more than 80% for both 2 cases of binary and 5 classes. Moreover, we achieved the development of the User Interface (UI) to facilitate the user by implementing the 2D CNN-trained model.

กิตติกรรมประกาศ

การพัฒนาระบบคัดกรองผู้ป่วยภาวะซึมเศร้าโดยใช้แบบจำลองการเรียนรู้เชิงลึกด้วยข้อมูลภาพและเสียง มีขั้นตอนในการปฏิบัติงานหลายขั้นตอน ทำให้ผู้จัดทำประสบกับปัญหา และอุปสรรคต่าง ๆ ดังนั้นในการทำปริญาานิพนธ์ฉบับนี้จึงต้องได้รับความช่วยเหลือ และคำแนะนำจากบุคคลหลายท่าน ซึ่งทุกท่านได้ให้ความช่วยเหลือเป็นอย่างดี ทางผู้จัดทำปริญาานิพนธ์จึงอยากขอกราบขอบพระคุณทุกท่านมา ณ โอกาสนี้

ผู้จัดทำขอกราบขอบพระคุณ บิดา มารดา ที่คอยให้กำลังใจในการทำงานเสมอมา และเป็นแรงบันดาลใจสำหรับผู้จัดทำเป็นอย่างมาก

ขอขอบพระคุณผศ. ดร. ตรีสุขนธ์ ตรีบุพชาติสกุล ผศ. ดร. สุวิทย์ ภูมิฤทธิกุลและ ผศ. นพ.คมสันต์ เกียรติรุ่งฤทธิ์ที่ปรึกษาปริญาานิพนธ์ และบุคลากรโรงพยาบาลรามธิบดีทุกท่านร่วมไปถึงอาสาสมัครทุกท่านที่ให้ความร่วมมือจนประสบความสำเร็จไปด้วยดี

นายพนมกร หอมเสียง

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	III
กิตติกรรมประกาศ.....	IV
สารบัญ.....	V
สารบัญภาพ.....	VIII
สารบัญตาราง.....	XIII
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์.....	2
1.3 ขอบเขตของการวิจัย.....	2
1.4 ประโยชน์ของการวิจัย.....	2
1.5 แผนการดำเนินงานปริญญานิพนธ์.....	3
บทที่ 2 แนวคิดและทฤษฎีที่เกี่ยวข้อง.....	4
2.1 Patient Health Questionnaire-9 (PHQ-9).....	4
2.2 ทฤษฎีเสียงสามารถแยกแยะได้.....	5
2.3 ทฤษฎีการจำแนกระดับความรุนแรงของภาวะซึมเศร้า.....	7
2.4 ทฤษฎีการคัดกรองภาวะซึมเศร้าโดยใช้คำพูดโดยอัตโนมัติโดยใช้ โครงข่ายประสาทเทียมแบบลึก.....	10
2.5 ทฤษฎีการแยกแยะภาวะซึมเศร้าด้วยภาษาเกาหลี.....	12
2.6 Mel-Frequency Cepstral Coefficients (MFCC).....	13
2.7 Artificial Intelligence (AI).....	18
2.8 Machine Learning.....	19
2.9 Deep learning.....	20
2.10 Convolutional Neural Network (CNN).....	23
2.11 1D CNN and 2D CNN.....	27
2.12 Recurrent Neural Network (RNN).....	31
2.13 LSTM and GRU.....	32
2.14 Model ทั้ง 10 ของ Machine Learning ที่ใช้ในการทดลอง.....	35

สารบัญ (ต่อ)

		หน้า
	2.15 การประเมินประสิทธิภาพของแบบจำลอง (Evaluate Model)	53
	2.16 Operating Characteristics of Tests or Procedures	56
	2.17 การแปลผลค่า Sensitivity and Specificity	58
บทที่ 3	การดำเนินงาน.....	60
	3.1 จริยธรรมการวิจัยในมนุษย์และการเก็บข้อมูลอาสาสมัคร	60
	3.2 การเก็บข้อมูล	61
	3.3 แบบทดสอบภาวะซึมเศร้า Patient Health Questionnaire – 9 Thai Version (PHQ-9)	63
	3.4 การพัฒนาโปรแกรมเพื่อใช้เป็นแบบทดสอบ PHQ-9 และเก็บข้อมูลของอาสาสมัคร	64
	3.5 วิธีการใช้งานเว็บไซต์และตัวอย่างหน้าตาของเว็บไซต์	70
	3.6 การวัดคุณภาพของ API บนเว็บไซต์	78
	3.7 การพัฒนาโมเดลการเรียนรู้เชิงลึก (Deep Learning)	82
	3.8 การสร้างโมเดลการเรียนรู้ของเครื่อง (Machine Learning)	84
	3.9 ขั้นตอนการพัฒนาแบบจำลองการเรียนรู้เชิงลึก (Deep Learning) และแบบจำลองการเรียนรู้ของเครื่อง (Machine Learning)	86
บทที่ 4	ผลการทดลอง.....	91
	4.1 การเก็บข้อมูลภาพและเสียง	91
	4.2 ผลการทดลองแบบจำลองการเรียนรู้เชิงลึกและการเรียนรู้ของเครื่อง	92
	4.3 การทดลองอื่น ๆ	155
	4.4 ส่วนต่อประสานกับผู้ใช้ (User Interface)	160
บทที่ 5	สรุปผลการศึกษา.....	163
	5.1 สรุปผลของโมเดลแบบจำลองการเรียนรู้เชิงลึกและแบบจำลองการเรียนรู้ของเครื่อง	163
	5.2 สรุปผลการวิจัย	166
	5.3 ข้อจำกัด	168
	5.4 ข้อเสนอแนะ	168

สารบัญ (ต่อ)

	หน้า
บรรณานุกรม.....	168
ภาคผนวก.....	172
ภาคผนวก ก Source Code ของเว็บไซต์ PHQ-9	172
ภาคผนวก ข Source Code ของเว็บไซต์ Deep Learning	198
ภาคผนวก ค Source Code ของเว็บไซต์ Machine Learning	200



สารบัญภาพ

ภาพที่		หน้า
บทที่ 2	แนวคิดและทฤษฎีที่เกี่ยวข้อง.....	4
2.1	ตัวอย่างแบบทดสอบ Patient Health Questionnaire-9	5
2.2	ผลการทดลองของงานวิจัย Speech based Depression Severity Level Classification Using a Multi-Stage Dilated CNN-LSTM Model	9
2.3	สรุปผลการจำแนกประเภทสำหรับสถาปัตยกรรม CNN	12
2.4	สรุปผลการทดลองของงานวิจัย Automatic Depression Detection Using Smartphone Based Text Dependent Speech Signals Deep Convolutional Neural Network Approach	13
2.5	ขั้นตอนในการคำนวณหาค่าสัมประสิทธิ์เซปตริ่มบนสเกลเมล	16
2.6	ชุดตัวกรองฟิลเตอร์แบงค์	17
2.7	ภาพตัวอย่างอัลกอริทึม Deep learning	22
2.8	Convolutional Layer	25
2.9	Max Pooling Layer	25
2.10	Activation Layer	25
2.11	Fully Connected Layer	26
2.12	สถาปัตยกรรม CNN	27
2.13	สถาปัตยกรรมเครือข่ายประสาทเทียม 1D CNN	29
2.14	สถาปัตยกรรมเครือข่ายประสาทเทียม 2D CNN	31
2.15	โครงสร้างของโครงข่ายระบบประสาทแบบย้อนกลับ	32
2.16	สถาปัตยกรรมของ LSTM	34
2.17	สถาปัตยกรรมของ GRU	35
2.18	KNN อัลกอริทึม	37
2.19	Possible hyperplanes	38
2.20	Support Vectors	39
2.21	SVM อัลกอริทึม	39
2.22	Random Forest อัลกอริทึม	41
2.23	Decision Tree อัลกอริทึม	43

VIII

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

ภาพที่		หน้า
2.24	การจำแนกประเภท Gaussian Naive Bayes	44
2.25	Logistic Regression อัลกอริทึม	47
2.26	AdaBoost อัลกอริทึม	49
2.27	Stochastic Gradient Descent อัลกอริทึม	51
2.28	Multi-Layer Perceptron (MLP)	52
2.29	Confusion Matrix Binary Classification	54
บทที่ 3	การดำเนินงาน.....	60
3.1	เอกสารการรับรองผ่านการประเมินจริยธรรมในมนุษย์ คณะแพทยศาสตร์โรงพยาบาลรามาธิบดี มหาวิทยาลัยมหิดล	60
3.2	ตัวอย่างอาสาสมัครที่เก็บข้อมูลในห้องตรวจของภาควิชาจิตเวชศาสตร์ คณะแพทยศาสตร์ โรงพยาบาลรามาธิบดี	61
3.3	เอกสารอนุญาตให้ใช้เครื่องมือ แบบทดสอบภาวะซึมเศร้า Patient Health Questionnaire – 9 Thai Version (PHQ-9)	63
3.4	ตัวอย่าง Source code ของเว็บไซต์ฝั่ง Front-End	65
3.5	ตัวอย่าง Source code ของเว็บไซต์ฝั่ง Back-End	67
3.6	ตัวอย่าง Database ของเว็บไซต์	68
3.7	Server and Network diagram	69
3.8	OS, Hardware and software Specification	69
3.9	IP Assigned	69
3.10	Access Matrix	69
3.11	ตัวอย่าง Windows Server ของเว็บไซต์	70
3.12	แผนภาพการใช้งานระบบ	71
3.13	ตัวอย่าง Login Page	72
3.14	ตัวอย่าง Main Page	72
3.15	ตัวอย่าง Consent Form Page	73
3.16	ตอบแบบสอบถามด้วยการบันทึกวิดีโอ	76
3.17	ยืนยันคำตอบ	76
3.18	หน้าจอคำถามและตอบคำถาม PHQ-9	77

สารบัญภาพ (ต่อ)

ภาพที่		หน้า
3.19	หน้าจอแสดงคำอธิบายเกณฑ์คะแนน	77
3.20	หน้าจอแสดงผลการประเมิน	78
3.21	Workflow Diagram ของรูปแบบที่ 1 ผู้มีภาวะซึมเศร้าและไม่มีภาวะซึมเศร้า	83
3.22	Workflow Diagram ของรูปแบบที่ 2 แบ่งออกเป็น 5 ประเภทตามเกณฑ์ของ PHQ-9	83
3.23	Workflow Diagram ของรูปแบบที่ 1 ผู้มีภาวะซึมเศร้าและไม่มีภาวะซึมเศร้าของ 10 Model	85
3.24	Workflow Diagram ของรูปแบบที่ 2 แบ่งออกเป็น 5 ประเภทตามเกณฑ์ของ PHQ-9 ของ 10 Model	86
3.25	การแปลงคุณสมบัติจาก WAV เป็น MFCC	89
3.26	ตัวอย่างของการบันทึกรูปแบบเสียงคลื่น (WAV)	89
3.27	ตัวอย่างของรูปแบบ MFCC ที่แปลงแล้ว	89
3.28	การแบ่งข้อมูลทั้ง 3 รูปแบบ	90
บทที่ 4	ผลการทดลอง.....	91
4.1	Confusion Matrix ของ Test Result (1D CNN)	100
4.2	Training and test accuracy and loss (1D CNN)	101
4.3	Confusion Matrix ของ Test Result (2D CNN)	102
4.4	Training and test accuracy and loss (2D CNN)	103
4.5	Confusion Matrix ของ Test Result (LSTM)	104
4.6	Training and test accuracy and loss (LSTM)	105
4.7	Confusion Matrix ของ Test Result (GRU)	106
4.8	Training and test accuracy and loss (GRU)	107
4.9	Confusion Matrix ของ Test Result KNN (K Nearest Neighbour) Model	108
4.10	Confusion Matrix ของ Test Result MLP (Multi-Layer Perceptron) Model	109
4.11	Confusion Matrix ของ Test Result Random Forest Model	111

สารบัญภาพ (ต่อ)

ภาพที่		หน้า
4.12	Confusion Matrix ของ Test Result Logistic Regression Model	112
4.13	Confusion Matrix ของ Test Result SVM (Linear) Model	113
4.14	Confusion Matrix ของ Test Result SVM (RBF) Model	114
4.15	Confusion Matrix ของ Test Result Gaussian Naive Bayes Model	115
4.16	Confusion Matrix ของ Test Result Decision Tree Model	116
4.17	Confusion Matrix ของ Test Result AdaBoost Model	117
4.18	Confusion Matrix ของ Test Result Stochastic Gradient Descent (SGD) Model	120
4.19	Confusion Matrix ของ Test Result (1D CNN-5 Class)	132
4.20	Training and test accuracy and loss (1D CNN-5 Class)	133
4.21	Confusion Matrix ของ Test Result (2D CNN-5 Class)	134
4.22	Training and test accuracy and loss (2D CNN-5 Class)	135
4.23	Confusion Matrix ของ Test Result (LSTM-5 Class)	136
4.24	Training and test accuracy and loss (LSTM-5 Class)	137
4.25	Confusion Matrix ของ Test Result (GRU-5 Class)	138
4.26	Training and test accuracy and loss (GRU-5 Class)	139
4.27	Confusion Matrix ของ Test Result KNN Model (5 Class)	140
4.28	Confusion Matrix ของ Test Result MLP Model (5 Class)	142
4.29	Confusion Matrix ของ Test Result Random Forest Model (5 Class)	143
4.30	Confusion Matrix ของ Test Result Logistic Regression Model	145
4.31	Confusion Matrix ของ Test Result SVM (Linear) Model (5 Class)	146
4.32	Confusion Matrix ของ Test Result SVM (RBF) Model (5 Class)	148
4.33	Confusion Matrix ของ Test Result Gaussian Naive Bayes	150
4.34	Confusion Matrix ของ Test Result Decision Tree Model (5 Class)	151
4.35	Confusion Matrix ของ Test Result AdaBoost Model (5 Class)	153

สารบัญภาพ (ต่อ)

ภาพที่		หน้า
4.36	Confusion Matrix ของ Test Result Stochastic Gradient Descent (SGD) Model (5 Class)	154
4.37	ตัวอย่างเว็บไซต์แยกแยะเสียงผู้ที่มีภาวะซึมเศร้าหรือไม่มีภาวะซึมเศร้า	160
4.38	ตัวอย่างผลที่แสดงในเว็บไซต์แยกแยะเสียงผู้ที่มีภาวะซึมเศร้าหรือไม่มีภาวะซึมเศร้า	161
4.39	ตัวอย่างภาพรวมเว็บไซต์แยกแยะเสียงผู้ที่มีภาวะซึมเศร้าหรือไม่มีภาวะซึมเศร้า	161



สารบัญญัตราสาร

ตารางที่		หน้า
บทที่ 1	บทนำ.....	1
1.1	แผนการดำเนินงานปริญญาโท	3
บทที่ 3	การดำเนินงาน.....	60
3.1	คำถาม PHQ-9	74
3.2	คุณภาพของเว็บไซต์ PHQ-9	78
3.3	แบ่งประเภทผู้มีภาวะซึมเศร้าและไม่มีภาวะซึมเศร้าโดยใช้พื้นฐานคะแนนของแบบทดสอบ PHQ-9	87
3.4	แบ่งกลุ่มผู้มีภาวะซึมเศร้า 5 ระดับโดยใช้เกณฑ์คะแนนของแบบทดสอบ PHQ-9	87
บทที่ 4	ผลการทดลอง.....	91
4.1	ตารางแสดงจำนวนการเก็บข้อมูลอาสาสมัครประเภทที่ 1	91
4.2	ตารางแสดงจำนวนการเก็บข้อมูลอาสาสมัครประเภทที่ 2	91
4.3	ตารางแสดงจำนวนการเก็บข้อมูลอาสาสมัครแบ่งตามเพศ	91
4.4	ตารางแสดงจำนวนการเก็บข้อมูลอาสาสมัครประเภทที่ 1 แบบ Balance ข้อมูล	92
4.5	ตารางแสดงจำนวนการแบ่งข้อมูล Training set = 60% 2. Validation set = 20% 3. Unseen Test set = 20%	92
4.6	ผลของ Training Result, Validation Result, Test Result แสดงค่า Precision, Recall, F1-Score และ Accuracy ของการจำแนกประเภทระหว่างผู้มีภาวะซึมเศร้าและไม่มีภาวะซึมเศร้าของทั้ง 4 Model (1D CNN, 2D CNN, LSTM, GRU)	93
4.7	ผลของ Training Result, Validation Result, Test Result แสดงค่า Precision, Recall, F1-Score และ Accuracy ของการจำแนกประเภทระหว่างผู้มีภาวะซึมเศร้าและไม่มีภาวะซึมเศร้าของทั้ง 10 Model (KNN, MLP, Random Forest, Logistic Regression, SVM (Linear), SVM (RBF), GNB, Decision Tree, AdaBoost, SGD)	96
4.8	เปรียบเทียบผลรวมค่าความแม่นยำของประเภทที่ 1 การจำแนกประเภทระหว่างผู้มีภาวะซึมเศร้าและไม่มีภาวะซึมเศร้าระหว่างแบบจำลองการเรียนรู้เชิงลึกและการเรียนรู้ของเครื่อง	98

สารบัญตาราง (ต่อ)

ตารางที่		หน้า
4.9	Sensitivity และ Specificity (1D CNN)	101
4.10	Sensitivity และ Specificity (2D CNN)	103
4.11	Sensitivity และ Specificity (LSTM)	105
4.12	Sensitivity และ Specificity (GRU)	107
4.13	Sensitivity และ Specificity KNN (K Nearest Neighbour)	108
4.14	Sensitivity และ Specificity MLP (Multi-Layer Perceptron)	110
4.15	Sensitivity และ Specificity (Random Forest)	111
4.16	Sensitivity และ Specificity (Logistic Regression)	112
4.17	Sensitivity และ Specificity (SVM (Linear))	113
4.18	Sensitivity และ Specificity (SVM (RBF))	115
4.19	Sensitivity และ Specificity (Gaussian Naive Bayes)	116
4.20	Sensitivity และ Specificity (Decision Tree)	117
4.21	Sensitivity และ Specificity (AdaBoost)	119
4.22	Sensitivity และ Specificity (SGD)	120
4.23	ตารางแสดงจำนวนการเก็บข้อมูลอาสาสมัครประเภทที่ 2 แบบไม่ Balance ข้อมูล	121
4.24	ตารางแสดงจำนวนการแบ่งข้อมูล Training set = 60% 2. Validation set = 20% 3. Unseen Test set = 20%	121
4.25	ผลของ Training Result, Validation Result, Test Result แสดงค่า Precision, Recall, F1-Score และ Accuracy ของการจำแนกประเภทที่แบ่งออกเป็น 5 ประเภทตามเกณฑ์ของ PHQ-9 ได้แก่ 1. ไม่มีอาการ 2. น้อย 3. ปานกลาง 4. มาก 5. รุนแรงของทั้ง 4 Model การเรียนรู้เชิงลึก	122
4.26	ผลของ Training Result, Validation Result, Test Result แสดงค่า Precision, Recall, F1-Score และ Accuracy ของการจำแนกประเภทที่แบ่งออกเป็น 5 ประเภทตามเกณฑ์ของ PHQ-9 ได้แก่ 1. ไม่มีอาการ 2. น้อย 3. ปานกลาง 4. มาก 5. รุนแรงของทั้ง 10 Model การเรียนรู้ของเครื่อง	126

สารบัญตาราง (ต่อ)

ตารางที่		หน้า
4.27	เปรียบเทียบผลรวมค่าความแม่นยำของประเภทที่ 2 การจำแนกประเภทที่แบ่งออกเป็น 5 ประเภทตามเกณฑ์ของ PHQ-9 ได้แก่ 1. ไม่มีอาการ 2. น้อย 3. ปานกลาง 4. มาก 5. รุนแรง	130
4.28	Sensitivity และ Specificity (1D CNN-5 Class)	133
4.29	Sensitivity และ Specificity (2D CNN-5 Class)	135
4.30	Sensitivity และ Specificity (LSTM-5 Class)	137
4.31	Sensitivity และ Specificity (GRU-5 Class)	139
4.32	Sensitivity และ Specificity (KNN-5 Class)	140
4.33	Sensitivity และ Specificity (MLP-5 Class)	142
4.34	Sensitivity และ Specificity (Random Forest-5 Class)	144
4.35	Sensitivity และ Specificity (Logistic Regression-5 Class)	145
4.36	Sensitivity และ Specificity (SVM (Linear-5 Class))	147
4.37	Sensitivity และ Specificity (SVM (RBF-5 Class))	148
4.38	Sensitivity และ Specificity (Gaussian Naive Bayes-5 Class)	150
4.39	Sensitivity และ Specificity (Decision Tree-5 Class)	152
4.40	Sensitivity และ Specificity (AdaBoost-5 Class)	153
4.41	Sensitivity และ Specificity (Stochastic Gradient Descent-5 Class)	155
4.42	ตารางแสดงจำนวนของอาสาสมัครแบ่งตามเพศที่ใช้ในการทดลอง	155
4.43	Sum Result แบ่งตามเพศ	156
4.44	Sum Result ของการทดลองที่ 1	156
4.45	Sum Result ของการทดลองที่ 2	157
4.46	Sum Result ของการทดลองที่ 3	157
4.47	Sum Result ของการทดลองที่ 4	157
4.48	Sum Result ของการทดลองที่ 5	158
4.49	Sum Result ของการทดลองที่ 6	158
4.50	สรุปผลการทดลองทั้ง 6 การทดลอง	158
4.51	รวมฐานข้อมูล PHQ-9 และ Daic-Woz	159

สารบัญตาราง (ต่อ)

ตารางที่		หน้า
4.52	ผลการทดลองการรวมฐานข้อมูลรูปแบบที่ 1 รวม Dataset ภาษาไทย และภาษาอังกฤษ (Dataset PHQ-9 + Dataset DAIC-WOZ)	160
4.53	ผลการทดลองการรวมฐานข้อมูลรูปแบบที่ 2 เฉพาะภาษาไทยอย่างเดียว (Dataset PHQ-9)	160
4.54	ผลการทดลองการรวมฐานข้อมูลรูปแบบที่ 3 เฉพาะภาษาอังกฤษอย่างเดียว (Dataset DAIC-WOZ)	160
บทที่ 5	สรุปผลการศึกษา.....	201
5.1	ค่าความแม่นยำของ Case 1 ที่มีอาสาสมัคร 26 คนและ Case 2 ที่มีอาสาสมัคร 30 คน	164
5.2	ค่าความแม่นยำของ Case 1 ที่มีอาสาสมัคร 26 คนและ Case 2 ที่มีอาสาสมัคร 30 คน	165
5.3	ผลการทดลองการรวมฐานข้อมูล PHQ-9 และ Daic-Woz ของทั้ง 3 รูปแบบ	165
5.4	สรุปประสิทธิภาพของโมเดล CNN, LSTM, GRU สำหรับชุดข้อมูล SOUND WAVE	167

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

เนื่องด้วยสถานการณ์ปัจจุบันที่มีการแพร่ระบาดของอย่างหนักจากเชื้อไวรัส Covid-19 [1] ที่เริ่มมาตั้งแต่เดือนธันวาคมปี ค.ศ 2019 ก่อให้เกิดความกังวลกับประชาชนไทย ทั้งความกังวลจากการติดโรคและความกังวลจากผลกระทบทางเศรษฐกิจและสังคมที่จะตามมา จึงทำให้คนไทยเกิดสภาวะเครียดสะสมและมีความเสี่ยงเป็นโรคซึมเศร้าได้ แม้ว่าปัจจุบันจะมีแบบคัดกรองภาวะซึมเศร้าต่าง ๆ เช่น PHQ-9, CES-D, Q2, Q9 ซึ่งประชาชนสามารถเข้าถึงและคัดกรองภาวะซึมเศร้าด้วยตนเองผ่านอินเทอร์เน็ต [2] แม้ว่าเครื่องมือดังกล่าวจะมีความน่าเชื่อถือ (Reliability) และความแม่นยำ (Validity) อยู่ในเกณฑ์ที่ดีสามารถนำมาใช้ในทางคลินิกได้ แต่อย่างไรก็ตามการประเมินดังกล่าวมักมีอคติ (Bias) ที่เกิดจากการรายงานด้วยตัวผู้ประเมินเอง และเป็นข้อมูลที่เป็นเชิงจิตวิสัย (Subjective) ดังนั้นการมีเครื่องมือที่ปราศจากอคติดังกล่าว มีข้อมูลที่มีความเป็นวัตถุวิสัย (Objective) เช่นการใช้การวิเคราะห์อารมณ์ผ่านปัญญาประดิษฐ์ (Artificial Intelligence; AI) จึงได้รับความสนใจมากยิ่งขึ้น [3]

จากปัญหาข้างต้นที่กล่าวมาคณะผู้จัดทำจึงพัฒนาเครื่องมือช่วยระบุภาวะซึมเศร้าโดยใช้เสียงโดยผู้วิจัยได้พัฒนาเครื่องมือเพื่อจัดเก็บข้อมูลภาพและเสียงของอาสาสมัคร เพื่อนำมาเป็น Dataset ที่จำเป็นสำหรับการทำโมเดลการเรียนรู้เชิงลึกและการเรียนรู้ของเครื่องสำหรับคัดกรองผู้ป่วยโรคซึมเศร้า โดยระบบหรือเครื่องมือที่ใช้สำหรับเก็บข้อมูลภาพและเสียงจะอยู่ในรูปแบบของเว็บไซต์โดยใช้แบบทดสอบ PHQ-9 เป็นแบบประเมินเพื่อจัดกลุ่มเสียงตามคะแนนการทำแบบทดสอบ PHQ-9 และนำ Dataset ที่เก็บรวบรวมนำมาประยุกต์ใช้กับเทคโนโลยีการเรียนรู้เชิงลึก (Deep Learning) และการเรียนรู้ของเครื่อง (Machine Learning) ดังนั้นงานวิจัยนี้ถูกจัดทำขึ้นเพื่อสร้างแบบจำลองการเรียนรู้เชิงลึกและการเรียนรู้ของเครื่องที่สามารถแยกแยะหรือระบุภาวะซึมเศร้าจากข้อมูลเสียงของประชากรไทย เพื่อใช้ในการคัดกรองและจำแนกผู้ป่วยภาวะซึมเศร้าและไม่มีภาวะซึมเศร้าและแยกแยะระดับตามเกณฑ์ผลการประเมินของ PHQ-9 ได้แก่ 1. ไม่มีอาการซึมเศร้า 2. น้อย 3. ปานกลาง 4. มาก 5. รุนแรง

1.2 วัตถุประสงค์

1. พัฒนาระบบหรือเครื่องมือเพื่อเก็บข้อมูล (Dataset) ภาพและเสียงในรูปแบบ Website
2. พัฒนาแบบจำลองการเรียนรู้เชิงลึก (Deep learning) และการเรียนรู้ของเครื่อง (Machine Learning) เพื่อคัดกรองผู้ป่วยที่มีภาวะโรคซึมเศร้า ซึ่งข้อมูลจะถูกแบ่งเป็น 2 กลุ่มคือ ผู้ป่วยภาวะซึมเศร้า และ ผู้ไม่มีภาวะซึมเศร้า หรือนำข้อมูลแบ่งเป็น 5 กลุ่มตามเกณฑ์ผลการประเมิน PHQ-9 ได้แก่ 1. ไม่มีอาการซึมเศร้า 2. น้อย 3. ปานกลาง 4. มาก 5. รุนแรง โดยใช้ข้อมูลภาพสัญญาณของเสียงชนิด MFCC
3. พัฒนาส่วนติดต่อกับผู้ใช้ (User Interface : UI) ในการคัดกรองผู้ป่วยภาวะซึมเศร้าและไม่มีภาวะซึมเศร้า

1.3 ขอบเขตของการวิจัย

โครงการนี้ได้พัฒนาระบบที่ใช้สำหรับเก็บข้อมูลภาพและเสียงจะอยู่ในรูปแบบของเว็บไซต์ โดยใช้แบบทดสอบ PHQ-9 เป็นแบบประเมินเพื่อเก็บ Dataset จากนั้นนำ Dataset ที่รวบรวมได้มาทดลองกับแบบจำลองการเรียนรู้เชิงลึก (Deep learning) และ การเรียนรู้ของเครื่อง (Machine Learning)

1.4 ประโยชน์ของการวิจัย

1. เว็บไซต์แบบทดสอบ PHQ-9 ที่บอกระดับภาวะซึมเศร้าของอาสาสมัคร
2. Machine learning Model ในการแยกแยะผู้ป่วยภาวะซึมเศร้าและไม่มีภาวะซึมเศร้า รวมถึงแยกแยะระดับของผู้ป่วยภาวะซึมเศร้าตามเกณฑ์ PHQ-9
3. Deep learning Model ในการแยกแยะผู้ป่วยภาวะซึมเศร้าและไม่มีภาวะซึมเศร้า รวมถึงแยกแยะระดับของผู้ป่วยภาวะซึมเศร้าตามเกณฑ์ PHQ-9

1.5 แผนการดำเนินงานปริญญาโท

ตารางที่ 1.1 แผนการดำเนินงานปริญญาโท

ที่	แผนการ	2565			2566								
		พ.ย	ธ.ค	ม.ค	ก.พ	มี.ค	เม.ย	พ.ย	มิ.ย	ก.ค	ส.ค	ก.ย	
1	ศึกษางานวิจัยที่เกี่ยวข้อง และเขียนโครงร่างงานวิจัย												
2	ค้นหาข้อมูลซอฟต์แวร์และวิธีเขียนโปรแกรม												
3	เขียนโปรแกรม												
4	ทดสอบโปรแกรม												
5	เก็บข้อมูลอาสาสมัคร												
6	พัฒนาโมเดลการเรียนรู้เชิงลึกและการเรียนรู้ของเครื่องโดยใช้ข้อมูลเสียง												
7	วิเคราะห์ผลที่ได้สรุปผลข้อมูล												
8	ทำรูปเล่ม												

บทที่ 2

แนวคิดทฤษฎีที่เกี่ยวข้อง

การพัฒนาระบบคัดกรองผู้ป่วยภาวะซึมเศร้าโดยใช้แบบจำลองการเรียนรู้เชิงลึกด้วยข้อมูลเสียงได้ศึกษาแนวคิดทฤษฎี และงานวิจัยที่เกี่ยวข้องเพื่อนำมาพัฒนาระบบ โดยมีรายละเอียดดังต่อไปนี้

2.1 Patient Health Questionnaire-9 (PHQ-9)

แบบสอบถามสุขภาพผู้ป่วย PHQ-9 [4] เป็นเครื่องมือสำหรับการวินิจฉัยตามเกณฑ์ของโรคซึมเศร้า ที่พบได้บ่อยในการดูแลเบื้องต้นความถูกต้องในการวินิจฉัยของ PHQ-9 เพิ่งได้รับการจัดตั้งขึ้นในการศึกษาที่เกี่ยวข้องกับผู้ป่วย 3,000 รายใน primary care clinics ทั้งหมด 8 แห่ง และผู้ป่วย 3,000 รายในคลินิกสูติ-นรีเวชวิทยา 7 แห่ง [4] โดยคำถามทั้งหมด 9 รายการที่ประเมินภาวะซึมเศร้าของ PHQ-9 โดย PHQ-9 มีศักยภาพในการเป็นเครื่องมือแบบสองวัตถุประสงค์คือสามารถวินิจฉัยโรคซึมเศร้าและระดับความรุนแรงของอาการซึมเศร้าได้ [4]

แบบสอบถามภาวะซึมเศร้า PHQ-9 เป็นแบบสอบถามเกี่ยวกับอาการโรคซึมเศร้าในช่วง 2 สัปดาห์ที่ผ่านมา มีจำนวน 9 ข้อ แต่ละข้อมี 4 ตัวเลือกได้แก่ ไม่เลย (0 คะแนน) มีบางวันไม่บ่อย (1 คะแนน) มีค่อนข้างบ่อย (2 คะแนน) มีเกือบทุกวัน (3 คะแนน) การแปลผลแบ่งออกเป็นสองแบบคือวินิจฉัยโรคซึมเศร้าและระดับความรุนแรงของอาการซึมเศร้าโดยคะแนนมีเกณฑ์ดังต่อไปนี้

แบบที่ 1 วินิจฉัยโรคซึมเศร้า

- คะแนน 0 – 14 หมายถึง ไม่มีอาการซึมเศร้า
- คะแนน 15 – 27 หมายถึง ผู้ป่วยภาวะซึมเศร้า

แบบที่ 2 ระดับความรุนแรงของอาการซึมเศร้า

- คะแนน 0 – 4 หมายถึง อยู่ในระดับไม่มีอาการซึมเศร้า
- คะแนน 5 – 8 หมายถึง มีภาวะซึมเศร้าอยู่ในระดับน้อย
- คะแนน 9 – 14 หมายถึง มีภาวะซึมเศร้าอยู่ในระดับปานกลาง
- คะแนน 15 – 19 หมายถึง มีภาวะซึมเศร้าอยู่ในระดับมาก
- คะแนน 20 – 27 หมายถึง มีภาวะซึมเศร้าอยู่ในระดับรุนแรง

แบบทดสอบภาวะซึมเศร้า PHQ-9				
** ในช่วง 2 สัปดาห์ ที่ผ่านมา ท่านมีอาการดังต่อไปนี้บ่อยแค่ไหน? **				
โปรดใส่คะแนนให้ตรงกับคำตอบของท่าน	เกณฑ์ให้คะแนน			
	ไม่เลย = 0, มีบางวันหรือไม่บ่อย = 1, มีค่อนข้างบ่อย = 2, มีเกือบทุกวัน = 3			
	ไม่เลย	มีบางวันหรือไม่	มีค่อนข้างบ่อย	มีเกือบทุกวัน
1. เบื่อทำอะไร ๆ ก็ไม่เพลิดเพลิน				
2. ไม่สบายใจ ซึมเศร้า หรือท้อแท้				
3. หลับยาก หรือหลับ ๆ ตื่น ๆ หรือหลับมากไป				
4. เหนื่อยง่าย หรือไม่ค่อยมีแรง				
5. เบื่ออาหาร หรือกินมากเกินไป				
6. รู้สึกไม่ดีกับตัวเอง คิดว่าตัวเองล้มเหลวหรือเป็นคนทำให้ตัวเอง หรือครอบครัวผิดหวัง				
7. สมาธิไม่ดีเวลาทำอะไร เช่น ดูโทรทัศน์ ฟังวิทยุ หรือทำงานที่ต้องใช้ความตั้งใจ				
8. พูดหรือทำอะไรซ้ำจนคนอื่นมองเห็น หรือกระสับกระส่ายจนท่านอยู่ไม่นิ่งเหมือนเคย				
9. คิดทำร้ายตนเอง หรือคิดว่าถ้าตาย ๆ ไปเสียคงจะดี				

ภาพที่ 2.1 ตัวอย่างแบบทดสอบ Patient Health Questionnaire-9

2.2 ทฤษฎีเสียงสามารถแยกแยะได้

ในงานวิจัยของ SRI International, Menlo Park, CA, USA. [5]

ความเป็นมา: งานวิจัยนี้เกิดขึ้นเนื่องจากความสำคัญของการตรวจจับภาวะซึมเศร้าจากเสียงพูดที่เพิ่มขึ้นในปัจจุบัน การวิเคราะห์เสียงพูดเพื่อตรวจจับภาวะซึมเศร้ามีความสามารถที่จะช่วยให้ผู้ประสบภาวะซึมเศร้าได้รับการวินิจฉัยและการรักษาที่เหมาะสมได้เร็วขึ้น และงานวิจัยในอดีตมีการศึกษาเกี่ยวกับการวิเคราะห์เสียงพูดเพื่อตรวจจับภาวะซึมเศร้าอย่างมากมาย แต่งานวิจัยเหล่านั้นมักใช้เงื่อนไขที่เสียบสงบหรือไม่มีรบกวนเสียงหรือการสะท้อนเสียง ซึ่งไม่สอดคล้องกับสภาวะที่เกิดขึ้นในชีวิตประจำวัน นอกจากนี้ยังมีความจำเป็นที่จะศึกษาผลกระทบของเสียงรบกวนและการสะท้อนเสียงต่อการทำนายภาวะซึมเศร้า และการศึกษาเกี่ยวกับความทนทานของระบบการทำนายภาวะซึมเศร้าในสภาวะที่ไม่เหมือนกัน งานวิจัยนี้จึงเกิดขึ้นเพื่อศึกษาและวิเคราะห์ผลกระทบของเสียง

รบกวนและการสะท้อนเสียงต่อการทำนายภาวะซิมเคร้าจากเสียงพูด และเพื่อหาวิธีการที่สามารถทำให้ระบบการทำนายภาวะซิมเคร้ามีความทนทานและสามารถใช้งานได้ในสถานะที่ไม่เจียบสงบ

วัตถุประสงค์: การศึกษาและวิเคราะห์ผลกระทบของเสียงรบกวนและการสะท้อนเสียงต่อการทำนายภาวะซิมเคร้าจากเสียงพูด โดยมีวัตถุประสงค์หลักคือการทดสอบความทนทานของระบบการทำนายภาวะซิมเคร้าในสถานะที่ไม่เหมือนกัน และการหาวิธีการที่สามารถเพิ่มความทนทานและประสิทธิภาพของระบบในการทำนายภาวะซิมเคร้าได้ในสถานะที่ไม่เหมือนกัน

วิธีการ: มีขั้นตอนการดำเนินงานดังนี้

- สร้างชุดข้อมูลโดยใช้ชุดข้อมูล AVEC-2014 ที่ประกอบด้วยวิดีโอและเสียงของผู้ประสบภาวะซิมเคร้า โดยแบ่งข้อมูลออกเป็นชุดฝึกฝน ชุดพัฒนา และชุดทดสอบ
- สกัดลักษณะเสียงโดยใช้เทคนิคการสกัดลักษณะเสียงเช่น mel-frequency cepstral coefficients (MFCCs) และ Damped Oscillator Cepstral Coefficients (DOCCs) เพื่อใช้ในการทำนายภาวะซิมเคร้า
- สร้างโมเดลการทำนายโดยใช้โมเดลการทำนายเช่น Support Vector Regression (SVR) หรือ Artificial Neural Networks (ANNs) เพื่อทำนายคะแนนภาวะซิมเคร้าจากลักษณะเสียงที่สกัดมา
- สร้างโมเดลการทำนาย: ใช้โมเดลการทำนายเช่น Support Vector Regression (SVR) หรือ Artificial Neural Networks (ANNs) เพื่อทำนายคะแนนภาวะซิมเคร้าจากลักษณะเสียงที่สกัดมา

ผลลัพธ์: พบว่าการใช้ลักษณะเสียงที่ทนทานต่อเสียงรบกวนและการสะท้อนเสียง (เช่น DOCCs) มีประสิทธิภาพที่ดีกว่าในการทำนายภาวะซิมเคร้า โดยเฉพาะในสถานะที่มีเสียงรบกวนและการสะท้อนเสียง นอกจากนี้ยังพบว่าการใช้โมเดลการทำนายแบบ Artificial Neural Networks (ANNs) มีประสิทธิภาพที่ดีกว่า Support Vector Regression (SVR) ในการทำนายภาวะซิมเคร้า นอกจากนี้ยังพบว่าเสียงพูดแบบ spontaneous speech ให้ประสิทธิภาพที่ดีกว่า read speech ในการทำนายภาวะซิมเคร้า และพบว่าเสียงรบกวนมีผลกระทบต่อประสิทธิภาพการทำนายมากกว่าการสะท้อนเสียง นอกจากนี้ยังพบว่าการทำนายภาวะซิมเคร้าในสถานะที่ไม่เหมือนกัน (cross-corpus) มีประสิทธิภาพที่ดีกว่าการทำนายในสถานะที่เหมือนกัน (in-corpus) โดยเฉพาะในเงื่อนไขของเสียงรบกวนและการสะท้อนเสียง

สรุป: การทำนายภาวะซิมเคร้าจากเสียงพูดในงานวิจัยนี้ได้ดังนี้

- การใช้คุณลักษณะเสียงที่มีความทนทานต่อสภาวะเสียงรบกวนและการสะท้อนเสียง เช่น Damped Oscillator Cepstral Coefficients (DOCCs) มีประสิทธิภาพที่ดีกว่า Mel-Frequency Cepstral Coefficients (MFCCs) ในการทำนายภาวะซิมเสร้า
- การใช้โมเดล Artificial Neural Networks (ANNs) ในการทำนายมีประสิทธิภาพที่ดีกว่า Support Vector Regression (SVR)
- การใช้เสียงพูดแบบ spontaneous speech ให้ประสิทธิภาพที่ดีกว่า read speech ในการทำนายภาวะซิมเสร้า
- การทำนายภาวะซิมเสร้าในสภาวะที่ไม่เหมือนกัน (cross-corpus) มีประสิทธิภาพที่ดีกว่าการทำนายในสภาวะที่เหมือนกัน (in-corpus) โดยเฉพาะในเงื่อนไขของเสียงรบกวนและการสะท้อนเสียง

อย่างไรก็ตามงานวิจัยนี้ไม่ได้ระบุค่าความแม่นยำของการทำนายภาวะซิมเสร้า ดังนั้นจึงไม่สามารถสรุปผลการทำนายเป็นตัวเลขได้โดยตรง

2.3 ทฤษฎีการจำแนกระดับความรุนแรงของภาวะซิมเสร้า

ในงานวิจัยของ Nadee Seneviratne, Carol Espy-Wilson [6]

ความเป็นมา: งานวิจัยนี้เกี่ยวกับการจำแนกระดับความรุนแรงของภาวะซิมเสร้าโดยใช้เสียงพูดเป็นตัวแบบ งานวิจัยก่อนหน้านี้มักเน้นการจำแนกเฉพาะระหว่างผู้ที่มีภาวะซิมเสร้าและผู้ที่ไม่ มีภาวะซิมเสร้าเท่านั้น แต่งานวิจัยนี้ได้นำเสนอการจำแนกระดับความรุนแรงของภาวะซิมเสร้าเป็น ปัญหาการจำแนก เพื่อให้ได้ข้อมูลที่มีความละเอียดมากขึ้นในผลลัพธ์ของการจำแนก ในงานวิจัยนี้ นักวิจัยใช้ข้อมูลเกี่ยวกับการปรับปรุงการประสานงานของระบบปากกับการพูด (articulatory coordination) ที่เกิดขึ้นเมื่อเกิดภาวะซิมเสร้า โดยใช้ข้อมูลเสียงพูดที่ได้จากการวัดตัวแปรต่างๆ ของ ทางเสียง (vocal tract variables) และนำมาสร้างโมเดลการจำแนกโดยใช้เทคนิคของ Convolutional Neural Network (CNN) และ Long Short-Term Memory (LSTM) Recurrent Neural Network (RNN)

วัตถุประสงค์: การจำแนกระดับความรุนแรงของภาวะซิมเสร้าโดยใช้เสียงพูดเป็นตัวแบบ โดยเน้นการให้ข้อมูลที่มีความละเอียดและความแม่นยำมากขึ้นในการจำแนก ซึ่งจะช่วยให้สามารถ ระบุระดับความรุนแรงของภาวะซิมเสร้าได้อย่างมีประสิทธิภาพ นอกจากนี้ยังมีวัตถุประสงค์ที่ เกี่ยวข้องกับการพัฒนาโมเดลการจำแนกที่สามารถใช้งานได้ในสภาวะที่ข้อมูลมีขนาดใหญ่และมีความ

ซับซ้อน และการนำเสนอเทคนิคการใช้งานโมเดลแบบหลายระดับ (Multi-Stage Model) ที่ช่วยเพิ่มประสิทธิภาพในการจำแนกของโมเดลในระดับสูงขึ้นไป

วิธีการ: ประกอบด้วยขั้นตอนต่อไปนี้

- เตรียมข้อมูลโดยนำข้อมูลเสียงพูดที่เกี่ยวข้องกับภาวะซึมเศร้ามาจากฐานข้อมูลที่มีอยู่ โดยแบ่งข้อมูลออกเป็นชุดฝึกฝน (Training Set), ชุดตรวจสอบ (Validation set), และ ชุดทดสอบ (Test set) โดยให้แต่ละชุดมีการกระจายคลาดเคลื่อนที่คล้ายกันและมีจำนวนตัวอย่างในแต่ละกลุ่มคล้ายกัน
- การสกัดลักษณะเชิงลึก (Deep Feature Extraction): ใช้เทคนิคการสกัดลักษณะเชิงลึก เช่น Mel Frequency Cepstral Coefficients (MFCCs), Formants, และ Extended Geneva Minimalistic Acoustic Parameter Set (eGeMAPS) เพื่อแปลงเสียงพูดให้เป็นลักษณะเชิงลึกที่สามารถนำมาใช้ในการจำแนกได้
- การสร้างโมเดลการจำแนก: ใช้โมเดล Convolutional Neural Network (CNN) ที่มีการใช้งานร่วมกับ Dilated Convolutional Layers เพื่อจำแนกลักษณะเชิงลึกของเสียงพูด และใช้โมเดล Long Short-Term Memory (LSTM) Recurrent Neural Network (RNN) เพื่อจำแนกระดับความรุนแรงของภาวะซึมเศร้าในระดับของเซสชัน
- การฝึกโมเดลโดยนำข้อมูลในชุดฝึกฝนมาฝึกโมเดลโดยใช้วิธีการคำนวณค่าความคลาดเคลื่อนระหว่างผลลัพธ์จากโมเดลกับค่าเป้าหมาย และใช้วิธีการปรับค่าพารามิเตอร์ของโมเดลเพื่อให้ได้ผลลัพธ์ที่ดีที่สุด
- ACFs (Articulatory Coordination Features) คือคุณลักษณะทางเสียงที่ถูกพัฒนาขึ้นเพื่อจำแนกภาวะซึมเศร้า โดยใช้ข้อมูลเกี่ยวกับการปรับปรุงการประสานงานของระบบปากเสียง (Neuromotor Coordination) ที่เกิดขึ้นเมื่อมีการลดความเร็วในการเคลื่อนไหว (Psychomotor Slowing) ซึ่งเป็นลักษณะที่จำเป็นในภาวะซึมเศร้าระดับสำคัญ (Major Depressive Disorder)
- ACFs ที่ได้มาจากข้อมูลเสียงจากทางเสียงของช่องปาก (Vocal Tract Variables) ถูกนำมาใช้ในการฝึกโมเดลการจำแนกภาวะซึมเศร้าด้วยโครงข่ายประสาทเชิงลึกแบบ Convolutional Neural Network (CNN) ที่มีการขยายตัว (dilated) และโมเดลการจำแนกภาวะซึมเศร้าระดับเซสชันด้วยโครงข่ายประสาทเชิงลึกแบบ Long Short-Term Memory (LSTM)

สรุป: งานวิจัยนี้เป็นการวิเคราะห์การใช้เสียงพูดในการจำแนกภาวะซึมเศร้า โดยใช้โมเดล Multi-Stage Dilated CNN-LSTM ที่ใช้ ACFs จาก TVs เป็นข้อมูลเข้าสู่โมเดล โดยโมเดลนี้สามารถทำนายความรุนแรงของภาวะซึมเศร้าได้ทั้งแบบ segment-level และ session-level โดยมีผลลัพท์การทดสอบแสดงให้เห็นถึงความแม่นยำของโมเดลในการจำแนกภาวะซึมเศร้าในระดับต่าง ๆ ด้วย Unweighted Average Recall (UAR) และ F1 score โดยโมเดลที่ใช้ ACFs จาก TVs มีประสิทธิภาพดีกว่า ACFs จาก Mel Freq และโมเดลนี้สามารถนำไปพัฒนาเป็นระบบ multi-modal ที่ใช้ข้อมูลจาก Automatic Speech Recognition tools และ linguistic features เพื่อเพิ่มประสิทธิภาพในการจำแนกภาวะซึมเศร้าได้อีกด้วย

2.4 ทฤษฎีการคัดกรองภาวะซึมเศร้าโดยใช้คำพูดโดยอัตโนมัติโดยใช้โครงข่ายประสาทเทียมแบบลึก

ในงานวิจัยของ Karol Chlasta, Krzysztof Wotk, Izabela Krejtz [7]

ความเป็นมา: งานวิจัยนี้เกี่ยวกับการพัฒนาวิธีการตรวจสอบภาวะซึมเศร้าโดยใช้เทคนิคการเรียนรู้เชิงลึกแบบ Convolutional Neural Networks (CNN) และการฝึกอบรมแบบ multipart interactive training โดยใช้ข้อมูลเสียงเป็นตัวแปรในการวิเคราะห์ งานวิจัยนี้นำเสนอผลการทดลองที่ใช้ข้อมูลเสียงจากบุคคลที่ไม่มีภาวะซึมเศร้าและบุคคลที่มีภาวะซึมเศร้า โดยใช้โมเดล CNN และการสร้างสเปกโตรแกรม (spectrogram) จากข้อมูลเสียง ผลการทดลองแสดงให้เห็นถึงความสามารถในการจำแนกภาวะซึมเศร้าอย่างมีประสิทธิภาพ โดยมีความแม่นยำในการจำแนกสูงถึง 78% ซึ่งเป็นผลลัพธ์ที่มีความสำคัญในการพัฒนาเทคนิคการคัดกรองภาวะซึมเศร้าในระดับเบื้องต้น

วัตถุประสงค์: การพัฒนาวิธีการตรวจสอบภาวะซึมเศร้าอัตโนมัติโดยใช้เสียงพูด โดยมีวัตถุประสงค์หลักคือการพัฒนาวิธีการตรวจสอบภาวะซึมเศร้าที่มีความแม่นยำและประสิทธิภาพสูง ที่สามารถใช้งานได้ง่ายและสะดวกสำหรับผู้ใช้งาน โดยเฉพาะอย่างยิ่งในกรณีที่ผู้ที่ต้องการความช่วยเหลือด้านสุขภาพจิตมีความเร่งด่วนในการได้รับความช่วยเหลือ

วิธีการ: ขั้นตอนการทำงานสำหรับการฝึกอบรมโมเดลประกอบด้วยการเรียนรู้แบบส่วนประกอบหลายส่วน (multipart interactive training) ที่ใช้กระบวนการฝึกอบรมแบบวนซ้ำเพื่อปรับแต่งและปรับปรุงโมเดลให้มีประสิทธิภาพสูงสุด โดยใช้การปรับแต่งอัตราการเรียนรู้ (learning rate) และจำนวนรอบการฝึกอบรม (epochs) เพื่อให้โมเดลสามารถจำแนกภาวะซึมเศร้าได้อย่างแม่นยำและมีประสิทธิภาพสูงสุด การทดลองใช้โมเดล ResNet ในงานวิจัยนี้ได้แสดงให้เห็นถึง

ความสามารถในการจำแนกภาวะซึมเศร้าอย่างมีประสิทธิภาพ โดยมีความแม่นยำในการจำแนกสูงถึง 78% โมเดล ResNet ที่ใช้ในการทดลองได้แก่ ResNet-18, ResNet-34, ResNet-50, ResNet-101, และ ResNet-152 ที่ประกอบไปด้วยขั้นตอนต่างๆ ดังนี้

- การเตรียมข้อมูลโดยข้อมูลเสียงจากบุคคลที่ไม่มีภาวะซึมเศร้าและบุคคลที่มีภาวะซึมเศร้าถูกตัดแบ่งเป็นช่วงเสียงยาว 15 วินาที และถูกแปลงเป็นสเปกโตรแกรม (spectrogram) ที่มีขนาด 224x224 พิกเซล
- การฝึกอบรมโมเดลโดยใช้โมเดล Convolutional Neural Networks (CNN) ที่ถูกฝึกอบรมล่วงหน้า (pre-trained) โดยใช้ชุดข้อมูลที่เตรียมไว้ โดยการฝึกอบรมนี้ใช้วิธี fine-tuning โดยแทนที่เลเยอร์สุดท้ายของโมเดลด้วยเลเยอร์ที่เหมาะสมสำหรับงานวิจัยนี้
- การปรับแต่งพารามิเตอร์โดยปรับแต่งพารามิเตอร์ของโมเดลเพื่อให้เหมาะสมกับชุดข้อมูลที่ใช้ โดยการปรับแต่งนี้ใช้เทคนิค cyclic stochastic gradient descent with restart (SGDR) เพื่อลดอัตราการเรียนรู้ของโมเดลเมื่อเข้าสู่ขั้นตอนการฝึกอบรม
- การประเมินผลโดยนำโมเดลที่ได้ฝึกอบรมมาทดสอบกับชุดข้อมูลทดสอบ เพื่อประเมินความแม่นยำในการจำแนกภาวะซึมเศร้า

ผลลัพธ์: ผลลัพธ์ของงานวิจัยนี้แสดงให้เห็นถึงความสามารถในการจำแนกภาวะซึมเศร้าด้วยความแม่นยำที่สูงถึง 78% โดยใช้โมเดลประสาทเชิงลึกแบบคอนโวลูชันแนล (Convolutional Neural Networks) และข้อมูลเสียงจากบุคคลที่ไม่มีภาวะซึมเศร้าและบุคคลที่มีภาวะซึมเศร้าในกระบวนการฝึกอบรม ผลลัพธ์นี้เป็นการยืนยันว่าวิธีการที่นำเสนอมีความสามารถในการตรวจสอบภาวะซึมเศร้าอย่างมีประสิทธิภาพและสามารถนำไปใช้ในการคัดกรองภาวะซึมเศร้าในระดับเบื้องต้นได้ โดยทั้งหมดนี้สามารถใช้ในการจำแนกภาวะซึมเศร้าได้ แต่โมเดล ResNet-34 และ ResNet-50 ได้แสดงผลที่ดีที่สุด โดยมีความแม่นยำในการจำแนกสูงถึง 71% และสามารถเพิ่มความแม่นยำในการจำแนกได้ถึง 77% โดยใช้เทคนิค Test Time Augmentation (TTA) ดังที่แสดงในภาพที่ 2.3

Model	Type of Input	Hyperparameters (LR; EP)	Accuracy	F1 Score	Precision	Recall
ResNet 18	Image (224x224px)	0.01; 3	78%	0.0000	-	0.0000
ResNet 34	Image (224x224px)	0.001; 3	81%	0.6154	0.5714	0.6667
ResNet 50	Image (224x224px)	0.01; 3	67%	0.3077	0.3333	0.2857
ResNet 50	Image (512x512px)	0.01; 3	78%	0.5714	0.5714	0.5714
ResNet 101	Image (512x512px)	0.001; 4	81%	0.2857	0.2500	0.3333
ResNet 101	Image (1024x1024px)	0.0044; 4	78%	0.4000	0.2500	1.0000
ResNet 152	Image (512x512px)	0.0001; 3	63%	0.3750	0.3750	0.3750
ResNet 152	Image (1024x1024px)	0.003; 3	74%	0.5333	0.5000	0.5714

ภาพที่ 2.3 สรุปผลการจำแนกประเภทสำหรับสถาปัตยกรรม CNN [7]

สรุป: งานวิจัยนี้เสนอวิธีการตรวจสอบภาวะซึมเศร้าในเสียงด้วยโมเดล Convolutional Neural Networks (CNN) และสเปกโตรแกรม โดยการฝึกอบรมโมเดล CNN ด้วยข้อมูลเสียงที่แบ่งเป็นช่วงเสียงยาว 15 วินาที และแปลงเป็นสเปกโตรแกรม ซึ่งได้ผลลัพธ์ที่มีความแม่นยำในการจำแนกภาวะซึมเศร้าสูงถึง 78% โดยใช้เทคนิคการปรับแต่งพารามิเตอร์และการฝึกอบรมที่เหมาะสมกับชุดข้อมูลที่ใช้ ผลลัพธ์ที่ได้จากงานวิจัยนี้สามารถนำไปใช้ในการคัดกรองภาวะซึมเศร้าในระดับเบื้องต้นได้ วิธีการนี้มีความสามารถในการช่วยเหลือทั้งผู้ป่วยและผู้ให้การดูแลด้านสุขภาพจิตในการตรวจสอบภาวะซึมเศร้าอย่างรวดเร็วและมีประสิทธิภาพ

2.5 ทฤษฎีการแยกแยะภาวะซึมเศร้าด้วยภาษาเกาหลี

ในงานวิจัยของ Ah Young Kim, MSc; Eun Hye Jang, PhD; Seung-Hwan Lee, MD, PhD; Kwang-Yeon Choi, MD, PhD; Jeon Gue Park, PhD; Hyun-Chool Shin, PhD [8]

ความเป็นมา: การวินิจฉัยภาวะซึมเศร้าโดยอัตโนมัติจากคำพูดสามารถเสริมการรักษาสุขภาพจิตในอนาคตได้ การศึกษาก่อนหน้านี้ได้รายงานว่าคุณสมบัติทางเสียงสามารถใช้เพื่อระบุภาวะซึมเศร้าได้ อย่างไรก็ตาม มีงานวิจัยเพียงไม่กี่ชิ้นที่พยายามวินิจฉัยแยกโรคของผู้ป่วยโรคซึมเศร้าโดยใช้ลักษณะเสียงของผู้พูดที่ไม่ใช่ภาษาอังกฤษ

วัตถุประสงค์: การศึกษานี้เสนอกรอบการทำงานสำหรับการตรวจจับภาวะซึมเศร้าโดยอัตโนมัติโดยใช้ลักษณะเสียงขนาดใหญ่ตามภาษาเกาหลี

วิธีการ: คัดเลือกผู้ป่วย 153 รายที่เข้าเกณฑ์สำหรับโรคซึมเศร้า และ 165 รายที่ควบคุมสุขภาพได้โดยไม่มีอาการป่วยทางจิตในปัจจุบันหรือในอดีต เสียงของผู้เข้าร่วมถูกบันทึกไว้ในสมาร์ทโฟนขณะดำเนินการอ่านประโยคแบบข้อความที่กำหนดไว้ล่วงหน้า สามวิธีได้รับการประเมินและเปรียบเทียบเพื่อตรวจจับภาวะซึมเศร้าโดยใช้ชุดข้อมูลที่มิงานอ่านออกเสียงข้อความ: โมเดลแมช

ซินเลิร์นนิ่งแบบเดิมที่อิงตามคุณสมบัติทางเสียง แบบจำลองที่เสนอซึ่งฝึกและจำแนกสเปกตรัมของลึอกเมลโดยใช้โครงข่ายประสาทเทียมแบบลึก (CNN) ด้วยพารามิเตอร์จำนวนค่อนข้างน้อย และแบบจำลองที่ฝึกฝนและจัดประเภทสเปกตรัมของลึอกเมลโดยใช้เครือข่ายที่ได้รับการฝึกฝนมาเป็นอย่างดี

ผลลัพธ์: ลักษณะทางเสียงของการอ่านประโยคแบบข้อความที่กำหนดไว้ล่วงหน้าตรวจพบภาวะซึมเศร้าโดยอัตโนมัติโดยใช้แบบจำลอง CNN ที่เสนอ ความแม่นยำสูงสุดที่ CNN เสนอให้กับข้อมูลเสียงพูดคือ 78.14% ผลลัพธ์ของแสดงให้เห็นว่าคุณลักษณะทางเสียงที่เรียนรู้ได้อย่างลึกซึ้งนำไปสู่ประสิทธิภาพที่ดีกว่าที่ได้รับจากวิธีการทั่วไปและแบบจำลองที่ได้รับการฝึกอบรมล่วงหน้า

สรุป: การตรวจสอบอารมณ์ของผู้ป่วยโรคซึมเศร้าและการตรวจสอบความสอดคล้องของคำอธิบายวัตถุประสงค์เป็นหัวข้อการวิจัยที่สำคัญมาก การศึกษานี้ชี้ให้เห็นว่าการวิเคราะห์ข้อมูลเสียงพูดที่บันทึกในขณะที่อ่านประโยคที่ขึ้นอยู่กับข้อความสามารถช่วยทำนายสถานะภาวะซึมเศร้าได้โดยอัตโนมัติโดยการจับลักษณะของภาวะซึมเศร้า วิธีการที่ใช้สมาร์ทโฟน เข้าถึงได้ง่าย และสามารถช่วยในการระบุภาวะซึมเศร้าได้โดยอัตโนมัติ ดังภาพที่ 2.4

Models	Data sets		Text-dependent setting (mean of 10-fold)				Text-independent setting (single fold)			
	ACC ^a (%)	F1-score ^b (%)	CCC ^c	RMSE ^d	ACC (%)	F1-score (%)	CCC	RMSE		
	Proposed CNNs model	78.14	77.27	0.28	9.21	56.82	37.84	0.287	5.53	
GCNN-LSTM ^e [61]	51.65	50.90	0.43	8.10	58.57	39.78	0.497	5.70		

ภาพที่ 2.4 สรุปผลการทดลองของงานวิจัย Automatic Depression Detection Using Smartphone Based Text Dependent Speech Signals Deep Convolutional Neural Network Approach [8]

2.6 Mel-Frequency Cepstral Coefficients (MFCC)

MFCC หมายถึง "Mel-Frequency Cepstral Coefficients" [9] ซึ่งเป็นรูปแบบของคุณลักษณะเสียง (audio feature) ที่ใช้ในการวิเคราะห์เสียงและประมวลผลเสียงเพื่อการแยกแยะและจัดการกับข้อมูลเสียง การประมวลผลเสียงของ MFCC มีหลายรูปแบบเช่น การรู้จำเสียงพูด (speech recognition), การจัดกลุ่มเสียง (speaker clustering) และอื่น ๆ ที่เกี่ยวข้องกับการวิเคราะห์เสียงพูด

MFCC ถูกใช้ในการสร้างระบบการรู้จำเสียงพูดและการแยกแยะคำพูด เนื่องจากคุณลักษณะนี้ช่วยลดขนาดข้อมูลเสียงและเน้นที่คุณลักษณะที่สำคัญในการแยกแยะเสียงและคำพูด ซึ่งเป็นเทคนิคที่มีประสิทธิภาพอย่างมาก รวมถึงในการจัดการกับข้อมูลเสียงที่ซับซ้อน

2.6.1 การสกัดลักษณะเด่น (feature extraction)

การสกัดลักษณะเด่น เป็นการดึงลักษณะเฉพาะของหน่วยเสียงแต่ละหน่วยเสียง ที่แตกต่างกันออกมา และระบบทำการรู้จำลักษณะเด่นของหน่วยเสียงแต่ละหน่วยเสียง เมื่อสัญญาณที่เข้ามาภายหลัง มีลักษณะเด่นที่เหมือนหรือใกล้เคียงกับลักษณะเด่นของหน่วยเสียงระบบรู้จำจะสามารถบอกได้ว่าเป็นหน่วยเสียงกลุ่มใด หรือใกล้เคียงกับหน่วยเสียงกลุ่มใดมากที่สุด และสามารถลดจำนวนข้อมูล โดยที่ข้อมูลจำนวนมากจะถูกแปลงเป็นชุดข้อมูลที่มีจำนวนน้อยลง และยังคงคุณสมบัติสำคัญของข้อมูลเดิมไว้ได้อย่างถูกต้อง [9]

2.6.2 ขั้นตอนในการคำนวณค่าสัมประสิทธิ์เซปตรัมบนสเกล Mel-Frequency Cepstral Coefficients (MFCC)

การคำนวณ MFCC (Mel-Frequency Cepstral Coefficients) เป็นกระบวนการที่ใช้ในการแปลงสัญญาณเสียงเป็นคุณลักษณะที่เหมาะสมสำหรับการวิเคราะห์เสียงและเสียงพูด ระบบ MFCC มีลักษณะเด่นในการจัดลำดับความสำคัญของคุณลักษณะเสียงและคำนวณด้วยการใช้การแปลงทางคณิตศาสตร์แบบฟูริเยร์ (Fast Fourier Transform, FFT) และการปรับสเกล Mel ซึ่งจะแปลงสเปกตรัมความถี่ของสัญญาณเสียงให้อยู่ในรูปแบบ Mel-frequency scale และจากนั้นใช้การทำการลอกซ์เกอร์ (cepstral lifter) เพื่อควบคุมความถี่และลดความซับซ้อนของข้อมูลเสียง

ขั้นตอนในการคำนวณ MFCC มีดังต่อไปนี้

1. บันทึกเสียงให้อยู่ในรูปแบบคลื่น (Waveform)

บันทึกเสียงเป็นข้อมูลแบบดิจิทัล (Digital Audio Waveform) โดยใช้ไมโครโฟนหรือเครื่องบันทึกเสียงอื่น ๆ ข้อมูลนี้จะประกอบด้วยค่าความถี่สูงสุดสูง (sampling frequency) ที่สมควร

2. แบ่งสัญญาณเสียงเป็นกรอบเวลา (Frames)

แบ่งสัญญาณเสียงออกเป็นช่วงเวลาสั้น ๆ ที่เรียกว่า "กรอบเวลา (frames)" โดยแต่ละกรอบเวลามักมีความยาวประมาณ 20-30 มิลลิวินาทีและมีการเลื่อนช่วงเวลา (overlap) ระหว่างกรอบ

3. คำนวณสเปกตรัม Mel (Mel Spectrogram)

ใช้กรอบเวลาที่แบ่งมาในขั้นตอนที่ 2 เพื่อคำนวณสเปกตรัมของกรอบเวลาแต่ละอันโดยใช้การแปลงทางคณิตศาสตร์เช่น การแปลงทางฟูรีเยร์ (Fast Fourier Transform, FFT) เพื่อเปลี่ยนสัญญาณเวลาให้เป็นสเปกตรัมของความถี่ Mel ที่เหมาะสมสำหรับการวิเคราะห์เสียง

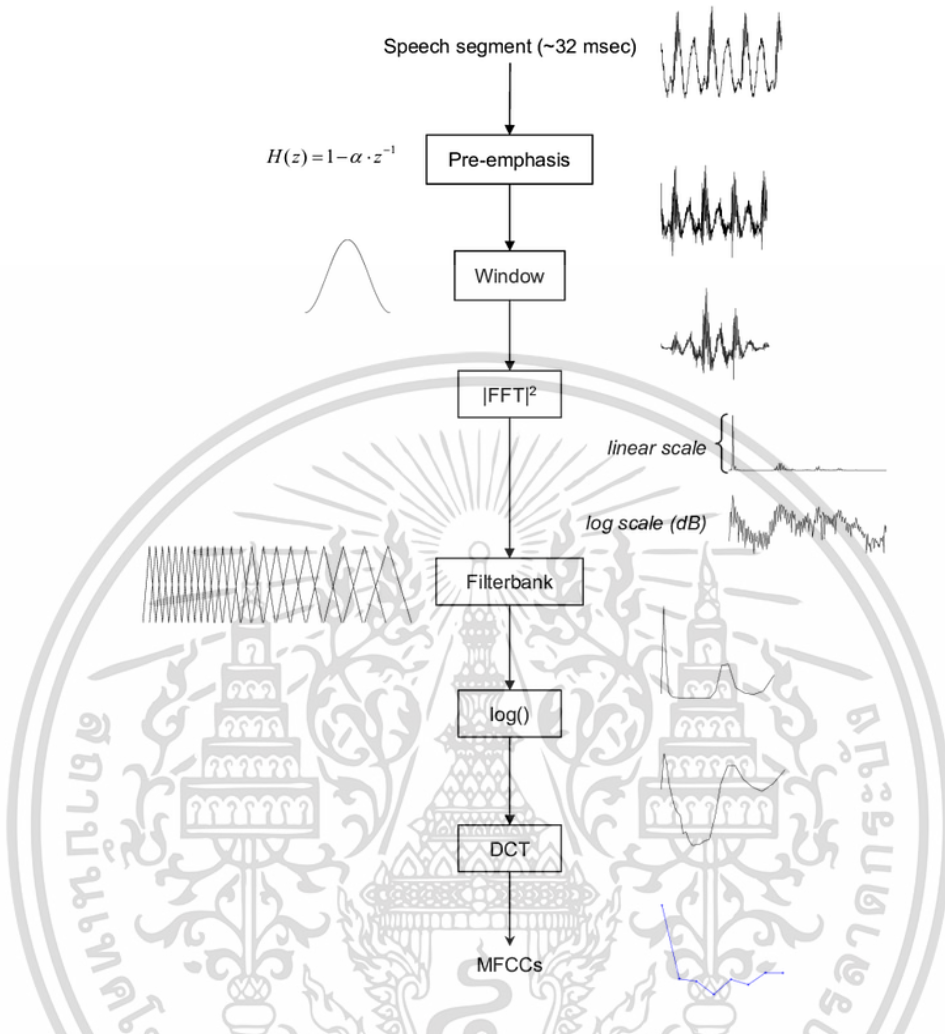
4. คำนวณสเปกตรัมเล็กซ์เซปตรัม (Cepstral Spectrum)

การคำนวณสเปกตรัมเล็กซ์เซปตรัม (Cepstral Spectrum) โดยการใช้การทำลอกซ์เกอร์ (cepstral lifter) เพื่อควบคุมความถี่และลดขนาดข้อมูล

5. คำนวณ MFCC

โดยทั่วไปจะเลือกเอาเฉพาะค่าลอกซ์เซม (cepstral coefficients) ที่ 2 ถึง 13 และค่าลอกซ์เซมแบบที่ 0 (energy coefficient) จากนั้นจะได้ค่า MFCC ของแต่ละกรอบเวลา ขั้นตอนี่ 5 นี้เป็นขั้นตอนสุดท้ายโดยค่า MFCC สำหรับแต่ละกรอบเวลาที่แปลงจากสัญญาณเสียงเรียบริ้วและสามารถนำไปใช้ในการรู้จำเสียงพูดหรือแตกแยกคำพูดในการประมวลผลเสียงและเสียงพูดอย่างมีประสิทธิภาพโดยมีตัวอย่างขั้นตอนในการคำนวณหาค่าสัมประสิทธิ์เซปตรัมบนสเกลเมล ดังภาพที่

2.5



ภาพที่ 2.5 ขั้นตอนในการคำนวณหาค่าสัมประสิทธิ์เซปตรัมบนสเกลเมล [9]

2.6.3 Discrete Fourier Transform (DFT)

การแปลงฟูริเยร์ Discrete Fourier Transform (DFT) เป็นกระบวนการทางคณิตศาสตร์ที่ใช้ในการแปลงสัญญาณแบบไม่ต่อเนื่อง (discrete-time signal) จากโดเมนเวลา (time domain) เป็นโดเมนความถี่ (frequency domain) โดยใช้การใชฟังก์ชันโคไซน์ (cosine) และไซน์ (sine) เป็นฐาน DFT ช่วยแยกสัญญาณออกเป็นความถี่ที่ประกอบสัญญาณนั้น ๆ ซึ่งเป็นสิ่งสำคัญในการวิเคราะห์และประมวลผลสัญญาณเช่นเสียง รูปภาพ หรือสัญญาณอื่น ๆ ที่แสดงอยู่ในโดเมนความถี่ (frequency domain) ตัวอย่างสมการของ DFT ดังสมการที่ (1)

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{-j2\pi kn/N} \quad [9] \quad (1)$$

$X[k]$ คือ DFT ของสัญญาณ $x[n]$ ที่ความถี่ k

$x[n]$ คือ สัญญาณไม่ต่อเนื่องในโดเมนเวลา

e คือค่าคงที่ของเลขจินตภาพ (complex exponential) ซึ่งเกี่ยวข้องกับความถี่และตำแหน่งในโดเมนความถี่

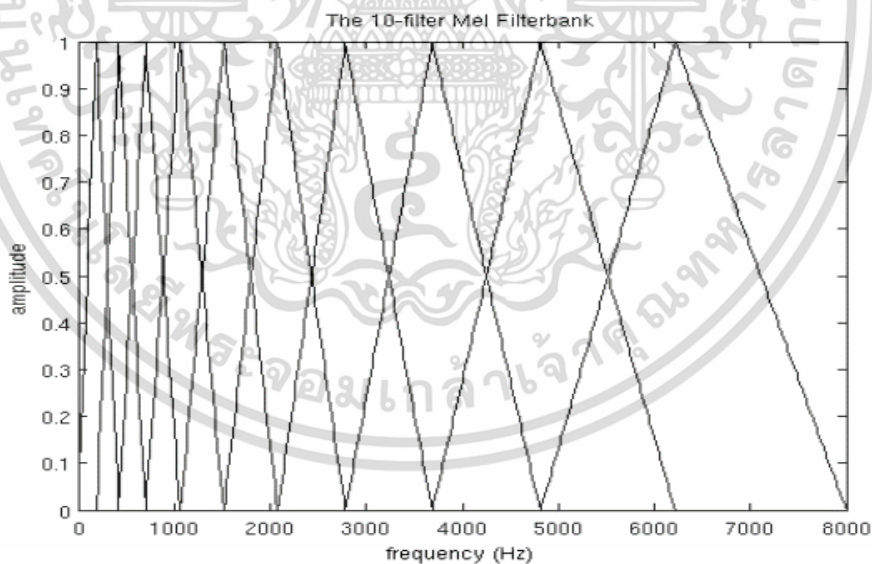
j คือหน่วยเชิงมุม (imaginary unit)

k คือค่าความถี่ที่ต้องการหา DFT สำหรับ ($k = 0, 1, 2, \dots, N-1$)

ผลลัพธ์ $X[k]$ จะแสดงความสัมพันธ์ระหว่างสัญญาณ $x[n]$ และความถี่ k ในโดเมนความถี่ (frequency domain) ซึ่งสามารถใช้ในการวิเคราะห์สัญญาณ หาความถี่หรือความสัมพันธ์ระหว่างความถี่ต่าง ๆ ของสัญญาณ หรือใช้ในงานอื่น ๆ ที่เกี่ยวกับความถี่และความแตกต่างระหว่างสัญญาณต่าง ๆ ในโดเมนความถี่ (frequency domain)

2.6.4 Mel-frequency filter bank

Mel-frequency filter bank การหาค่าสัมประสิทธิ์เซปสตรัมบนสเกลเมล เริ่มต้นจากการนำสัญญาณเสียงมาผ่านการประมวลผลสัญญาณเสียง หลังจากนั้นส่งสัญญาณไปผ่านชุดตัวกรองฟิลเตอร์แบงค์ (filter bank) เพื่อเน้นความสำคัญของความถี่ที่อยู่ในช่วงกลางของชุดตัวกรองแต่ละตัวกรอง ชุดตัวกรองฟิลเตอร์แบงค์มีลักษณะ ดังภาพที่ 2.6



ภาพที่ 2.6 ชุดตัวกรองฟิลเตอร์แบงค์ [9]

โดยที่ความถี่กลางของตัวกรองแต่ละชุดนั้นเกิดจากการแปลงค่าความถี่ปกติ (f) ให้อยู่บนสเกลเมล (f_{Mel}) ดังสมการที่ (2)

$$f_{Mel} = 2595 \log_{10}\left(1 + \frac{f}{700}\right) \quad [9] \quad (2)$$

การคำนวณหาพลังงานสเปกตรัมที่ผ่านตัวกรองขั้นตอนนี้ นำความถี่ที่ได้จากขั้นตอนคำนวณสเปกตรัมมาหาขนาดกำลังสอง $|x(k)|^2$ ส่งผ่านชุดตัวกรองแบบสามเหลี่ยมในสเกลเมลเพื่อเน้นความสำคัญของความถี่ที่อยู่ในช่วงกลางของชุดตัวกรองแต่ละตัวกรอง ดังสมการที่ (3)

$$s(m) = \sum_{k=0}^{N-1} [|X(k)|^2 H_m(k)]; \quad 0 \leq m \leq M-1 \quad [9] \quad (3)$$

การคำนวณสัมประสิทธิ์เซปสตรัมบนสเกลเมล (MFCC) ในขั้นตอนนี้ นำลอการิทึมของพลังงานมาผ่านการแปลงโคไซน์แบบไม่ต่อเนื่องทำให้ได้ค่าสัมประสิทธิ์เซปสตรัมบนสเกลเมล C ลำดับที่ m ตามสมการ ดังสมการที่ (4) และ (5)

$$c_m = w_t(m) \sum_{j=1}^J \log_{10}(E_j) \cos_{m=0,1,2,\dots,J-1} \left(\frac{\pi}{J}(j-0.5)m \right) \quad [9] \quad (4)$$

$$\text{เมื่อ } w_t(m) = \begin{cases} \frac{1}{\sqrt{J}}, & m=0 \\ \frac{\sqrt{2}}{J}, & 1 \leq m < J \end{cases} \quad [9] \quad (5)$$

2.7 Artificial Intelligence (AI)

ปัญญาประดิษฐ์ (AI) [10] คือซอฟต์แวร์หรือระบบคอมพิวเตอร์ที่ถูกสร้างขึ้นเพื่อมาทำงานในลักษณะที่คล้ายมนุษย์ในด้านการคิด, การตัดสินใจ, และการเรียนรู้จากข้อมูล โดยอาศัยเทคโนโลยีและวิธีการทางคณิตศาสตร์เชิงคอมพิวเตอร์ เช่น Machine Learning, Deep Learning, การประมวลผลข้อมูลแบบควอนตัม, การจำลองการตัดสินใจแบบควอนตัม, และการประมวลผลภาษาธรรมชาติ เพื่อทำงานในหลายสถานการณ์และสามารถปรับปรุงประสิทธิภาพของการทำงานตามสถานการณ์ รวมถึงการเรียนรู้และปรับตัวจากประสบการณ์ในขณะที่ทำงาน โดยมีวัตถุประสงค์หลักในการทำงานคือการแก้ปัญหา, การวิเคราะห์ข้อมูล, การตัดสินใจ, และการเข้าใจและประมวลผลข้อมูลแบบธรรมชาติอย่างเป็นระบบ เพราะฉะนั้นจึงสามารถกล่าวได้ว่า AI ถือกำเนิดขึ้นเมื่อซอฟต์แวร์หรือระบบคอมพิวเตอร์มีความสามารถที่จะเรียนรู้ โดย AI ถูกแบ่งออกเป็นหลายระดับตามความสามารถหรือความฉลาด โดยจะวัดจากความสามารถในการ ให้เหตุผล การพูด และทัศนคติของ AI ตัวนั้นๆ เมื่อเปรียบเทียบกับมนุษย์

AI ถูกจำแนกเป็น 3 ระดับตามความสามารถ ดังนี้

- ปัญญาประดิษฐ์เชิงแคบ Artificial Narrow Intelligence (ANI) หรือ Narrow AI โดยระดับนี้มีความสามารถในการทำงานเฉพาะเจาะจงในงานหรือภารกิจที่เฉพาะเจาะจง ไม่มีความเข้าใจหรือความตระหนักเหมือนมนุษย์

- ปัญญาประดิษฐ์ทั่วไป Artificial General Intelligence (AGI) หรือ General AI โดยระดับนี้มีความสามารถแบบมนุษย์ทั่วไปในการคิด, การเรียนรู้, และการทำงานในหลายสถานการณ์

- ปัญญาประดิษฐ์แบบอัจฉริยะ Superintelligent AI โดยระดับนี้เกิดจากการพัฒนาของ AGI และมีความสามารถในการคิดและแก้ปัญหาที่มีความซับซ้อนมากขึ้นจากระบบมนุษย์ Superintelligent AI มีความสามารถในการเรียนรู้และปรับตัวอย่างมากและมีความเชื่อมโยงกับความรู้มากมาย ระบบ Superintelligent AI อาจมีความสามารถในการทำนายเหตุการณ์อันมีผลกระทบมากๆ หรือมีความสามารถในการสร้างเทคโนโลยีและการแก้ปัญหาที่เป็นความซับซ้อน

2.8 Machine Learning

Machine Learning [11] คือกระบวนการที่ระบบคอมพิวเตอร์สามารถเรียนรู้และปรับปรุงการทำงานของตนเองได้โดยไม่ต้องรับคำสั่งโดยตรงจากผู้เขียนโปรแกรม ในกระบวนการนี้ ระบบคอมพิวเตอร์จะใช้ข้อมูลที่มีอยู่ในระบบเพื่อวิเคราะห์และค้นหาความสัมพันธ์ โดยการใช้วิธีการเรียนรู้ทางสถิติและอัลกอริทึมต่าง ๆ เพื่อสร้างแบบจำลองทางคณิตศาสตร์ ที่สามารถใช้ในการทำนายผลลัพธ์หรือการตัดสินใจในสถานการณ์ที่มีความซับซ้อนหรือไม่แน่นอนได้

2.8.1 ประเภทของ Machine Learning

ประเภทของ Machine Learning สามารถแบ่งออกได้เป็น 3 รูปแบบได้แก่ Supervised Learning, Unsupervised Learning และ Reinforcement Learning โดยในรายละเอียดของแต่ละประเภทจะเป็นดังนี้

Supervised Learning (การเรียนรู้แบบมีการควบคุม)

ระบบคอมพิวเตอร์ได้รับชุดข้อมูลที่มีคำตอบ (label) และเรียนรู้ในการสร้างแบบจำลองที่สามารถทำนายผลลัพธ์จากข้อมูลใหม่ได้ ตัวอย่างเช่น การจำแนกภาพว่าเป็นภาพของสุนัขหรือแมว

Unsupervised Learning (การเรียนรู้แบบไม่มีการควบคุม)

ระบบคอมพิวเตอร์ไม่มีคำตอบ (Label) ในข้อมูลที่ใช้ในการเรียนรู้ แต่จะสร้างแบบจำลองที่สามารถจัดกลุ่มหรือแบ่งแยกข้อมูลได้ ตัวอย่างเช่น การจัดกลุ่มลูกค้าที่มีพฤติกรรมการซื้อสินค้าคล้ายกัน

Reinforcement Learning (การเรียนรู้แบบเสริม)

ระบบคอมพิวเตอร์เรียนรู้โดยการทดลองและตอบรับต่อสิ่งแวดล้อม โดยระบบจะได้รับรางวัลหรือลื่นชกเมื่อดำเนินการที่ถูกต้อง และได้รับลื่นชกลบเมื่อดำเนินการที่ไม่ถูกต้อง ตัวอย่างเช่น การเรียนรู้การเล่นเกมหรือการควบคุมหุ่นยนต์ให้ทำงานในสภาวะที่ต้องการ

นอกจากนี้ยังมีการแบ่ง Machine Learning เป็นประเภทอื่น ๆ เช่น Semi-supervised Learning, Transfer Learning, Active Learning, Online Learning, และ Deep Learning ซึ่งเป็นการเรียนรู้เชิงลึก (เรียนรู้จากข้อมูลที่มีมิติสูง) โดยใช้โครงข่ายประสาทเทียม (Neural Networks) ในการสร้างแบบจำลอง

2.9 Deep learning

Deep learning [12] คือเทคนิคการเรียนรู้เชิงลึก (Deep Learning) ที่ใช้โครงข่ายประสาทเทียม (Neural Networks) ในการสร้างแบบจำลองที่มีความซับซ้อนและมีความลึกมากเพื่อใช้ในการทำนายหรือการตัดสินใจในข้อมูลที่มีมิติสูง โครงข่ายประสาทเทียมใน Deep Learning มีโครงสร้างที่คล้ายกับโครงข่ายประสาทในระดับบนของระบบประสาทของมนุษย์ โดยมีการเชื่อมต่อหลายชั้น (Layers) และมีการปรับค่าน้ำหนัก (Weights) ของโหนด (Nodes) ในแต่ละชั้นเพื่อให้โมเดลสามารถเรียนรู้และสร้างความเข้าใจเกี่ยวกับข้อมูลได้เอง ด้วยความสามารถในการเรียนรู้เชิงลึก โมเดล Deep Learning สามารถจัดการกับข้อมูลที่มีมิติสูงและซับซ้อนได้ดีกว่าเทคนิคการเรียนรู้ทางสถิติแบบอื่น ๆ อย่างเช่นการเรียนรู้แบบเสริม (Supervised Learning) หรือการเรียนรู้แบบไม่มีการควบคุม (Unsupervised Learning) ซึ่งทำให้ Deep Learning เป็นเครื่องมือที่สำคัญในการแก้ไขปัญหาทางการประมวลผลภาพ การประมวลผลเสียง การแปลภาษา การวิเคราะห์ข้อมูล และงานอื่น ๆ ที่เกี่ยวข้องกับการจัดการข้อมูลที่มีความซับซ้อนและมีมิติสูง

2.9.1 โครงข่ายประสาทเทียม (Neural Networks)

Neural Networks [12] เป็นโมเดลคณิตศาสตร์ที่จำลองการทำงานของระบบประสาทของมนุษย์ โดยมีโครงสร้างที่คล้ายกับเซลล์ประสาทของสมองมนุษย์ โครงข่ายประสาทเทียมประกอบด้วยหน่วยประมวลผลที่เรียกว่าโหนด (Nodes) และเชื่อมต่อกันเป็นชั้น (Layers) เพื่อส่งสัญญาณข้อมูลระหว่างโหนด โดยการเชื่อมต่อนี้จะมีน้ำหนัก (Weights) ที่กำหนดค่าความสำคัญของการส่งสัญญาณระหว่างโหนด นอกจากนี้ยังมีฟังก์ชันการกระตุ้น (Activation Function) ที่ใช้ในการประมวลผลของแต่ละโหนด เพื่อให้ได้ผลลัพธ์ที่เหมาะสม

โครงข่ายประสาทเทียมถูกนำมาใช้ในหลายงานที่เกี่ยวข้องกับการจัดการข้อมูล อาทิเช่น การจัดกลุ่มข้อมูล (Clustering) การจัดลำดับข้อมูล (Sequence Learning) การจำแนกข้อมูล (Classification) และการทำนาย (Prediction) โดยการปรับค่าน้ำหนักในโครงข่ายประสาทเทียมและการเรียนรู้จากข้อมูล (Training) โครงข่ายประสาทเทียมสามารถเรียนรู้และปรับปรุงความสามารถในการทำงานได้

2.9.2 ตัวอย่างความสามารถของ Deep learning ในการจำแนกเสียงพูดหรือการจดจำเสียงพูด

Deep Learning มีความสามารถหลายอย่างที่ทำให้ Deep Learning เป็นเทคโนโลยีที่สำคัญและมีประโยชน์ในหลายสาขา โดย Deep Learning มีความสามารถในการจำแนกและแปลงเสียงพูดให้อัตโนมัติ ดังนี้

การแปลงเสียงพูดเป็นข้อความ (Speech-to-Text) โดย Deep Learning สามารถใช้ในการแปลงเสียงพูดเป็นข้อความ โดยสร้างระบบที่สามารถตรวจจับและแปลงคำพูดให้อยู่ในรูปของข้อความเข้าใจได้ ตัวอย่างเช่น การแปลงเสียงพูดเป็นข้อความ (Speech-to-Text) [13]

การจดจำเสียงพูด (Speaker Recognition) โดย Deep Learning สามารถใช้ในการจดจำเสียงพูดของบุคคลหรือรูปแบบเสียงพูด เป็นความสามารถที่ใช้ในระบบตรวจจับเสียงพูดของผู้ใช้ ตัวอย่างเช่น ระบบการจดจำและยืนยันตัวตนของผู้พูด [14]

การจัดการเสียง (Audio Processing) โดย Deep Learning สามารถใช้ในการปรับแต่งและจัดการเสียง ตัวอย่างเช่น การแยกแยะและประมวลผลเสียงเพลง [15]

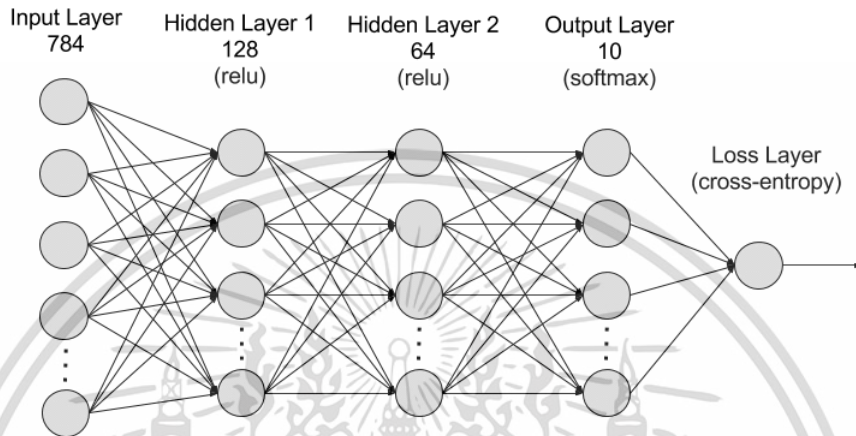
การรู้จำแบบเสียง (Voice Pattern Recognition) โดย Deep Learning สามารถใช้ในการรู้จำแบบเสียงของบุคคลหรือสาธารณชน ตัวอย่างเช่น การสร้างโมเดลที่จดจำแบบเสียงในระบบรู้จำเสียงพูด [16]

การจำแนกคุณสมบัติเสียง (Classification of Voice Properties) โดย Deep Learning สามารถใช้ในการจำแนกและวิเคราะห์คุณสมบัติเสียง เช่น การแยกแยะเสียงพูดจากเสียงพื้นหลังหรือการตรวจจับอารมณ์จากเสียงพูด [17]

2.9.3 การทำงานของ Deep learning

Algorithm ของ Deep learning เป็น Neural Networks ที่สร้างแบบจำลองตามสมองของมนุษย์ ยกตัวอย่างเช่น สมองมนุษย์มีเซลล์ประสาทที่เชื่อมต่อกันและทำงานร่วมกันเพื่อเรียนรู้และประมวลผลข้อมูลในทำนองเดียวกัน เครือข่าย Deep learning หรือ Neural Networks มาจากเซลล์

ประสาทเทียมหลายเลเยอร์ที่ทำงานร่วมกันภายในคอมพิวเตอร์ เซลล์ประสาทเทียมเป็นโมดูลซอฟต์แวร์ที่เรียกว่าโหนด ซึ่งใช้การคำนวณทางคณิตศาสตร์เพื่อประมวลผลข้อมูล Neural Networks เป็นขั้นตอนที่ใช้โหนดเหล่านี้ในการแก้ปัญหาที่ซับซ้อน ดังภาพที่ 2.7 [18]



ภาพที่ 2.7 ภาพตัวอย่างอัลกอริธึม Deep learning [18]

2.9.4 ส่วนประกอบของเครือข่าย Deep learning

Deep Learning ประกอบด้วยส่วนประกอบหลายอย่างที่ทำงานร่วมกันเพื่อการเรียนรู้และการทำนายข้อมูลในรูปแบบที่ซับซ้อน ส่วนประกอบสำคัญของเครือข่าย Deep Learning ประกอบด้วย

Input Layer (ชั้นนำเข้า) คือส่วนที่รับข้อมูลเข้าสู่โครงข่าย Deep Learning โดยปกติจะเป็นข้อมูลที่เป็นตัวเลข ตัวอย่างเช่นรูปภาพขนาดพิกเซลหรือเสียงเป็นคลื่นไฟฟ้า

โครงข่ายนิวรัลเน็ตเวิร์กมีโหนดหลายโหนดที่อินพุตข้อมูลเข้าไปในนั้น โหนดเหล่านี้ทำขึ้นมาจากการป้อนข้อมูลของระบบ

Hidden Layers (ชั้นซ่อน) คือส่วนที่อยู่ระหว่างชั้นนำเข้าและชั้นส่งออก มีหน้าที่ประมวลผลข้อมูลโดยใช้การคำนวณทางคณิตศาสตร์ ประกอบด้วยหลายๆ โหนด (Neurons) ที่ทำงานร่วมกัน

Weights and Bias (น้ำหนักและค่าอคติ) โดยน้ำหนัก (weights) และค่าอคติ (bias) เป็นพารามิเตอร์ที่ปรับค่าในแต่ละโหนดในชั้นซ่อน เพื่อให้โครงข่ายประสาทเทียมสามารถเรียนรู้และปรับปรุงการทำนายข้อมูล

Activation Functions (ฟังก์ชันการกระตุ้น) คือฟังก์ชันการกระตุ้น (Activation Functions) ใช้เพื่อกำหนดค่าของโหนดในชั้นซ่อน แต่ละโหนดจะใช้ฟังก์ชันนี้เพื่อคำนวณผลลัพธ์ของการประมวลผล

Output Layer (ชั้นส่งออก) คือชั้นส่งออกเป็นส่วนที่ให้ผลลัพธ์สุดท้ายหลังจากที่ข้อมูลผ่านการประมวลผลในชั้นซ่อน ประกอบด้วยโหนดที่ใช้ในการทำนายผลลัพธ์

Loss Function (ฟังก์ชันความคลาดเคลื่อน) คือฟังก์ชันความคลาดเคลื่อน (Loss Function) ใช้ในการวัดความแตกต่างระหว่างค่าที่ทำนายได้และค่าจริง ซึ่งจะถูกนำมาใช้ในกระบวนการปรับพารามิเตอร์ของโครงข่าย

Optimization Algorithm (อัลกอริทึมการปรับค่าพารามิเตอร์) คืออัลกอริทึมการปรับค่าพารามิเตอร์ (optimization algorithm) เช่น Gradient Descent ถูกใช้ในการปรับค่าน้ำหนักและค่าตัดเพื่อลดความคลาดเคลื่อนของโมเดล

Training Data (ข้อมูลการฝึก) คือข้อมูลการฝึกคือชุดข้อมูลที่ใช้ในการฝึกโครงข่าย Deep Learning เพื่อให้โมเดลเรียนรู้ความสัมพันธ์ในข้อมูล

Validation Data (ข้อมูลการตรวจสอบความถูกต้อง) คือข้อมูลการตรวจสอบความถูกต้องใช้ในกระบวนการตรวจสอบและปรับปรุงโมเดลในขณะฝึก

Test Data (ข้อมูลการทดสอบ) คือข้อมูลการทดสอบใช้ในการประเมินความสามารถในการทำนายของโมเดลหลังจากการฝึกและการปรับปรุง

2.10 Convolutional Neural Network (CNN)

Convolutional Neural Network [19] คือ Neural Network แบบหนึ่งที่ถูกนำมาใช้ในงาน Computer Vision หรือ วิเคราะห์รูปภาพ เช่น Image Classification จำแนกรูปภาพ, Object Detection ตรวจจับวัตถุ, Face Recognition เรียนรู้จดจำใบหน้า, เป็นต้น และ CNN เป็นโครงข่ายประสาทเทียมหนึ่งในกลุ่ม Bio-Inspired โดยที่ CNN จะจำลองการมองเห็นของมนุษย์ที่มองเห็นพื้นที่เป็นที้อยๆ และนำกลุ่มของพื้นที่ย่อย ๆ มาผสานกัน เพื่อดูว่าสิ่งที่เห็นอยู่เป็นอะไร การมองเห็นที้อยๆ ของมนุษย์จะมีการแยกคุณลักษณะ Feature ของพื้นที่ย่อยนั้น เช่น ลายเส้น และการตัดกันของสี ซึ่งการที่มนุษย์รู้ว่าพื้นที่ตรงนี้เป็นเส้นตรงหรือสี่เหลี่ยม เพราะมนุษย์ดูทั้งจุดที่สนใจและบริเวณรอบ ๆ ประกอบกัน

2.10.1 การทำงานของ Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) เริ่มต้นด้วยการรับภาพเป็นข้อมูลนำเข้า ซึ่งจากนั้นจะถูกแปลงเป็นแผนที่คุณลักษณะผ่านชั้นคอนโวลูชันและชั้นการสรุป ชั้นคอนโวลูชันจะใช้ชุดของฟิลเตอร์หลาย ๆ ชุดในการประมวลผลภาพนำเข้า แต่ละฟิลเตอร์จะสร้างแผนที่ลักษณะเด่นที่เน้นเฉพาะด้านของภาพนำเข้า ชั้นพูลลิ่งจะลดขนาดแผนที่ลักษณะเด่นข้อมูลลง ในขณะที่ยังคงข้อมูลที่สำคัญที่สุดอยู่

แผนที่คุณลักษณะเด่นที่สร้างโดยชั้นคอนโวลูชันจะถูกส่งผ่านชั้นคอนโวลูชันและชั้นพูลลิ่งเพิ่มเติมหลายชั้น แต่ละชั้นจะเรียนรู้คุณลักษณะเด่นที่ซับซ้อนขึ้นของภาพนำเข้า ผลลัพธ์สุดท้ายของเครือข่ายคือป้ายชื่อคลาสที่คาดการณ์หรือคะแนนความน่าจะเป็นสำหรับแต่ละคลาส ขึ้นอยู่กับประสิทธิภาพ

2.10.2 โครงสร้างของ Convolutional Neural Network (CNN)

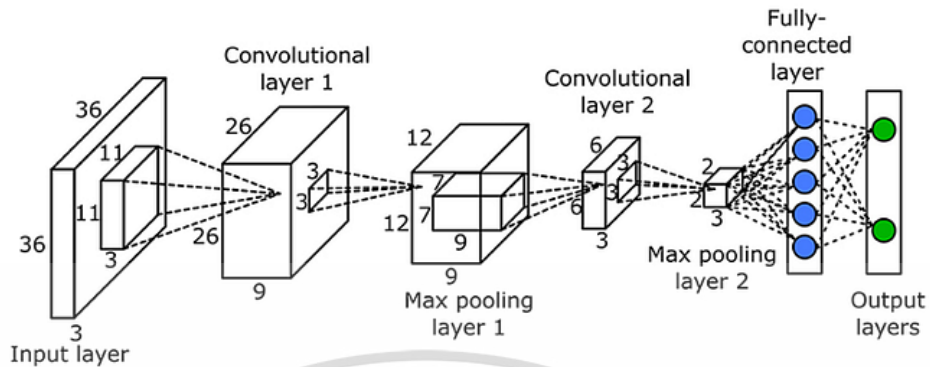
โครงสร้างทั่วไปของ Convolutional Neural Network (CNN) ประกอบด้วยสามส่วนหลัก ได้แก่ 1. ชั้นนำเข้า (input layer) 2. ชั้นที่ซ่อนอยู่ (hidden layers) และ 3. ชั้นเอาต์พุต (output layer) ชั้นนำเข้ารับรูปภาพนำเข้าและส่งต่อไปยังชั้นที่ซ่อนอยู่ ซึ่งประกอบด้วยชั้นคอนโวลูชันและชั้นพูลลิ่งหลาย ๆ ชั้น ชั้นเอาต์พุตให้ผลลัพธ์ที่ทำนายคลาสหรือคะแนนความน่าจะเป็นสำหรับแต่ละคลาส

ชั้นที่ซ่อนอยู่เป็นส่วนที่สำคัญที่สุดของ CNN และจำนวนชั้นซ่อนและจำนวนฟิลเตอร์ในแต่ละชั้นสามารถปรับเพื่อปรับปรุงประสิทธิภาพของเครือข่าย โครงสร้างที่พบบ่อยสำหรับ CNN คือการมีชั้นคอนโวลูชันหลายชั้น ตามด้วยชั้นพูลลิ่งหนึ่งหรือมากกว่า และจากนั้นคือชั้นแบ็กโพรพาคชันทั้งหมดที่ให้ผลลัพธ์สุดท้าย

2.10.3 เลเยอร์ของ Convolutional Neural Network (CNN)

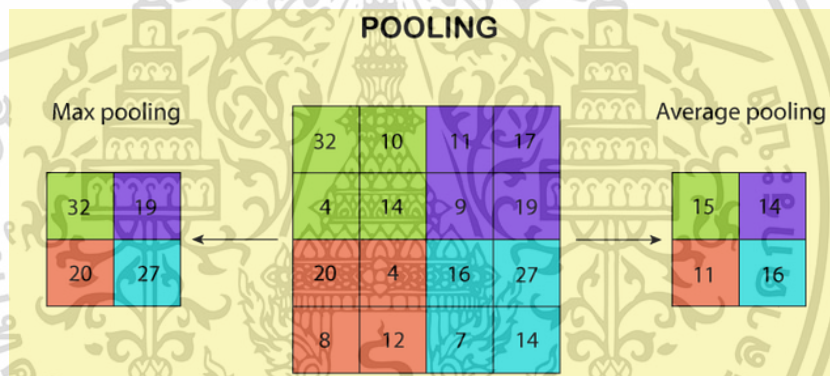
CNN ประกอบด้วยชั้นการประมวลผลหลายชั้น ซึ่งประกอบไปด้วยชั้นต่างๆ เช่น Convolutional Layer, Pooling Layer, Activation Layer และ Fully Connected Layer

Convolutional Layer คือชั้นนี้ใช้ในการสกัดลักษณะเด่น (Feature Extraction) จากภาพ โดยมีการนำเอาเมทริกซ์คอนโวลูชัน (Convolutional Matrix) ที่เรียกว่า Kernel มาทำการคูณกับพิกเซลในภาพ โดยทำให้เกิดการสกัดลักษณะเด่นออกมา เช่น ขอบ รูปร่าง หรือลวดลาย ซึ่งเป็นขั้นตอนสำคัญในการเรียนรู้ลักษณะของวัตถุในภาพ ดังภาพที่ 2.8



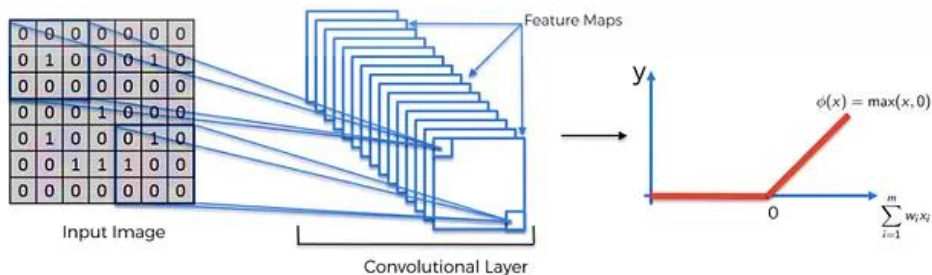
ภาพที่ 2.8 Convolutional Layer [19]

Pooling Layer คือชั้นนี้ใช้ในการลดขนาดของภาพและลดการคำนวณ โดยการนำเอาพิกเซลบางส่วนในเอกซ์ตรางของภาพมาเลือกจากนั้นใช้วิธีการรวมค่าเข้ากัน เช่น วิธีการ Max Pooling หรือ Average Pooling เพื่อให้การคำนวณที่ซับซ้อนลดลง ดังภาพที่ 2.9



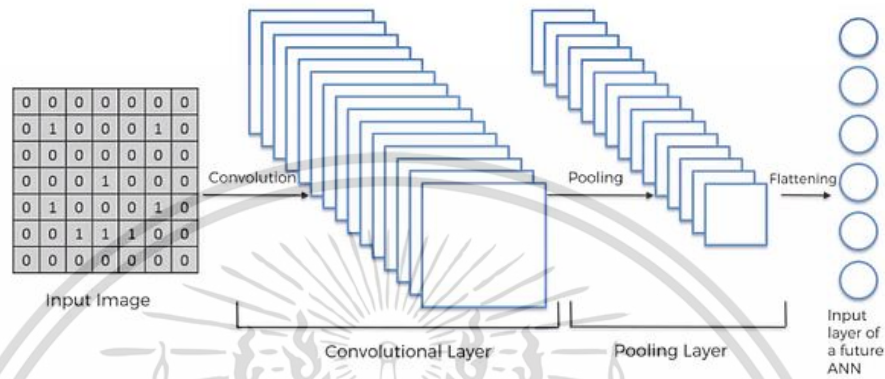
ภาพที่ 2.9 Max Pooling Layer [19]

Activation Layer สำหรับปรับใช้ฟังก์ชันการกระตุ้นที่ไม่เชิงเส้น เช่น ฟังก์ชัน ReLU (Rectified Linear Unit) กับผลลัพธ์ที่ได้จากชั้นพูลลิ่ง (Pooling Layer) หรือชั้นคอนโวลูชัน (Convolutional Layer) และมีวัตถุประสงค์เพื่อเพิ่มความไม่เชิงเส้นให้กับโมเดล ซึ่งจะช่วยให้โมเดลเรียนรู้การแทนที่ของข้อมูลนำเข้าที่ซับซ้อนได้มากขึ้น ดังภาพที่ 2.10



ภาพที่ 2.10 Activation Layer [19]

Fully Connected Layer คือชั้นนี้ใช้ในการจำแนกวัตถุในภาพ โดยนำคุณลักษณะที่ได้จากชั้นก่อนหน้ามาเป็นปัจจัยในการตัดสินใจในการจำแนกวัตถุ โดยใช้โครงข่ายประสาทเชิงลึก (Deep Neural Network) เพื่อทำนายคลาสของวัตถุในภาพ ดังภาพที่ 2.11

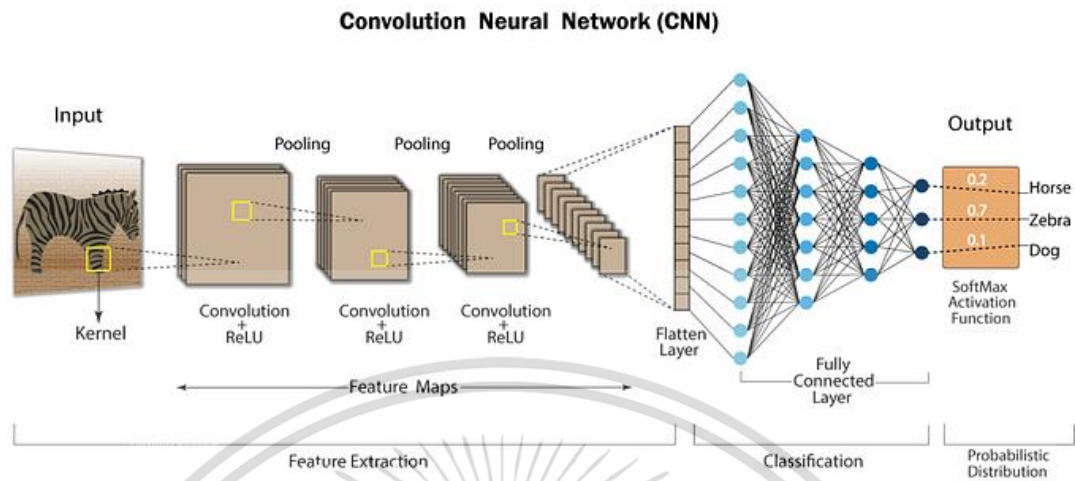


ภาพที่ 2.11 Fully Connected Layer [19]

2.10.4 สถาปัตยกรรมของ Convolutional Neural Network (CNN)

ตัวอย่างสถาปัตยกรรมของ Convolutional Neural Network (CNN) โดยตัวอย่างคือ

1. ชั้นนำเข้ารับภาพเป็นข้อมูลต้นฉบับที่จะถูกประมวลผล
2. ชั้นคอนโวลูชันใช้การคอนโวลูชันเพื่อสกัดลักษณะเด่นจากภาพ โดยการใช้ฟิลเตอร์หรือเคอร์เนลแต่ละตัวในการสร้างแผนที่ลักษณะเด่น (Feature Map) ที่เน้นบริเวณต่าง ๆ ของภาพ
3. ชั้น Activation หลังจากชั้นคอนโวลูชัน CNN ใช้ฟังก์ชันการกระตุ้นที่ไม่เชิงเส้น เช่น ReLU (Rectified Linear Unit) เพื่อเพิ่มความไม่เชิงเส้นให้กับโมเดล
4. ชั้นพูลลิ่งลดขนาดของแผนที่ลักษณะเด่นที่ได้จากชั้นคอนโวลูชัน โดยรักษาข้อมูลสำคัญและลดการซ้ำซ้อน
5. ชั้น Fully Connecte คือชั้นแบ็กโพรพาเกชัน (Fully Connected Layer) เป็นชั้นที่เชื่อมโครงข่ายให้เชื่อมต่อบนแบ็กโพรพาเกชัน สามารถเรียนรู้ความสัมพันธ์ของข้อมูลนำเข้าได้มากขึ้น
6. ชั้นเอาต์พุตชั้นสุดท้ายให้ผลลัพธ์สำหรับงานที่กำหนด ในงานจำแนกภาพอาจเป็นคลาสที่ทำนายได้ หรือหาค่าความน่าจะเป็นสำหรับแต่ละคลาส ดังภาพที่ 2.12



ภาพที่ 2.12 สถาปัตยกรรม CNN [19]

2.11 1D CNN and 2D CNN

2.11.1 1D CNN (1-Dimensional Convolutional Neural Network) [20]

1D CNN คือโครงข่ายประสาทเทียมที่ใช้ในการประมวลผลข้อมูลที่มีลำดับ เช่น ชุดข้อมูลชุดเวลา (Time Series Data) หรือ ข้อมูลเสียง โดยมีการใช้ชั้น Convolutional Layer ในการสกัดลักษณะเด่นจากข้อมูล โดยใช้ Kernel หรือ Filter ในการคูณกับข้อมูลแต่ละจุดในลำดับเพื่อสกัดลักษณะเด่นที่ซับซ้อนขึ้นมา และใช้ Pooling Layer เพื่อลดขนาดของข้อมูล จากนั้นจะมีชั้น Fully Connected Layer เพื่อเรียนรู้ความสัมพันธ์ระหว่างลักษณะเด่นกับคลาสของข้อมูล

1D CNN มักถูกนำมาใช้ในงานทางด้านการประมวลผลข้อมูลที่มีลำดับ เช่น การจดจำเสียง การวิเคราะห์ภาพการเคลื่อนไหว การทำนายแนวโน้มราคาหุ้น และอื่นๆ

2.11.2 สถาปัตยกรรมเครือข่ายประสาทเทียม 1D CNN

ระบบประมวลผลข้อมูลแบบลำดับหรือข้อมูลชุดเวกเตอร์ 1 มิติ (เช่น ชุดข้อมูลตัวเลขชุดเดียว) และมีการใช้ Convolutional Layer เพื่อสกัดลักษณะเด่นของข้อมูลในลำดับหรือชุดข้อมูลด้วยการคอนโวลูชัน (Convolution) ในขณะที่สร้างลักษณะเด่นใหม่ ตัวอย่างของ 1D CNN Architecture สามารถอธิบายได้ดังนี้

Input layer รับข้อมูลลำดับหรือชุดข้อมูลเวกเตอร์ 1 มิติเข้าสู่โมเดล ขนาดของข้อมูลจะขึ้นอยู่กับงานและแหล่งข้อมูล ตัวอย่างเช่น ข้อมูลลำดับเสียง (Audio Signal) อาจเป็นเวกเตอร์ที่มีจำนวนชั้น (Channels) แทนความแตกต่างระหว่างชุดข้อมูล

Convolutional layer คือชั้นที่ใช้ในการสกัดลักษณะเด่นจากข้อมูลที่มีลำดับ โดยใช้ Kernel หรือ Filter ในการคำนวณเพื่อสกัดลักษณะเด่นที่ซับซ้อนขึ้นมา และสร้าง Feature Maps โดยแต่ละ Feature Map จะมีขนาดเท่ากับข้อมูลเข้าสู่ชั้นนี้

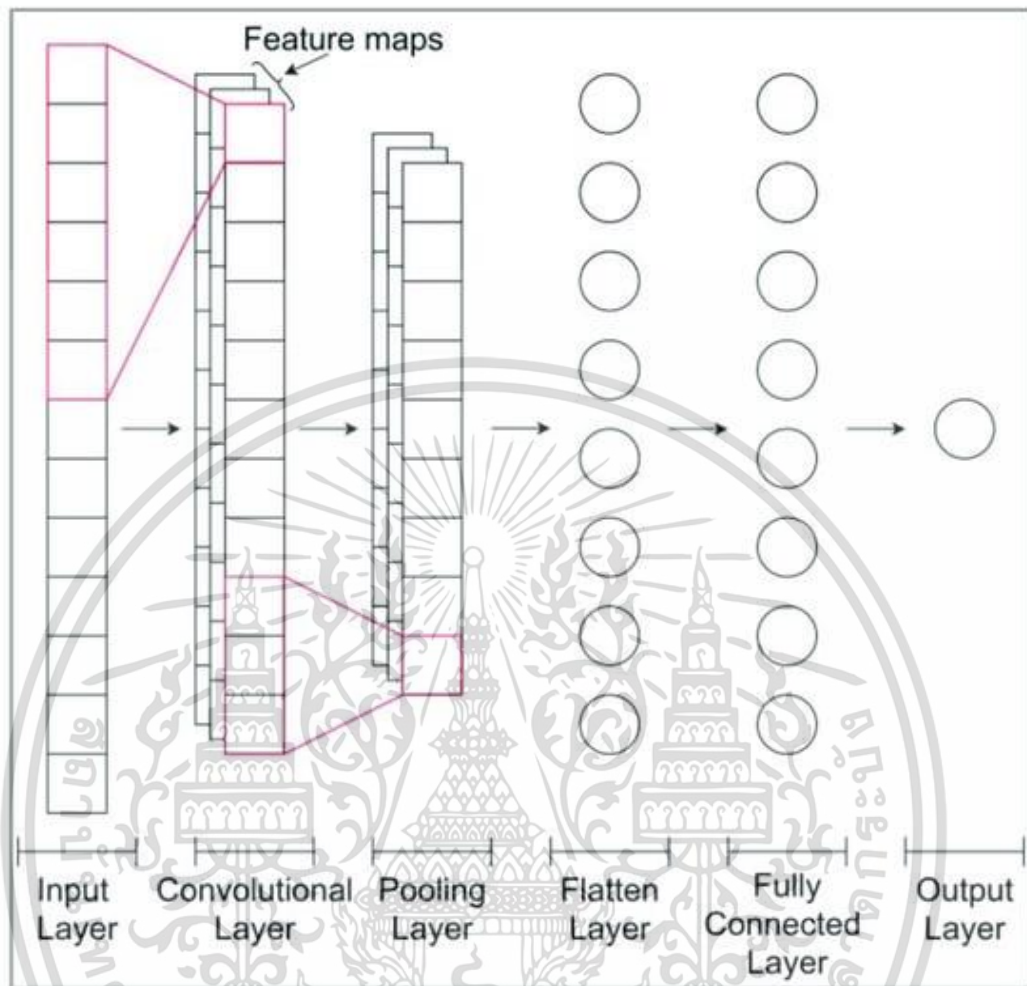
Activation Function คือฟังก์ชันที่ใช้ในการเปิดใช้งานหรือปิดใช้งาน Convolutional Layer ก่อนหน้า โดยผลลัพธ์จากชั้น Convolutional จะผ่านผ่านฟังก์ชันนี้เพื่อให้ได้ผลลัพธ์ที่ไม่เป็นเชิงเส้น

Pooling layer คือชั้นที่ใช้ในการลดขนาดข้อมูล โดยทำการกระจายค่าสูงสุดหรือค่าเฉลี่ยในบริเวณที่กำหนด เพื่อลดขนาดข้อมูลและลดความซับซ้อนของข้อมูล

Flatten Layer คือชั้นที่ใช้ในการแปลงข้อมูลจากรูปแบบที่มีลำดับเป็นรูปแบบที่ไม่มีลำดับ โดยทำการเรียงลำดับข้อมูลในแต่ละตัวอย่างให้เป็นลักษณะเวกเตอร์ (Vector) แทนที่จะเป็นเมทริกซ์ (Matrix) หรือเวกเตอร์ที่มีลำดับเพื่อเตรียมส่งเข้าชั้น Fully Connected

Fully Connected Layer คือชั้นที่เชื่อมต่อกับทุกโหนดในชั้นก่อนหน้า โดยมีความสัมพันธ์ระหว่างลักษณะเด่นที่สกัดมาจากชั้น Convolutional กับคลาสของข้อมูล เพื่อเรียนรู้และทำนายคลาสของข้อมูล

Output layer คือชั้นสุดท้ายที่ใช้ในการสร้างผลลัพธ์ที่เป็นไปได้ของปัญหาที่ต้องการแก้ไข เช่น การจำแนกคลาสของข้อมูล โดยใช้ฟังก์ชันการกระจายหรือฟังก์ชันเชิงเส้นขึ้นอยู่กับประเภทของปัญหา ดังภาพที่ 2.13



ภาพที่ 2.13 สถาปัตยกรรมเครือข่ายประสาทเทียม 1D CNN [20]

2.11.3 2D CNN (2-Dimensional Convolutional Neural Network) [20]

2D CNN เป็นโมเดลปัญญาประดิษฐ์ที่ใช้ในการประมวลผลข้อมูลที่มีลักษณะเป็นภาพหรือแผนที่ โดยมีการใช้ชั้น Convolutional Layer ในการสกัดลักษณะเด่น (Feature Extraction) ของภาพ และชั้น Pooling Layer ในการลดขนาดของข้อมูล ซึ่งช่วยลดความซับซ้อนของข้อมูลและเพิ่มความสัมพันธ์ระหว่างลักษณะเด่นที่สกัดได้ โดยปกติแล้ว 2D CNN มักถูกนำมาใช้ในงานด้านคอมพิวเตอร์วิชันการประมวลผลภาพ เช่น การจำแนกภาพ (Image Classification), การจำแนกเสียง (Audio Classification) การตรวจจับวัตถุ (Object Detection), การตรวจจับใบหน้า (Face Detection), และงานอื่นๆ ที่เกี่ยวข้องกับการประมวลผลภาพ โดยหลักการทำงานของ 2D CNN นั้นจะมีขั้นตอนหลักได้แก่

1. สกัดลักษณะเด่น (Feature Extraction) โดยใช้ชั้น Convolutional Layer เพื่อสกัดลักษณะเด่นของภาพ โดยการทำให้ Convolution ร่วมกับการใช้ฟังก์ชัน Activation เพื่อให้เกิดการกรองลักษณะเด่นที่สำคัญจากภาพ
2. ลดขนาดข้อมูล (Pooling) โดยใช้ชั้น Pooling Layer เพื่อลดขนาดของข้อมูลที่ได้จาก Convolutional Layer เพื่อลดความซับซ้อนของข้อมูลและเพิ่มความสัมพันธ์ระหว่างลักษณะเด่นที่สกัดได้
3. การจำแนกคลาส (Classification) โดยใช้ชั้น Fully Connected Layer เพื่อรวมข้อมูลที่ได้จาก Convolutional Layer และ Pooling Layer เข้าด้วยกัน และนำมาประมวลผลเพื่อทำนายคลาสของภาพ

2D CNN เป็นโมเดลที่มีประสิทธิภาพสูงในการประมวลผลภาพและมีความสามารถในการเรียนรู้ลักษณะเด่นของภาพอย่างมีประสิทธิภาพ ซึ่งทำให้เป็นที่นิยมในการใช้งานด้านการประมวลผลภาพและปัญญาประดิษฐ์

2.11.4 สถาปัตยกรรมเครือข่ายประสาทเทียม 2D CNN

โครงสร้างของ 2D CNN เป็นโมเดลสำหรับการประมวลผลข้อมูลรูปภาพ โดยมีการใช้ Convolutional Layer และ Pooling Layer เพื่อสกัดลักษณะเด่น (Features) จากภาพในลักษณะของแผนที่ลักษณะเด่น (Feature Maps) และลดขนาดของข้อมูลเพื่อลดการคำนวณและเพิ่มความคลาดเคลื่อนต่อข้อมูล ตัวอย่างของ 2D CNN Architecture สามารถอธิบายได้ดังนี้

Input Layer รับข้อมูล 2D เช่น รูปภาพที่มีความกว้าง \times ความสูง \times จำนวนช่องสี (Channels) ตามลำดับ

Convolutional Layers คือโครงสร้างของ Convolutional Layers ใช้ในการสกัดลักษณะเด่นจากข้อมูล 2D โดยใช้ Convolutional filters และทำการคำนวณ Convolution บนข้อมูล หลังจาก Convolutional Layer แต่ละชั้นมักมีการใช้ Activation Function เช่น ReLU เพื่อเพิ่มความไม่เชิงเส้นในโมเดล

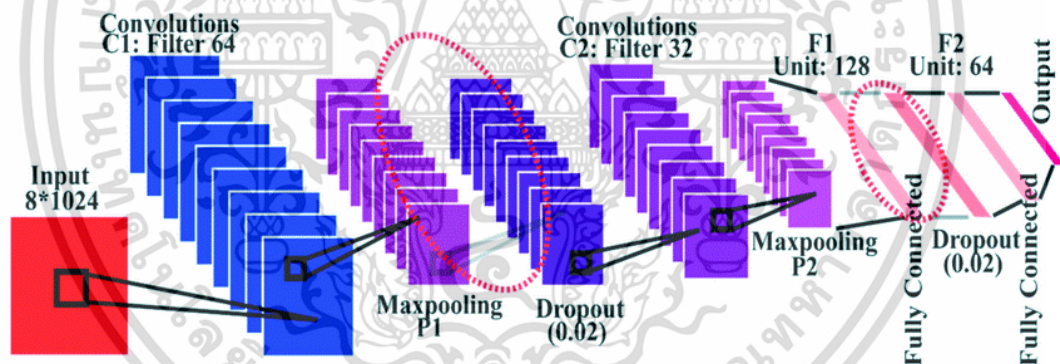
Pooling Layers ชั้น Pooling Layers ใช้ในการลดขนาดของแผนที่ลักษณะเด่น (Feature Maps) ที่ถูกสกัดออกโดยทั่วไปใช้ Max-Pooling เลือกค่าสูงสุดจากบริเวณใหญ่ๆ ในแผนที่ลักษณะเด่น

Dropout Layers ชั้น Dropout Layers เป็นชั้นที่ใช้ป้องกัน Overfitting โดยการปิดการใช้งานบางโครงข่ายในแต่ละรอบการฝึกโดยการสุ่ม เช่น ในตัวอย่างข้างต้น Dropout layer คือชั้นที่ใช้ปิดการใช้งาน Fully Connected Layer โดยระบุ Dropout Rate คือ 50% และ 30% ตามลำดับ การปิดการใช้งานโครงข่ายนี้ช่วยในการลดความเกี่ยวข้องระหว่างโครงข่ายและช่วยป้องกัน Overfitting

Flatten Layer โดยชั้นนี้แปลงข้อมูลลักษณะเด่นที่ได้มาจาก Pooling Layers ล่าสุดเป็นเวกเตอร์ 1 มิติเพื่อเตรียมส่งเข้าสู่ชั้น Fully Connected

Fully Connected Layers โดยชั้น Fully Connected Layers ใช้ในการประมวลผลข้อมูลเป็นเวกเตอร์ขนาดใหญ่และสร้างผลลัพธ์สำหรับงานที่ต้องการ เช่น การจำแนกคลาส โดย Dropout Layer อาจถูกใช้ระหว่างชั้น Fully Connected เพื่อป้องกัน Overfitting

Output Layer โดยชั้น Output Layer ใช้ในการสร้างผลลัพธ์สุดท้ายสำหรับงานที่กำหนด เช่น การจำแนกคลาส และใช้ Activation Function ที่เหมาะสมสำหรับงานนั้น เช่น Softmax สำหรับงานจำแนกคลาส ดังภาพที่ 2.14



ภาพที่ 2.14 สถาปัตยกรรมเครือข่ายประสาทเทียม 2D CNN [20]

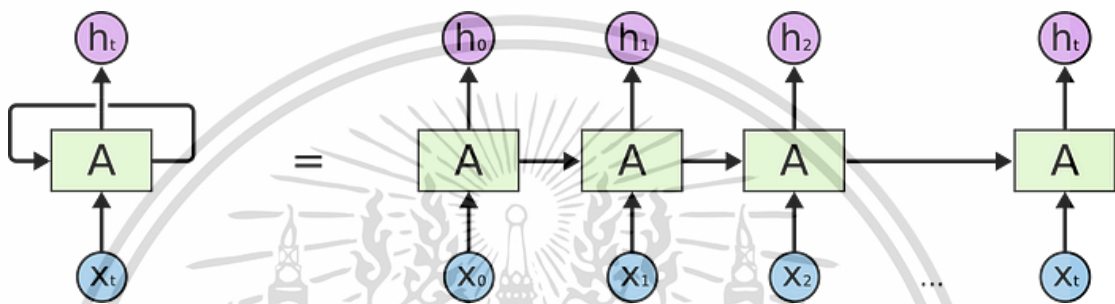
2.12 Recurrent Neural Network (RNN)

Recurrent Neural Network (RNN) [21] เป็นแบบของโครงข่ายประสาทเทียมที่ออกแบบมาเพื่อการประมวลผลข้อมูลลำดับหรือข้อมูลที่มีความเชื่อมโยงในลำดับเวลา โครงข่าย RNN ถูกใช้อย่างแพร่หลายในงานที่เกี่ยวข้องกับข้อมูลลำดับ RNN คือ โครงข่ายประสาทเทียม (Artificial Neuron Network : ANN) รูปแบบหนึ่ง ที่ออกแบบมาเพื่อแก้ปัญหาเกี่ยวกับข้อมูลประเภทอนุกรมเวลา (Time Series Data) หลักการของ RNN คือการปรับรูปแบบของโครงข่ายประสาทเทียม

แบบเดิม เพื่อให้มีชั้นซ่อน (Hidden State) ในการจดจำความรู้หรือข้อมูลก่อนหน้า มารวมเข้ากับข้อมูลตัวใหม่ที่เข้ามา (Input Data) เพื่อใช้ในการทำนายหรือพยากรณ์

2.12.1 โครงสร้างของ Recurrent Neural Network (RNN)

โครงสร้างของ Recurrent Neural Network (RNN) ประกอบด้วยส่วนสำคัญที่เรียกว่า Node หรือ Cell ที่ถูกเชื่อมต่อกันเป็นลำดับเพื่อประมวลผลข้อมูลลำดับในลำดับเวลา ด้านล่างแสดงโครงสร้างของ RNN พื้นฐาน ดังภาพที่ 2.15



ภาพที่ 2.15 โครงสร้างของโครงข่ายระบบประสาทแบบย้อนกลับ [21]

โครงสร้างของ RNN ประกอบด้วยส่วนประกอบหลักดังต่อไปนี้

Cell คือ เซลล์ใน RNN จะเป็นส่วนที่จำลองเซลล์ประสาทในสมอง โดยมีหน้าที่รับข้อมูลเข้ามา ประมวลผล และส่งผลลัพธ์ออกไปยังเซลล์ถัดไปในลำดับ

Hidden State คือ ลูกน้ำเป็นสถานะภายในของเซลล์ ซึ่งจะถูกส่งต่อไปยังเซลล์ถัดไปในลำดับ โดยมีหน้าที่เก็บข้อมูลที่สำคัญจากข้อมูลที่ผ่านมา

Weights คือ น้ำหนักใน RNN เป็นพารามิเตอร์ที่ใช้ในการปรับค่าของข้อมูลเข้าและออกของเซลล์ ซึ่งจะถูกเรียนรู้ในกระบวนการฝึกสอนของโมเดล

Activation Function คือ ฟังก์ชันเชื่อมต่อเป็นฟังก์ชันที่ใช้ในการประมวลผลข้อมูลในเซลล์ ซึ่งสามารถเป็นฟังก์ชันเชิงเส้นหรือฟังก์ชันไม่เชิงเส้นได้ เช่น ฟังก์ชันเชื่อมต่อเชิงเส้น (linear activation function) หรือ ฟังก์ชันเชื่อมต่อแบบซิมพ์สัน (sigmoid activation function)

2.13 LSTM and GRU

2.13.1 Long Short-Term Memory (LSTM) [22]

LSTM คือแบบจำลองของเซลล์ประสาทประเภทหนึ่งที่ถูกใช้ในโครงสร้างของ RNN (Recurrent Neural Network) เพื่อช่วยให้ RNN สามารถจดจำและใช้ข้อมูลในลำดับยาวนาน

LSTM มีโครงสร้างที่ซับซ้อนกว่าเซลล์ประสาทประเภทอื่นๆ เนื่องจากมีการเพิ่มเติมส่วนประกอบเพื่อช่วยให้ LSTM สามารถจัดการกับปัญหาของประสิทธิภาพการเรียนรู้ในเครื่องมือ RNN ที่มีปัญหาในการจำข้อมูลในระยะยาว

LSTM มีส่วนประกอบหลักคือ Cell State ซึ่งเป็นสถานะภายในของเซลล์ LSTM ที่ใช้ในการจำข้อมูลที่สำคัญ และส่วนประกอบอื่นๆ เช่น Forget Gate และ Update Gate เพื่อควบคุมการลืมและการอัปเดตข้อมูลในเซลล์ LSTM

LSTM ได้รับความนิยมในการประยุกต์ใช้ในงานที่ต้องการจำข้อมูลในลำดับยาวนาน เช่น การแปลภาษาอัตโนมัติ การจดจำลำดับเสียงในภาษาธรรมชาติ และการจดจำลำดับของคำในข้อความ

2.13.2 โครงสร้าง Long Short-Term Memory (LSTM)

LSTM มีโครงสร้างพิเศษเพื่อการจดจำและการควบคุมการไหลของข้อมูลในช่วงระยะยาวและระยะสั้น โครงสร้าง LSTM ประกอบด้วยหน่วยควบคุมหลายๆ หน่วยที่ทำงานร่วมกันเพื่อดำเนินการต่างๆ ภายในเซลล์ ประกอบไปด้วย

Input เป็นข้อมูลที่จะถูกนำเข้าไปใน LSTM เช่น ลำดับข้อความที่ต้องการประมวลผล ในลำดับข้อความแต่ละคำหรือตัวอักษรจะถูกแปลงเป็นเวกเตอร์ (Vector Representation) เพื่อเป็นอินพุตให้กับ LSTM

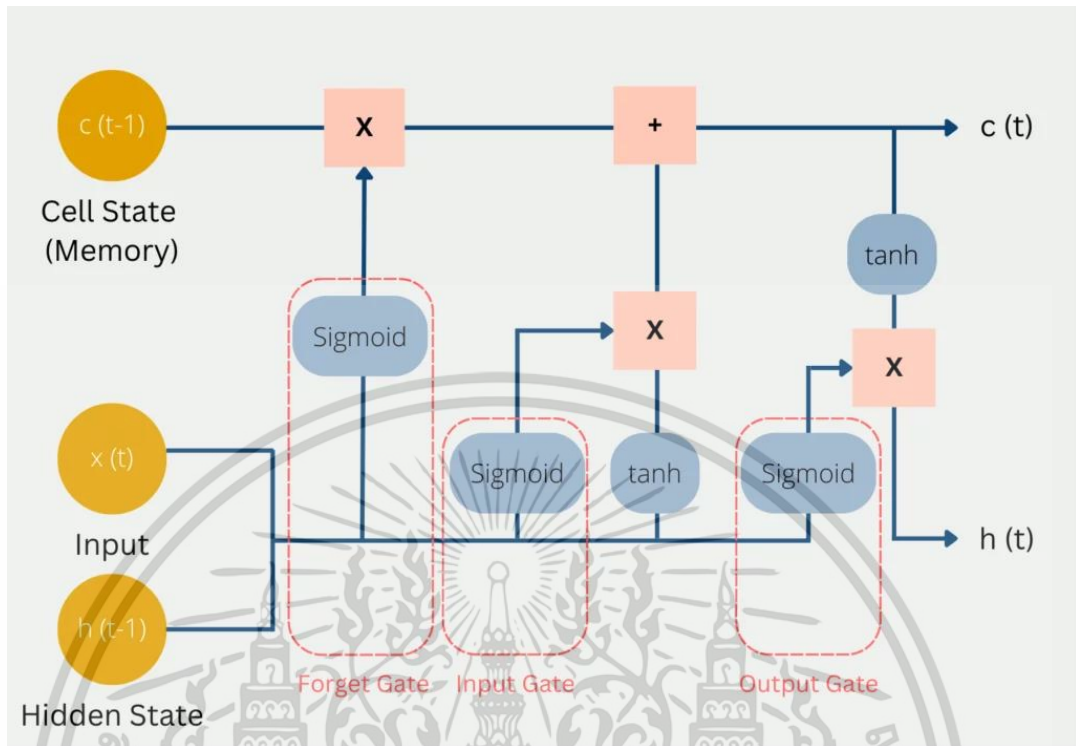
Cell State เป็นส่วนที่เก็บข้อมูลที่สำคัญในระยะยาว และสามารถอัปเดตหรือลบข้อมูลในขั้นตอนการประมวลผล Cell State ช่วยให้ LSTM สามารถจดจำข้อมูลที่มีความสำคัญ

Forget Gate ควบคุมว่าข้อมูลใน Cell State ในขั้นตอนก่อนหน้าควรถูกลบหรือไม่ โดยอิงจากข้อมูลอินพุตและสถานะก่อนหน้าส่วนนี้ช่วยในการเรียนรู้ว่าควรลืมข้อมูลที่ไม่สำคัญ

Input Gate ควบคุมการเพิ่มข้อมูลใหม่เข้าสู่ Cell State โดยอิงจากข้อมูลอินพุตและสถานะก่อนหน้า ส่วนนี้ช่วยในการเรียนรู้ว่าควรเพิ่มข้อมูลอะไรเข้าสู่สถานะ Cell State

Output Gate ควบคุมสถานะใน Cell State ที่ถูกส่งออกไปในขั้นตอนปัจจุบัน ซึ่งขึ้นอยู่กับข้อมูลอินพุตและสถานะปัจจุบันของ Cell State ส่วนนี้ช่วยในการควบคุมผลลัพธ์ที่ส่งออกจาก LSTM

ดั่งภาพที่ 2.16



ภาพที่ 2.16 สถาปัตยกรรมของ LSTM [22]

2.13.3 Gated Recurrent Unit (GRU)

Gated Recurrent Unit (GRU) เป็นแบบจำลองของโครงข่ายประสาทเชิงลำดับ (Recurrent Neural Network) ที่ใช้ในการจัดการกับปัญหาที่เกี่ยวข้องกับลำดับของข้อมูล เช่น การแปลภาษาธรรมชาติหรือการตรวจจับคำพูด

GRU ถูกพัฒนาขึ้นเพื่อแก้ปัญหาคความยาวของลำดับข้อมูลที่ส่งผลให้ RNN มีปัญหาในการเรียนรู้และส่งผลลัพธ์ที่ถูกต้อง เปรียบเทียบกับ LSTM ที่มีโครงสร้างที่ซับซ้อนมากขึ้น GRU ถูกออกแบบมาเพื่อแก้ไขบางปัญหาและความซับซ้อนของ LSTM ในทางทฤษฎีและการปฏิบัติ GRU เป็นโครงสร้างที่มีความซับซ้อนน้อยกว่า LSTM แต่ยังสามารถแก้ปัญหาลำดับข้อมูลได้ดี โดยมีการลดจำนวนเกตเวย์ในโครงสร้าง ทำให้ GRU มีประสิทธิภาพในการเรียนรู้และประมวลผลที่ดีกว่า RNN แบบพื้นฐาน

2.13.4 โครงสร้างของ Gated Recurrent Unit (GRU) [23]

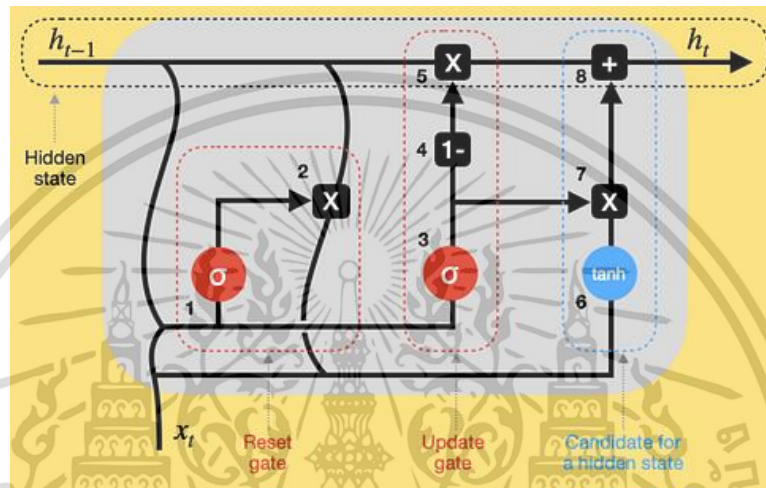
โครงสร้างของ Gated Recurrent Unit (GRU) ประกอบด้วยส่วนต่าง ๆ ดังนี้

Reset Gate เป็นเกตเวย์ที่ใช้ในการควบคุมการลืมข้อมูลจากลำดับก่อนหน้าเพื่อตัดสินใจเกี่ยวกับข้อมูลที่จะถูกลบออก

Update Gate เป็นเกตเวย์ที่ใช้ในการควบคุมการอัปเดตข้อมูลในลำดับปัจจุบัน โดยการเลือกที่จะเพิ่มข้อมูลใหม่เข้าไปในลำดับปัจจุบัน

Hidden State เป็นส่วนที่เก็บข้อมูลที่สำคัญในระยะสั้น โดยมีการควบคุมการอัปเดตข้อมูลโดยเกตเวย์รีเซ็ตและเกตเวย์ปรับปรุง

Output เป็นผลลัพธ์ที่ได้จากการประมวลผลของ GRU ดังภาพที่ 2.17



ภาพที่ 2.17 สถาปัตยกรรมของ GRU [23]

โดยสรุป GRU คือโครงสร้างที่ใช้ในการจัดการข้อมูลลำดับในเวลา โดยมีการควบคุมการลบข้อมูลจากสถานะก่อนหน้าและการเลือกใช้ข้อมูลใหม่ ๆ ในการอัปเดตสถานะ มีการใช้ส่วนของเครื่องหมายคูณและฟังก์ชันเชิงเส้นในการทำงานของ GRU เพื่อคำนวณค่าและควบคุมการทำงานของโครงสร้าง

2.14 Model ทั้ง 10 ของ Machine Learning ที่ใช้ในการทดลอง

2.14.1 K-Nearest Neighbors (KNN) [24]

K-Nearest Neighbors (KNN) คืออัลกอริทึมในการเรียนรู้แบบความเหมือนใกล้ที่สุด ที่ใช้ในการจัดกลุ่มหรือจัดประเภทข้อมูล โดยอิงความคล้ายคลึงกันของข้อมูลในตัวอย่งการเรียนรู้ และใช้ในการทำนายคลาสของข้อมูลที่ไม่รู้ผลลัพธ์

KNN ทำงานโดยการหาเพื่อนบ้านที่ใกล้ที่สุด (Nearest Neighbors) ของข้อมูลที่ต้องการจัดประเภท โดยใช้วิธีการคำนวณระยะทางระหว่างข้อมูล และจากนั้นใช้โหวตจากเพื่อนบ้านที่ใกล้ที่สุดเพื่อกำหนดประเภทของข้อมูลที่ต้องการจัดประเภท เมื่อมีข้อมูลใหม่เข้ามา โมเดล KNN จะหาเพื่อนบ้านที่ใกล้ที่สุดของข้อมูลใหม่นั้น และใช้โหวตเพื่อกำหนดประเภทของข้อมูลใหม่

KNN เป็นอัลกอริทึมที่ไม่ต้องการการเรียนรู้ล่วงหน้า แต่ต้องคำนึงถึงการคำนวณระยะทางระหว่างข้อมูลทุกครั้งที่มีข้อมูลใหม่เข้ามา ซึ่งอาจใช้เวลานานในกรณีที่มีข้อมูลมาก และยังคงต้องคำนึงถึงการเลือกค่า K (จำนวนเพื่อนบ้านที่ใช้ในการโหวต) ที่เหมาะสม

2.14.2 โครงสร้าง K-Nearest Neighbors (KNN)

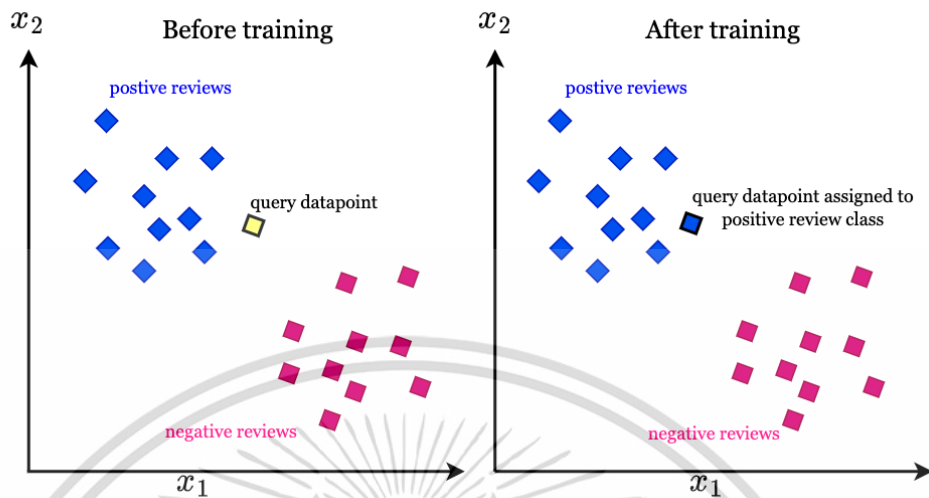
โครงสร้างของ K-Nearest Neighbors (KNN) โดยตัวอัลกอริทึมจะใช้ข้อมูลในชุดการฝึก (Training Dataset) ในการจำแนกหรือทำนายข้อมูลใหม่โดยตรง โครงสร้างพื้นฐานของ KNN ประกอบด้วยขั้นตอนดังนี้

เลือกค่า K ก่อนที่จะใช้ KNN จำเป็นต้องเลือกค่า K ที่เหมาะสม เรียก K ว่า "จำนวนเพื่อนบ้านที่ใกล้ที่สุด" (Nearest Neighbors) ซึ่งเป็นจำนวนของข้อมูลที่จะถูกใช้ในการตัดสินใจ ค่า K ที่เลือกจะมีผลต่อความแม่นยำของการทำนาย โดยค่า K ที่น้อยอาจทำให้โมเดลอยู่ในขนาดของข้อมูลตัวอย่างและความสัมพันธ์ซับซ้อน ในขณะที่ค่า K ที่มากอาจทำให้โมเดลมีความสัมพันธ์กับข้อมูลที่น้อยลง และค่า K ที่เหมาะสมจะต้องถูกคำนวณให้เหมาะสมกับงานที่มีอยู่

คำนวณระยะห่าง เพื่อหาข้อมูลที่ใกล้เคียงที่สุด K ข้อมูล จะใช้หลักการวัดระยะห่าง (Distance Metric) เช่น ระยะห่างยูคลิเดียน (Euclidean Distance) หรือระยะห่างแมนฮัตตัน (Manhattan Distance) เพื่อคำนวณระยะห่างระหว่างข้อมูลในชุดการฝึกและข้อมูลที่ต้องการจำแนกหรือทำนาย

ค้นหา K ข้อมูลที่ใกล้เคียง หลังจากคำนวณระยะห่างระหว่างข้อมูล จะเลือก K ข้อมูลที่มีระยะห่างน้อยที่สุดจากข้อมูลที่ต้องการจำแนกหรือทำนาย ข้อมูลเหล่านี้จะเป็น "เพื่อนบ้านที่ใกล้ที่สุด" (Nearest Neighbors) ของข้อมูลตัวอย่าง

การตัดสินใจโดยส่วนใหญ่ (Majority Voting) หลังจากที่ได้ K ข้อมูลที่ใกล้เคียง จะทำการตัดสินใจโดยการประมวลผลผ่านกระบวนการโหวต (Voting) โดยเลือกคลาสหรือประเภทข้อมูลที่มีความถี่มากที่สุดเป็นผลลัพธ์ ดังภาพที่ 2.18



ภาพที่ 2.18 KNN อัลกอริทึม [24]

2.14.3 Support Vector Machines (SVM)

Support Vector Machines (SVM) [25] คืออัลกอริทึมในการเรียนรู้ของเครื่องจักร (Machine Learning) ที่ใช้ในการจำแนกและวิเคราะห์ข้อมูล โดยการสร้างแบ่งกลุ่ม (Classification) หรือพยากรณ์ (regression) โดยใช้การเรียนรู้จากตัวอย่างที่มีการตั้งค่าเริ่มต้น (Training set) ของข้อมูลที่ถูกตั้งค่าไว้ล่วงหน้า

SVM ทำงานโดยการสร้างเส้นแบ่งกลุ่มหรือเส้นแบ่งเชิงเส้น (Hyperplane) ในตัวแบบหลายมิติของข้อมูล โดยพยากรณ์กลุ่มที่ไม่เหมือนกัน (หรือค่าต่อไป) และสร้างเส้นที่เหมาะสมที่สุดในการแบ่งกลุ่มข้อมูล ทำให้ SVM เป็นวิธีการจำแนกที่มีประสิทธิภาพสูงในการแบ่งกลุ่มข้อมูลที่ซับซ้อนและมีขนาดใหญ่

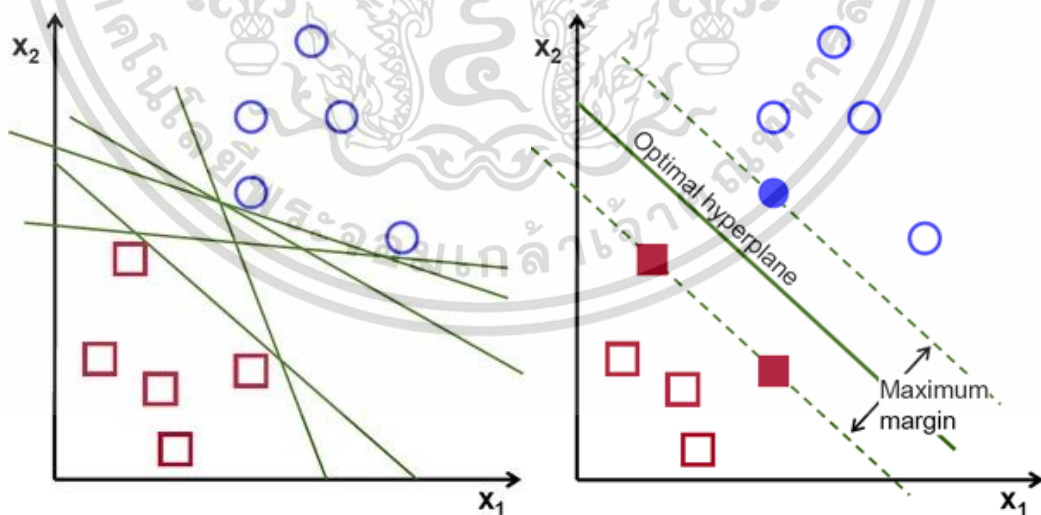
SVM ใช้หลักการหาเส้นแบ่งกลุ่มที่มีระยะห่างระหว่างกลุ่มข้อมูล (Margin) ที่มากที่สุด โดยการสร้างเส้นแบ่งกลุ่มที่อยู่ห่างจากข้อมูลใกล้ที่สุดในแต่ละกลุ่ม และมีระยะห่างระหว่างเส้นแบ่งกลุ่มและข้อมูลใกล้เคียง (Support Vectors) ที่มากที่สุด

2.14.4 โครงสร้าง Support Vector Machines (SVM)

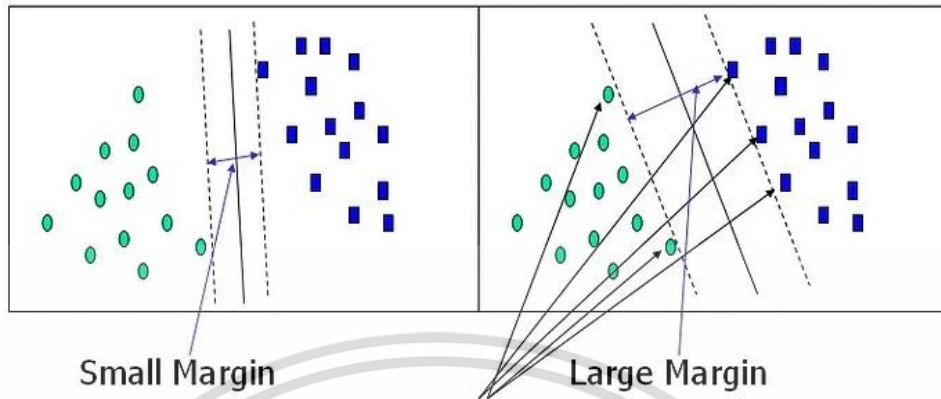
Support Vector Machines (SVM) เป็นอัลกอริทึมที่ใช้ในการจำแนกข้อมูล โดยหลักการของ SVM คือการหา hyperplane ที่เหมาะสมที่สุดในการแบ่งกลุ่มข้อมูล

โครงสร้างของ SVM ประกอบด้วยส่วนต่าง ๆ ดังนี้

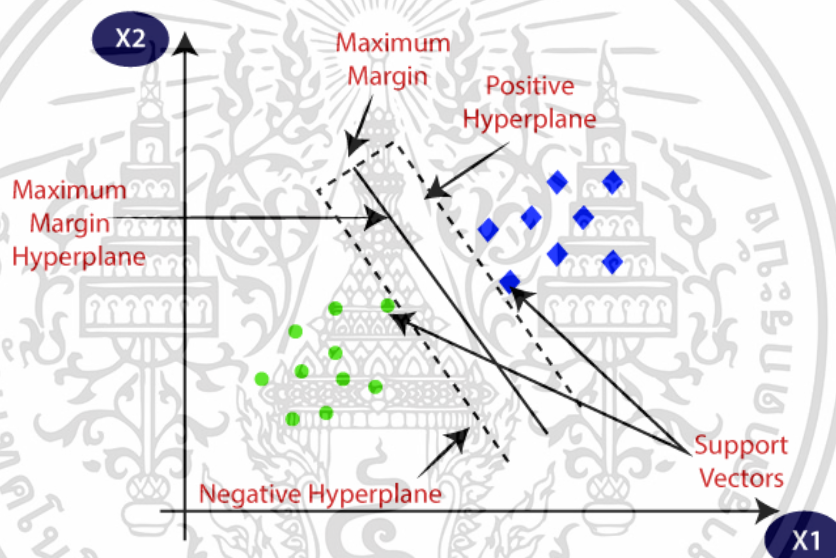
- Training Set เป็นชุดข้อมูลที่ใช้ในการเรียนรู้และปรับค่าพารามิเตอร์ของ SVM เพื่อหา hyperplane ที่เหมาะสมที่สุดในการแบ่งกลุ่มข้อมูล
- Feature Extraction เป็นกระบวนการที่นำข้อมูลเข้ามาแปลงเป็นรูปแบบที่เหมาะสมสำหรับการจำแนก เช่น การเลือกและสกัดคุณลักษณะ (Features) ที่เกี่ยวข้องกับการแบ่งกลุ่มข้อมูล
- Kernel Function เป็นฟังก์ชันที่ใช้ในการแปลงข้อมูลเข้าสู่มิติสูง (High-Dimensional space) เพื่อให้สามารถแบ่งกลุ่มข้อมูลได้โดยเส้นแบ่งเส้นเดียวหรือเส้นแบ่งหลายเส้น ฟังก์ชันเคอเนลที่ใช้บ่อยคือเคอเนลเริ่มต้น (Linear Kernel) และเคอเนลที่มีการแปลงเข้าสู่มิติสูง เช่น เคอเนลหลายเหลี่ยม (Polynomial Kernel) และเคอเนลเรดิอัล (Radial Basis Function Kernel)
- Margin เป็นระยะห่างระหว่าง Hyperplane และข้อมูลใกล้เคียงที่สุดในแต่ละกลุ่ม (Support Vectors) ซึ่ง SVM จะหา Hyperplane ที่มี Margin ที่สูงที่สุด เพื่อให้การแบ่งกลุ่มข้อมูลมีความแม่นยำสูงสุด
- Support Vectors เป็นข้อมูลใกล้เคียงที่สุดกับ Hyperplane ในแต่ละกลุ่ม ซึ่งเป็นข้อมูลที่มีผลต่อการปรับค่าพารามิเตอร์ของ SVM และเป็นส่วนสำคัญในการคำนวณ Hyperplane
- Decision Boundary เป็นเส้นแบ่งกลุ่มที่ได้จากการคำนวณ Hyperplane ใน SVM เส้นแบ่งกลุ่มนี้จะแบ่งกลุ่มข้อมูลออกเป็นสองกลุ่ม ในกรณีของการจำแนกหลายกลุ่ม SVM จะใช้หลายเส้นแบ่งกลุ่มเพื่อแยกแยะกลุ่มข้อมูล ดังภาพที่ 2.19, 2.20 และ 2.21



ภาพที่ 2.19 Possible hyperplanes [25]



ภาพที่ 2.20 Support Vectors [25]



ภาพที่ 2.21 SVM อัลกอริทึม [25]

2.14.5 Random Forest

Random Forest [26] คืออัลกอริทึมในการเรียนรู้ของเครื่องจักร (Machine Learning) ที่ใช้ในการจัดกลุ่มหรือทำนายข้อมูล โดยทำการสร้างประเภทของต้นไม้ (Decision Tree) หลายๆ ต้น แล้วรวมกันเป็นป่าไม้ แล้วทำการสุ่มตัวอย่างของข้อมูลที่ใช้ในการสร้างต้นไม้แต่ละต้น และใช้หลายๆ ต้นไม้ร่วมกันในการทำนายผลลัพธ์ ซึ่งจะถือว่าเป็นผลลัพธ์ที่ถูกต้องเมื่อมีจำนวนต้นไม้ที่มีความถูกต้องสูงสุด โดย Random Forest สามารถนำไปใช้ในการจัดกลุ่ม (Classification) หรือการทำนาย (Prediction) ข้อมูลได้ในหลากหลายงาน

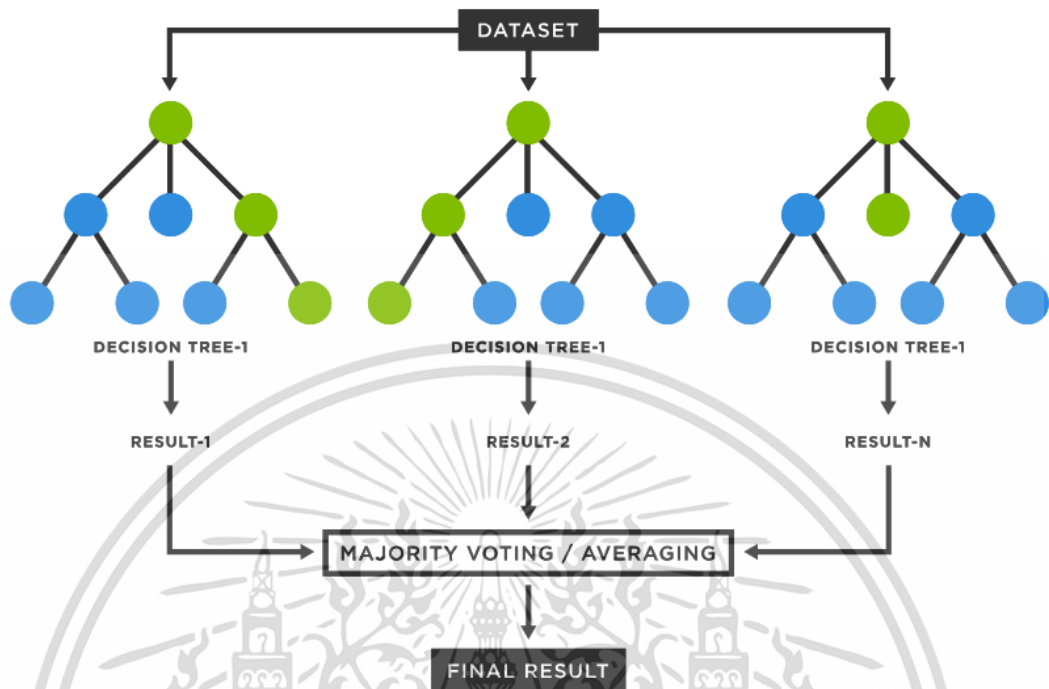
Random Forest ถูกเรียกว่า "Random" เพราะความหลากหลาย (Variety) และความสุ่ม (Randomness) เข้ามาในกระบวนการการเรียนรู้ ซึ่งทำให้เป็นหนึ่งในอัลกอริทึมที่ทนทานและมีประสิทธิภาพในการจำแนกข้อมูลที่มีขนาดใหญ่และซับซ้อน

โดยการสร้าง Random Forest จะสร้าง Decision Tree หลายๆ ต้น แล้วใช้การสุ่มตัวอย่างในการสร้าง Decision Tree แต่ละต้น เพื่อลดความผิดพลาดที่อาจเกิดขึ้นจากการ Overfitting โดยโมเดลจะใช้การ Voting หรือการเฉลี่ยผลลัพธ์จาก Decision Tree แต่ละต้น เพื่อให้ได้ผลลัพธ์ที่มีความแม่นยำและเสถียรมากกว่า

2.14.6 โครงสร้าง Random Forest

โครงสร้างของ Random Forest ประกอบด้วย Decision Tree หลายต้นที่สร้างจากการสุ่มตัวอย่างของข้อมูล โดยแต่ละ Decision Tree จะสร้างขึ้นโดยใช้ข้อมูลที่สุ่มมาจากตัวอย่างโดยไม่ซ้ำกัน โดยมีขั้นตอนการสร้าง Random Forest ดังนี้

- สุ่มตัวอย่างจากชุดข้อมูลที่มีอยู่ สุ่มตัวอย่างข้อมูลมาจากนั้นใช้ตัวอย่างนั้นในการสร้างต้นไม้แต่ละต้น
- Build a Tree สร้างต้นไม้โดยใช้ตัวอย่างข้อมูลที่สุ่มมาจากขั้นตอนที่ 1 โดยการสร้างต้นไม้สามารถใช้วิธีการสร้างต้นไม้ใดก็ได้ เช่น Decision Tree, CART, C4.5 เป็นต้น
- สร้าง Random Forest คือการทำซ้ำขั้นตอนที่ 1-2 เพื่อสร้างต้นไม้หลายต้น โดยจำนวนต้นไม้ที่จะสร้างเป็นไปตามค่าที่กำหนด
- ทำนายผลเมื่อ Random Forest สร้างเสร็จสิ้น สามารถนำข้อมูลที่ไม่เคยใช้ในการสร้างต้นไม้มาทำนายผลได้ โดยให้แต่ละต้นไม้ให้คะแนนในการทำนาย แล้วเลือกผลลัพธ์ที่ได้มากที่สุดหรือมากที่สุดในกลุ่ม (สำหรับงานการจำแนก) หรือเฉลี่ยผลลัพธ์ (สำหรับงานการทำนายตัวเลขหรือการคาดการณ์) ดังภาพที่ 2.22



ภาพที่ 2.22 Random Forest อัลกอริทึม [26]

ในตัวอย่างข้างต้น มี Random Forest ที่ประกอบด้วย 3 Decision Tree และแต่ละ Decision Tree สร้างจากการสุ่มตัวอย่างข้อมูลที่ไม่ซ้ำกัน ดังนั้นแต่ละ Decision Tree จะมีความแตกต่างกันไป และผลลัพธ์ที่ได้จากการทำนายของแต่ละ Decision Tree จะถูกนำมาใช้ในการตัดสินใจสุดท้าย

2.14.7 Decision Tree

Decision Tree [27] เป็นวิธีการสร้างโมเดลทางสถิติและการเรียนรู้ของเครื่อง (Machine Learning) ที่ใช้ในการจำแนกและทำนาย โดยทำงานตามหลักการวิเคราะห์ทางตรรกศาสตร์ โดยการสร้างต้นไม้ที่มีโหนด (Node) และสาขา (Branch) ซึ่งแทนคุณลักษณะ (Feature) และค่าของคุณลักษณะนั้น ๆ ตามลำดับ โดย Decision Tree จะทำการแบ่งข้อมูลออกเป็นกลุ่มย่อยๆ โดยใช้คุณลักษณะที่มีความสำคัญในการแยกกลุ่ม โดยการใช้เกณฑ์ต่าง ๆ เช่น ค่า Gini index, ค่า Information Gain, หรือค่า Chi-squared เป็นต้น จากนั้น Decision Tree จะสร้างกิ่ง (Branch) สำหรับแต่ละค่าของคุณลักษณะที่ใช้ในการแยกกลุ่ม และทำซ้ำกระบวนการนี้จนกว่าจะได้กลุ่มย่อยที่มีความสมดุลและสอดคล้องกับเกณฑ์ที่กำหนด

เมื่อ Decision Tree สร้างเสร็จสิ้น สามารถนำข้อมูลใหม่ที่ไม่เคยใช้ในการสร้างต้นไม้มาทำนายผลได้ โดยการผ่านผลลัพธ์ที่ได้จากการเดินลงมาตามกิ่ง (Branch) ตามค่าของคุณลักษณะที่ตรวจสอบ จนถึงใบ (Leaf) ที่แสดงผลลัพธ์ของการจำแนกหรือการทำนาย

2.14.8 โครงสร้าง Decision Tree

โครงสร้างของ Decision Tree ประกอบด้วยโหนด (Node) และสาขา (Branch) ซึ่งแทนคุณลักษณะ (Feature) และค่าของคุณลักษณะนั้น ๆ ตามลำดับ โดยมีโครงสร้างหลักดังนี้

- Root Node เป็นโหนดแรกที่เริ่มต้นของ Decision Tree ซึ่งอยู่ที่ระดับบนสุด โหนดนี้จะไม่ มีสาขาเข้ามา แต่จะมีสาขาออกไปยังโหนดย่อย ๆ ต่อไป

- Internal Node เป็นโหนดที่ไม่ใช่ Root Node และไม่ใช่ Leaf Node (โหนดใบ) โหนดเหล่านี้จะมีสาขาเข้ามาจากโหนดที่อยู่ด้านบน และสาขาออกไปยังโหนดย่อย ๆ ต่อไป โหนดย่อยเหล่านี้จะใช้ในการแบ่งข้อมูลออกเป็นกลุ่มย่อย โดยใช้เกณฑ์ที่กำหนดโดยอัลกอริทึม เช่น Gini index, Information Gain, หรือ Chi-squared

- Leaf Node เป็นโหนดที่ไม่มีสาขาออกไปยังโหนดย่อย โหนดเหล่านี้แทนผลลัพธ์ของการจำแนกหรือการทำนาย โดยส่วนใหญ่จะเป็นคลาสหรือค่าที่ต้องการทำนาย เช่น ในกรณีของการจำแนกกลุ่มของข้อมูล โหนดใบจะแทนกลุ่มข้อมูลที่อยู่ในกลุ่มนั้น

โดยทั่วไปการสร้าง Decision Tree ประกอบด้วยขั้นตอนดังนี้

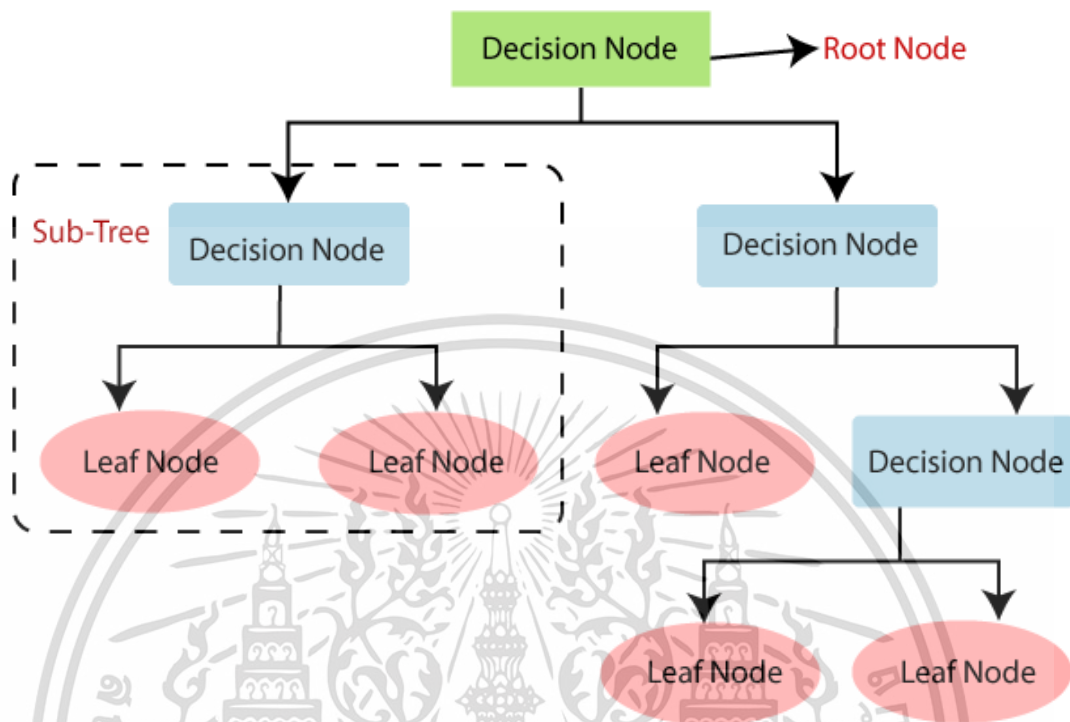
- การเลือกคุณลักษณะ (Feature Selection) คือเลือกคุณลักษณะที่เหมาะสมสำหรับการแบ่งข้อมูลแต่ละรายการ

- การแบ่งข้อมูล (Data Splitting) คือใช้เงื่อนไขที่เลือกมาจากขั้นตอนที่ 1 เพื่อแบ่งข้อมูลออกเป็นกลุ่มย่อย ๆ โดยเริ่มจากโหนดรากและเดินลงมาที่โหนดใบ

- การสร้างโหนดใหม่ (Node Creation) คือสร้างโหนดใหม่ในต้นไม้โดยใช้เงื่อนไขที่เลือกในขั้นตอนที่ 2

- การทำซ้ำ (Recursion) คือทำซ้ำขั้นตอนที่ 2-3 สำหรับกลุ่มย่อยของข้อมูลในแต่ละโหนด จนกว่าจะถึงเงื่อนไขสิ้นสุด เช่น การเลือกไม่มีคุณลักษณะให้แยกอีกต่อไป หรือถึงจำนวนโหนดสูงสุดที่กำหนดไว้ล่วงหน้า

- การกำหนดผลลัพธ์ (Leaf Node Assignment) คือ Leaf Node จะกำหนดผลลัพธ์หรือคำตอบของการตัดสินใจขึ้นอย่างเหมาะสมสำหรับคำตอบ ดังภาพที่ 2.23



ภาพที่ 2.23 Decision Tree อัลกอริทึม [27]

2.14.9 Gaussian Naive Bayes

Gaussian Naive Bayes [28] เป็นอัลกอริทึมในการจัดกลุ่มหรือจำแนกข้อมูลในปัญหาที่เป็นปัญหาการจำแนกแบบ Supervised Learning โดยใช้หลักการของ Naive Bayes และประมวลผลด้วยการประมาณการแบบ Gaussian Distribution (Normal Distribution) โดยมีการแบ่งคลาสของข้อมูลเป็น 2 กลุ่ม คือ Positive และ Negative โดยมีการใช้ Gaussian Distribution เพื่อคำนวณความน่าจะเป็นของแต่ละคลาส

โดย Gaussian Naive Bayes จะทำการคำนวณความน่าจะเป็นของแต่ละคลาสโดยใช้ Gaussian Distribution ในการคำนวณ โดยจะคำนวณหา Mean และ Standard Deviation ของแต่ละ Feature ในแต่ละคลาส จากนั้นจึงนำ Mean และ Standard Deviation ที่ได้มาใช้ในการคำนวณ Gaussian Distribution เพื่อหาความน่าจะเป็นของแต่ละคลาส

Gaussian Naive Bayes เหมาะสำหรับการจัดการกับข้อมูลที่มี Feature จำนวนมาก และเหมาะสำหรับงาน Classification โดยเฉพาะที่ข้อมูลมีการกระจายตัวเป็น Normal Distribution

2.14.10 โครงสร้างของ Gaussian Naive Bayes

โครงสร้างของ Gaussian Naive Bayes ไม่มีโหนดและเส้นเชื่อมเหมือนกับ Decision Tree หรือโครงสร้างทางกราฟอื่น ๆ แต่เป็นแบบจำลองทางสถิติที่ใช้ในการจำแนกประเภท (Classification) ข้อมูลโดยมีกลไกการทำงานดังนี้

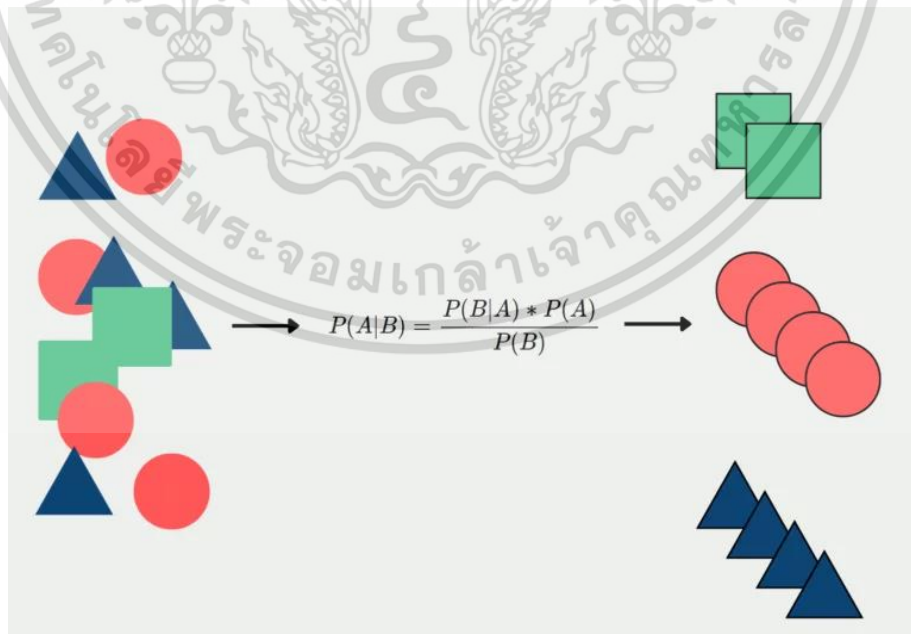
- การเตรียมข้อมูล (Data Preparation) ในขั้นตอนนี้จะทำการเตรียมข้อมูลให้พร้อมสำหรับการประมวลผล ซึ่งรวมถึงการแบ่งข้อมูลเป็นชุดฝึกฝน (Training set) และชุดทดสอบ (Test Set) และการทำ Scaling หรือ Normalization ของข้อมูลถ้าจำเป็น

- การคำนวณค่าเฉลี่ยและค่าเบี่ยงเบนมาตรฐาน (Mean and Standard Deviation Calculation) ในขั้นตอนนี้จะทำการคำนวณค่าเฉลี่ยและค่าเบี่ยงเบนมาตรฐานของคุณลักษณะที่เกี่ยวข้องกับแต่ละคลาสโดยใช้ Gaussian Distribution

- การคำนวณความน่าจะเป็น (Probability Calculation) ในขั้นตอนนี้จะใช้ค่าเฉลี่ยและค่าเบี่ยงเบนมาตรฐานที่คำนวณได้จากขั้นตอนก่อนหน้าเพื่อคำนวณความน่าจะเป็นที่ข้อมูลจะอยู่ในแต่ละคลาส โดยใช้ Gaussian Distribution

- การจำแนกข้อมูล (Classification) ในขั้นตอนสุดท้ายจะใช้ค่าความน่าจะเป็นที่คำนวณได้จากขั้นตอนก่อนหน้าเพื่อเลือกคลาสที่มีความน่าจะเป็นสูงสุดเป็นคำตอบในการจำแนกข้อมูล

Gaussian Naive Bayes สามารถคำนวณได้โดยใช้สูตรต่อไปนี้ดังภาพที่ 2.24



ภาพที่ 2.24 การจำแนกประเภท Gaussian Naive Bayes [28]

โดยมีรายละเอียดดังนี้

$P(B|A)$ = ความน่าจะเป็นที่เหตุการณ์ B จะเกิดขึ้นหากเหตุการณ์ A เกิดขึ้นแล้ว

$P(A)$ = ความน่าจะเป็นที่เหตุการณ์ A จะเกิดขึ้น

$P(B)$ = ความน่าจะเป็นที่เหตุการณ์ B จะเกิดขึ้น

สรุปโครงสร้างของ Gaussian Naive Bayes จะมีการคำนวณค่าเฉลี่ยและค่าเบี่ยงเบนมาตรฐานในขั้นตอนที่ 2 และใช้ค่าเฉลี่ยและค่าเบี่ยงเบนมาตรฐานในขั้นตอนที่ 3 เพื่อคำนวณความน่าจะเป็นของข้อมูลในแต่ละคลาส และในขั้นตอนที่ 4 จะใช้ค่าความน่าจะเป็นเพื่อจำแนกข้อมูลในคลาสที่มีความน่าจะเป็นสูงสุด

2.14.11 Logistic Regression

Logistic Regression [29] เป็นอัลกอริทึมในการเรียนรู้แบบจำแนกประเภท (Classification) ที่ใช้ในการพยากรณ์ความน่าจะเป็นของผลลัพธ์ที่เป็นไปได้สำหรับตัวแปรต้น (Independent Variables) ที่มีค่าเป็นตัวแปรต่อเนื่อง (Dependent Variable) ที่มีค่าเป็นตัวแปรสองค่า (Binary Variable) หรือตัวแปรหลายค่าที่เป็นตัวแปรอิสระ (Multinomial Variable)

Logistic Regression ใช้ฟังก์ชันสัมพันธ์ทางคณิตศาสตร์ที่เรียกว่าฟังก์ชันสัมพันธ์ โลจิสติก (logistic function) หรือฟังก์ชันสัมพันธ์ทางคณิตศาสตร์ Sigmoid เพื่อแปลงค่าเอาต์พุตของอัลกอริทึมเป็นค่าความน่าจะเป็น (Probability) ที่ผลลัพธ์จะอยู่ในกลุ่มหนึ่งหรือกลุ่มอื่น ๆ

การทำงานของ Logistic Regression คือการหาค่าพารามิเตอร์ที่เหมาะสมในฟังก์ชันสัมพันธ์โลจิสติก เพื่อให้โมเดลสามารถทำนายค่าความน่าจะเป็นของผลลัพธ์ได้ถูกต้องที่สุด โดยใช้วิธีการเรียนรู้แบบความแตกต่างเชิงเส้น (Gradient descent) หรือวิธีการเรียนรู้แบบอื่น ๆ เช่น Newton-Raphson หรือ Fisher scoring

คุณลักษณะหลักของ Logistic Regression คือการใช้ฟังก์ชัน Sigmoid (หรือ Logistic Function) เพื่อแปลงผลลัพธ์ให้อยู่ในช่วงระหว่าง 0 ถึง 1 ซึ่งจะแทนความน่าจะเป็นของคลาสหรือหมวดหมู่ 1 ในการจำแนกประเภทและคลาส 0 ในการจำแนกประเภทอีกหนึ่งอย่าง ฟังก์ชัน Sigmoid มีสมการดังต่อไปนี้ (สมการที่ 1)

$$\text{Sigmoid}(z) = \frac{1}{1+e^{-z}} \quad [29] \quad (1)$$

โดยที่ z คือผลรวมของน้ำหนัก (weights) ของคุณลักษณะ (features) และค่าคงที่ (bias) ซึ่งกำหนดให้มีค่าเป็น $Z = \omega_0 + \omega_1 x_1 + \omega_2 x_2 + \dots + \omega_n x_n$ [29]

z คือผลรวมของน้ำหนักและคุณลักษณะของข้อมูล

ω_0 คือค่าคงที่ (bias)

$\omega_1, \omega_2, \dots, \omega_n$ คือน้ำหนัก (weights) ที่ถูกปรับให้กับคุณลักษณะแต่ละตัว

x_1, x_2, \dots, x_n คือคุณลักษณะ (features) ของข้อมูล

2.14.12 โครงสร้าง Logistic Regression

โครงสร้างของ Logistic Regression ประกอบด้วยส่วนสำคัญต่อไปนี้

- ฟังก์ชันสัมพันธ์โลจิสติก (Logistic Function) เป็นฟังก์ชันที่ใช้ในการแปลงผลลัพธ์ของโมเดลเป็นค่าความน่าจะเป็น มักใช้ฟังก์ชันสัมพันธ์โลจิสติกแบบ Sigmoid

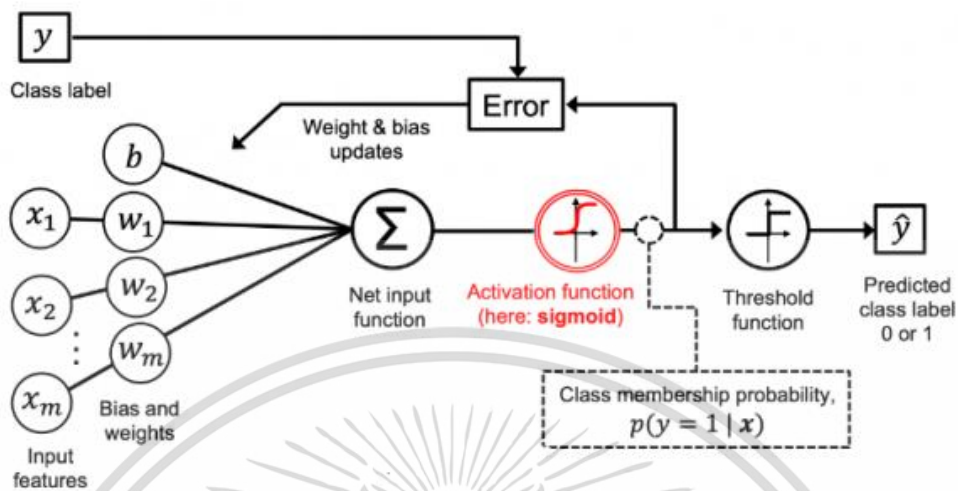
- พารามิเตอร์ (Parameters) เป็นค่าที่ต้องปรับเพื่อให้โมเดล Logistic Regression สามารถประมาณค่าความน่าจะเป็นของผลลัพธ์ได้ถูกต้อง พารามิเตอร์ที่สำคัญใน Logistic Regression คือ ค่าความน่าจะเป็นเริ่มต้น (Intercept) และค่าความสัมพันธ์ของตัวแปรต้น (Independent Variables)

- ฟังก์ชันค่าความสัมพันธ์ (Link Function) เป็นฟังก์ชันที่ใช้ในการเชื่อมโยงค่าความน่าจะเป็นกับตัวแปรต้นใน Logistic Regression จะใช้ฟังก์ชันลอจิท (Logit Function) ซึ่งมีสมการคือ

$$\text{logit}(p) = \ln \frac{P}{(1-P)} \quad [29]$$

- ฟังก์ชันสูญเสีย (Loss Function) เป็นฟังก์ชันที่ใช้ในการวัดความคลาดเคลื่อนระหว่างค่าความน่าจะเป็นจริงและค่าความน่าจะเป็นที่โมเดลทำนายได้ ฟังก์ชันสูญเสียที่ใช้ใน Logistic Regression มักเป็น Cross-Entropy Loss หรือ Log Loss

วิธีการเรียนรู้ (Learning Algorithm) เป็นวิธีการปรับค่าพารามิเตอร์ในโมเดล Logistic Regression เพื่อให้โมเดลสามารถทำนายค่าความน่าจะเป็นได้ถูกต้องที่สุด วิธีการเรียนรู้ที่ใช้ใน Logistic Regression สามารถเป็น Gradient Descent, Newton-Raphson, Fisher Scoring, หรือวิธีการเรียนรู้อื่น ๆ ได้ตามความเหมาะสม ดังภาพที่ 2.25



ภาพที่ 2.25 Logistic Regression อัลกอริทึม [29]

2.14.13 AdaBoost

AdaBoost (Adaptive Boosting) [30] เป็นอัลกอริทึมในการเรียนรู้ของเครื่อง (Machine Learning) ที่ใช้หลักการของ Ensemble Learning โดยการเรียนรู้จะเป็นการเรียนรู้แบบ Iterative ซึ่งจะทำให้การปรับค่า Weight ของข้อมูลในแต่ละรอบเพื่อให้โมเดลทำนายได้ถูกต้องมากขึ้น

AdaBoost จะสร้างโมเดลหลายๆ ตัว โดยแต่ละโมเดลจะเป็น Weak Learner หรือโมเดลที่มีความแม่นยำต่ำ แต่สามารถทำนายได้ถูกต้องเฉพาะกรณีบางส่วน เช่น Decision Tree ที่มีความลึกต่ำ หรือ Logistic Regression ที่มีความซับซ้อนต่ำ ในแต่ละรอบของการเรียนรู้ AdaBoost จะให้ Weight ของข้อมูลในชุดข้อมูลเพิ่มขึ้นสำหรับข้อมูลที่ทำนายผิด และลด Weight สำหรับข้อมูลที่ทำนายถูกต้อง จากนั้นจึงสร้างโมเดลใหม่โดยใช้ชุดข้อมูลที่มี Weight ปรับแล้วในแต่ละรอบ และทำซ้ำไปเรื่อยๆ จนกระทั่งได้โมเดลที่มีความแม่นยำสูง

AdaBoost เหมาะสำหรับการจัดการกับข้อมูลที่มี Feature จำนวนมากและเหมาะสมสำหรับงาน Classification โดยเฉพาะที่ข้อมูลมีความซับซ้อนและไม่สมดุล

2.14.14 โครงสร้างของ AdaBoost

AdaBoost (Adaptive Boosting) เป็นเทคนิคการเรียนรู้เชิงเสริม (Ensemble Learning) ซึ่งไม่มีโครงสร้างทางกราฟเหมือนกับ Decision Tree หรือโครงสร้างแบบอื่น ๆ แต่มีกระบวนการทำงานหลักที่เป็นลำดับขั้นตอนต่อไปนี้

- กำหนดชุดข้อมูลการฝึกสอน โดยกำหนดชุดข้อมูลการฝึกสอนที่ประกอบไปด้วยตัวอย่างที่มีค่าเป้าหมาย (Label) เพื่อใช้ในการสร้างโมเดลปัญญาประดิษฐ์แบบอ่อน (Weak Learner)

สร้างโมเดลปัญญาประดิษฐ์แบบอ่อน โดยสร้างโมเดลปัญญาประดิษฐ์แบบอ่อน เช่น Decision Tree หรือ Neural Network ที่มีความสามารถในการจำแนกข้อมูลตามเงื่อนไขที่กำหนด

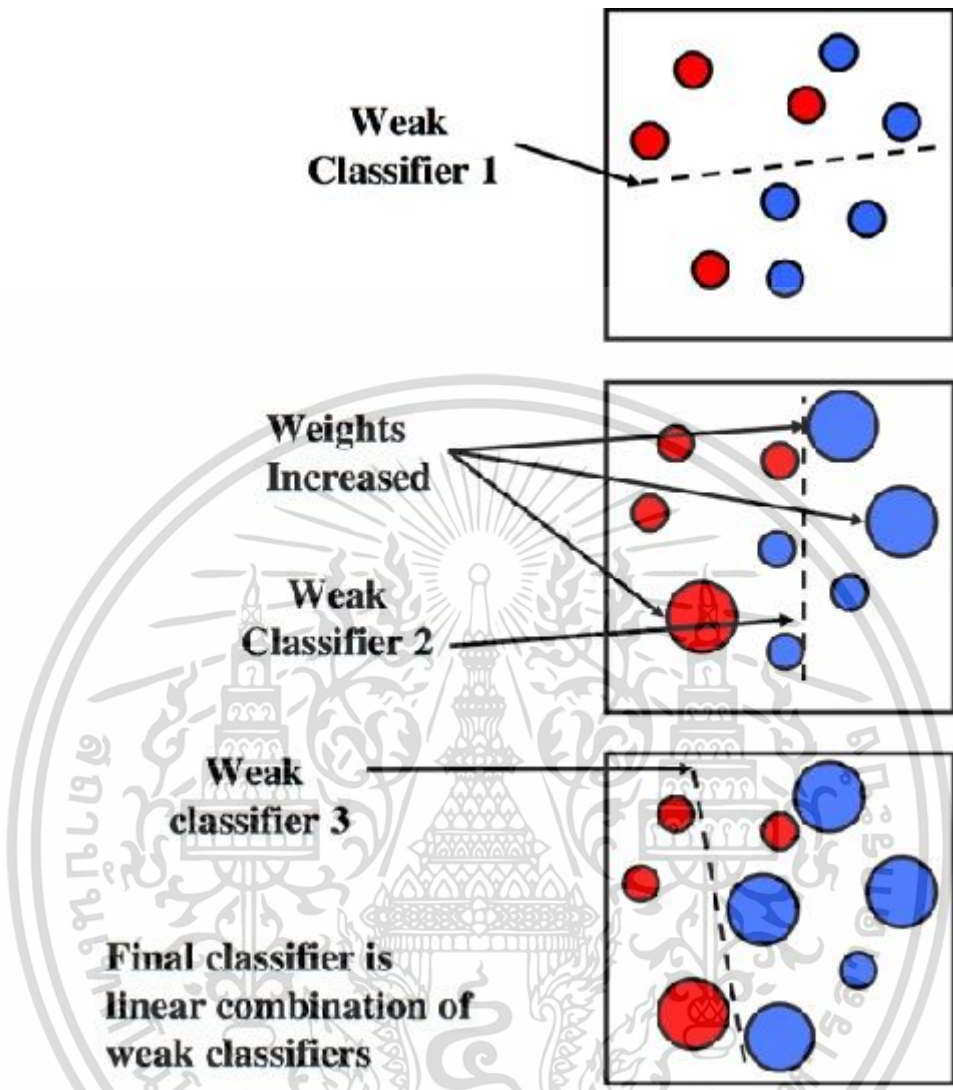
- ปรับน้ำหนักของตัวอย่าง โดยปรับน้ำหนักของตัวอย่างในชุดข้อมูลการฝึกสอนโดยการเพิ่มน้ำหนักของตัวอย่างที่โมเดลปัญญาประดิษฐ์แบบอ่อนทำนายผิดและลดน้ำหนักของตัวอย่างที่โมเดลปัญญาประดิษฐ์แบบอ่อนทำนายถูก

- คำนวณค่าความสำคัญของโมเดลปัญญาประดิษฐ์แบบอ่อน โดยคำนวณค่าความสำคัญของโมเดลปัญญาประดิษฐ์แบบอ่อนโดยใช้ค่าความผิดพลาด (Error) ของโมเดลปัญญาประดิษฐ์แบบอ่อน

- สร้างโมเดลปัญญาประดิษฐ์แบบองค์ประกอบ (Ensemble model) โดยสร้างโมเดลปัญญาประดิษฐ์แบบองค์ประกอบโดยรวมกันจากโมเดลปัญญาประดิษฐ์แบบอ่อนหลายๆ ตัว โดยให้น้ำหนักของแต่ละโมเดลปัญญาประดิษฐ์แบบอ่อนขึ้นอย่างเท่าเทียมกับค่าความสำคัญของโมเดลปัญญาประดิษฐ์แบบอ่อน

- ทำนายผล โดยใช้โมเดลปัญญาประดิษฐ์แบบองค์ประกอบที่ได้สร้างขึ้นในการทำนายผลข้อมูลใหม่

ปรับปรุงโมเดล โดยปรับปรุงโมเดลปัญญาประดิษฐ์แบบอ่อนในแต่ละรอบโดยใช้ข้อมูลการฝึกสอนและค่าความผิดพลาดของโมเดลปัญญาประดิษฐ์แบบองค์ประกอบในการปรับปรุงตัวอย่างการทำนาย ดังภาพที่ 2.26



ภาพที่ 2.26 AdaBoost อัลกอริทึม [30]

2.14.15 Stochastic Gradient Descent

Stochastic Gradient Descent (SGD) [31] คืออัลกอริทึมการเรียนรู้ของเครื่องจักรภายในสถิติที่ใช้ในการปรับค่าพารามิเตอร์ของโมเดลปัญญาประดิษฐ์แบบอ่อน (Weak Learner) ในแต่ละรอบของการฝึกสอน โดยการปรับค่าพารามิเตอร์แบบเชิงสุ่ม โดยใช้ข้อมูลตัวอย่างเพียงหนึ่งตัวอย่าง (Stochastic) ในการคำนวณค่าความผิดพลาดและปรับค่าพารามิเตอร์ วิธีนี้ช่วยให้การฝึกสอนของโมเดลปัญญาประดิษฐ์เร็วขึ้นและใช้ทรัพยากรน้อยลง โดยเฉพาะเมื่อมีข้อมูลมหาศาลหรือข้อมูลที่มีขนาดใหญ่

SGD เป็นการปรับค่าพารามิเตอร์ทีละตัวอย่างข้อมูล ไม่ใช่การคำนวณผลค่าสูญเสียสำหรับทุกตัวอย่างในชุดข้อมูล นี่คือเหตุผลที่มีคำว่า "Stochastic" ในชื่อ แต่การปรับค่าพารามิเตอร์จะมีความสัมพันธ์กับข้อมูลสัมพัทธ์ที่ถูกสุ่มออกมาจากชุดข้อมูลทั้งหมด ซึ่งทำให้การสรุปและการปรับค่าพารามิเตอร์นั้นเป็นแบบสุ่ม

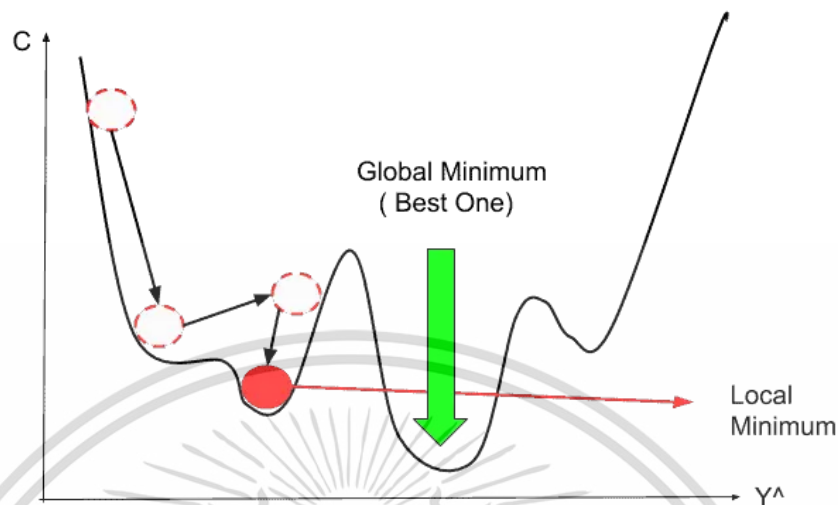
SGD จะทำการปรับค่า Weight ของโมเดลโดยใช้ Gradient ของฟังก์ชัน Cost Function ที่ต้องการจะ Minimize โดยการคำนวณ Gradient จะใช้ข้อมูลที่สุ่มมาจากชุดข้อมูลทั้งหมด (Mini-Batch) ซึ่งจะช่วยลดเวลาในการคำนวณและทำให้การปรับค่า Weight มีความสุ่มขึ้น

SGD เหมาะสำหรับการเรียนรู้ที่มีข้อมูลมหาศาล และสามารถใช้กับโมเดลที่มี Parameter จำนวนมาก เช่น Deep Neural Network แต่ SGD อาจจะไม่ใช่วิธีการที่ดีที่สุดสำหรับการเรียนรู้ในบางกรณี เช่นถ้าฟังก์ชัน Cost function เป็น Non-convex Function อาจจะทำให้เกิดปัญหา Local optima ซึ่ง SGD อาจจะไปติดอยู่ใน Local optima และไม่สามารถหา Global Optima ได้

2.14.16 โครงสร้าง Stochastic Gradient Descent

Stochastic Gradient Descent (SGD) เป็นอัลกอริทึมที่ไม่มีโครงสร้างทางกราฟเหมือนกับ Decision Tree หรือโมเดลทางกราฟอื่น ๆ แต่มีขั้นตอนการทำงานหลักที่ประกอบด้วยส่วนต่อไปนี้

- กำหนดพารามิเตอร์เริ่มต้น โดยกำหนดค่าเริ่มต้นของพารามิเตอร์ของโมเดลปัญญาประดิษฐ์แบบอ่อน
- สุ่มเลือกตัวอย่าง โดยสุ่มเลือกตัวอย่างหนึ่งตัวอย่างจากชุดข้อมูลการฝึกสอน
- คำนวณค่าความผิดพลาด โดยใช้ตัวอย่างที่ถูกสุ่มเลือกมาในขั้นตอนที่ 2 เพื่อคำนวณค่าความผิดพลาดของโมเดลปัญญาประดิษฐ์แบบอ่อน
- คำนวณค่าเกรเดียนต์ โดยคำนวณค่าเกรเดียนต์ของฟังก์ชันความผิดพลาดตามพารามิเตอร์ที่ต้องการปรับปรุง
- ปรับปรุงพารามิเตอร์ โดยใช้ค่าเกรเดียนต์ที่คำนวณได้ในขั้นตอนที่ 4 เพื่อปรับปรุงค่าพารามิเตอร์ของโมเดลปัญญาประดิษฐ์แบบอ่อน
- ตรวจสอบเงื่อนไขหยุด โดยตรวจสอบว่าเงื่อนไขหยุดการฝึกสอนถูกต้องหรือไม่ ถ้าไม่ตรงเงื่อนไข กลับไปทำซ้ำขั้นตอนที่ 2-5
- ส่งคืนโมเดลปัญญาประดิษฐ์แบบอ่อนที่ได้รับการปรับปรุงใหม่ ดังภาพที่ 2.27



ภาพที่ 2.27 Stochastic Gradient Descent อัลกอริทึม [31]

SGD มีลักษณะการทำงานแบบเพิ่มเติมอย่างหนึ่งที่เรียกว่า "Mini-batch" ซึ่งหมายถึงการสุ่มเลือกกลุ่มขนาดเล็กของตัวอย่างจากชุดข้อมูลฝึกอบรมในแต่ละรอบ การใช้ Mini-batch ช่วยลดการสั่นกระหึ่มข้อมูลและเพิ่มความเร็วในการคำนวณ Gradient โดยแบ่งชุดข้อมูลออกเป็นกลุ่มเล็ก ๆ ที่มีขนาดเท่ากับ Mini-batch ในการคำนวณ Gradient และปรับพารามิเตอร์

2.14.17 Multi-Layer Perceptron (MLP)

Multi-Layer Perceptron (MLP) [32] เป็นโมเดล Neural network ที่มีการเชื่อมต่อกันของหลายๆ ชั้น (Layer) โดยแต่ละชั้นจะประกอบด้วยหน่วยประมวลผล (Neuron) หลายๆ หน่วย แต่ละหน่วยจะมี Weight และ Bias ที่ต้องปรับเพื่อให้โมเดลสามารถทำนายผลได้ถูกต้อง

MLP เป็นโมเดลที่สามารถใช้ในการแก้ไขปัญหาที่ซับซ้อนได้โดย MLP จะมี Input Layer ที่รับข้อมูลเข้ามาและ Output Layer ที่ส่งผลลัพธ์ออกมาระหว่าง Input Layer และ Output Layer จะมี Hidden Layer หรือชั้นที่ไม่ใช่ Input Layer และ Output Layer ซึ่ง MLP สามารถมีหลายๆ Hidden Layer ได้

MLP ใช้ Activation Function เพื่อให้ Output ของแต่ละ Neuron อยู่ในช่วงที่ต้องการ เช่น Sigmoid Function, ReLU Function, Tanh Function เป็นต้น โดย Activation Function จะทำการคำนวณค่า Output ของ Neuron จาก Input และ Weight ของ Neuron ในการเรียนรู้

MLP จะใช้ Backpropagation Algorithm ในการปรับค่า Weight และ Bias เพื่อให้โมเดลสามารถทำนายผลได้ถูกต้องมากขึ้น

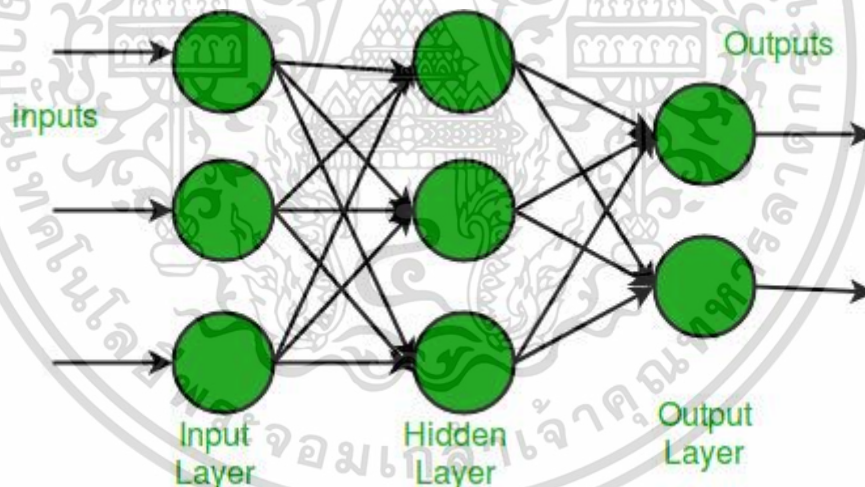
2.14.18 โครงสร้าง Multi-Layer Perceptron (MLP)

โครงสร้างของ Multi-Layer Perceptron (MLP) ประกอบด้วยหลายชั้นที่เรียกว่าชั้น (Layers) แต่ละชั้นมีเซลล์ประสาทหลายโหนด (Neurons) หรือโหนด โครงสร้างทั่วไปของ MLP ซึ่งประกอบด้วย 3 ชั้นหลักดังนี้

ชั้นนำเข้า (Input Layer) โดยชั้นแรกของ MLP ซึ่งรับข้อมูลเข้าสู่โมเดล แต่ละเซลล์ในชั้นนี้ จะรับข้อมูลเข้ามาแบบแยกตามคุณลักษณะหรือแบบเวกเตอร์ และไม่มีการประมวลผล

ชั้นซ่อน (Hidden Layers) โดยชั้นกลางที่อยู่ระหว่างชั้นนำเข้าและชั้นส่งออก แต่ละชั้นซ่อน ประกอบด้วยเซลล์ประมวลผล (Perceptron) ที่รับข้อมูลจากชั้นก่อนหน้าและประมวลผลด้วยฟังก์ชัน เชื่อมโยง (Activation Function) เพื่อสร้างความซับซ้อนในโมเดล

ชั้นส่งออก (Output Layer) โดยชั้นสุดท้ายของ MLP ซึ่งให้ผลลัพธ์ของโมเดล แต่ละเซลล์ใน ชั้นนี้จะแทนผลลัพธ์ที่ต้องการจากโมเดล เช่น การจำแนกหมวดหมู่หรือการทำนายค่า ดังภาพที่ 2.28



ภาพที่ 2.28 Multi-Layer Perceptron (MLP) อัลกอริทึม [32]

โครงสร้างของ MLP สามารถมีชั้นซ่อนได้หลายชั้น ซึ่งจำนวนชั้นซ่อนและจำนวนเซลล์ ประมวลผลในแต่ละชั้นจะขึ้นอยู่กับความซับซ้อนของปัญหาที่ต้องการแก้ไข การตั้งค่าและการ ปรับแต่งโครงสร้างของ MLP เป็นหนึ่งในขั้นตอนสำคัญในการสร้างและปรับปรุงโมเดล ปัญญาประดิษฐ์

2.15 การประเมินประสิทธิภาพของแบบจำลอง (Evaluate Model)

การประเมินประสิทธิภาพของแบบจำลอง (Evaluate Model) [33] หมายถึงกระบวนการที่ใช้ในการประเมินความสามารถของโมเดล (Model) ในการทำนายหรือคาดการณ์ผลลัพธ์ โดยการประเมินนี้จะช่วยให้ผู้ใช้งานสามารถตรวจสอบความถูกต้องของโมเดลได้ว่ามีความแม่นยำและเหมาะสมกับการใช้งานหรือไม่ โดยการประเมินนั้นจะใช้เครื่องมือหรือเทคนิคต่างๆ เช่น Confusion Matrix, Accuracy, Precision, Recall, F1-score, ROC Curve เป็นต้น ซึ่งจะช่วยให้ผู้ใช้งานสามารถปรับปรุงโมเดลให้ดียิ่งขึ้นได้ และสามารถนำไปใช้งานได้อย่างมีประสิทธิภาพมากยิ่งขึ้น โดยมีขั้นตอนการประเมินประสิทธิภาพของแบบจำลองดังต่อไปนี้

- เลือกชุดข้อมูลทดสอบ (Test Data) โดยขั้นตอนแรกคุณจะต้องเลือกชุดข้อมูลทดสอบที่ไม่ได้ใช้ในการฝึกอบรมแบบจำลอง ชุดข้อมูลทดสอบควรมีความคล้ายคลึงกับข้อมูลจริงที่คุณต้องการใช้แบบจำลองในสถานการณ์จริง

- ใช้แบบจำลองในการทำนาย โดยนำแบบจำลองที่คุณสร้างมาใช้กับชุดข้อมูลทดสอบ เพื่อทำนายผลลัพธ์ของข้อมูลในชุดทดสอบ

- คำนวณตัวชี้วัด โดยหลังจากที่คุณใช้แบบจำลองในการทำนายผลลัพธ์ของชุดข้อมูลทดสอบแล้ว คุณสามารถคำนวณตัวชี้วัดประสิทธิภาพต่าง ๆ ได้ เช่น

ความแม่นยำ (Accuracy): วัดสัดส่วนของตัวอย่างที่ถูกจำแนกถูกต้องโดยแบบจำลอง

ความถูกต้อง (Precision): วัดความสามารถในการจำแนกคลาสบวกโดยแบบจำลอง

ความครอบคลุม (Recall): วัดความสามารถในการจำแนกคลาสบวกโดยแบบจำลอง

ค่า F1 (F1-Score): ความสมดุลระหว่างความถูกต้องและความครอบคลุม

- เมตริกซ์การจำแนก (Confusion Matrix) โดยแสดงจำนวนของ True Positives, False Positives, True Negatives, และ False Negatives

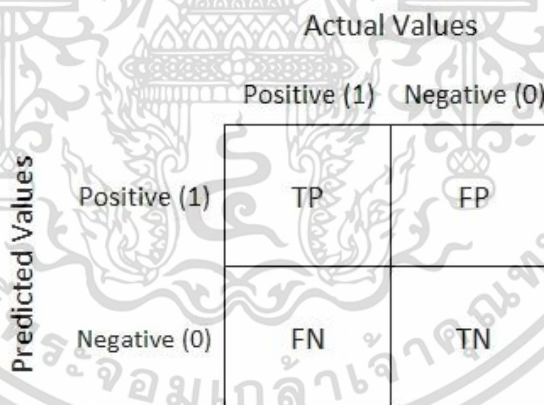
วิเคราะห์ผลลัพธ์ โดยหลังจากคำนวณตัวชี้วัดแล้วคุณควรวิเคราะห์ผลลัพธ์เพื่อเข้าใจว่าแบบจำลองมีประสิทธิภาพอย่างไร และอาจต้องพิจารณาการปรับปรุงแบบจำลองหรือชุดข้อมูลเพิ่มเติมหากจำเป็น

การทดสอบแบบจำลองอื่น ๆ โดยบางครั้งต้องทดสอบแบบจำลองที่แตกต่างกันหรือการเรียนรู้เชิงเครื่องที่ใช้อัลกอริทึมต่าง ๆ เพื่อเปรียบเทียบประสิทธิภาพและเลือกแบบจำลองที่ดีที่สุด

2.15.1 Confusion Matrix

Confusion Matrix [34] เป็นตารางที่ใช้ในการแสดงผลลัพธ์ของการทำนายหรือการจำแนกประเภท (Classification) โดยเฉพาะในงานที่มีคลาสหลายคลาส (Multi-Class Classification) หรืองานที่มีคลาสที่ไม่สมดุล (Imbalanced Classes) แม้ว่า Confusion Matrix จะใช้สำหรับงานที่มีคลาสสองคลาสบ่อยที่สุด คืองานจำแนกประเภท Binary Classification โดยมักแบ่งข้อมูลเป็น 4 ส่วนหลักได้แก่

- True Positive (TP) คือจำนวนข้อมูลที่โมเดลทำนายว่าเป็น Positive (บวก) และตรงกับค่าจริง
 - False Positive (FP) คือจำนวนข้อมูลที่โมเดลทำนายว่าเป็น Positive (บวก) แต่ไม่ตรงกับค่าจริง
 - False Negative (FN) คือจำนวนข้อมูลที่โมเดลทำนายว่าเป็น Negative (ลบ) แต่ไม่ตรงกับค่าจริง
 - True Negative (TN) คือจำนวนข้อมูลที่โมเดลทำนายว่าเป็น Negative (ลบ) และตรงกับค่าจริง
- ดังภาพที่ 2.29



A 2x2 confusion matrix for binary classification. The vertical axis is labeled 'Predicted Values' with categories 'Positive (1)' and 'Negative (0)'. The horizontal axis is labeled 'Actual Values' with categories 'Positive (1)' and 'Negative (0)'. The cells contain: TP (True Positive) at (1,1), FP (False Positive) at (1,2), FN (False Negative) at (2,1), and TN (True Negative) at (2,2).

Predicted Values	Actual Values	
	Positive (1)	Negative (0)
Positive (1)	TP	FP
Negative (0)	FN	TN

ภาพที่ 2.29 Confusion Matrix Binary Classification [34]

จาก Confusion Matrix สามารถคำนวณหาค่า Accuracy, Precision, Recall, F1-score และอื่นๆ ได้เพื่อประสิทธิภาพและประสิทธิผลของโมเดลในการจำแนกประเภทข้อมูล

2.15.2 Accuracy

Accuracy (ความแม่นยำ) [35] เป็นตัววัดประสิทธิภาพของโมเดลในการทำนายทั้งหมด ซึ่งคำนวณโดยหาสัดส่วนของจำนวนที่ทำนายถูกต้อง (True Positive และ True Negative) ต่อจำนวนทั้งหมดของข้อมูลที่ใช้ทดสอบ โดยคำนวณจากสมการที่ 1

$$\text{สมการคำนวณความแม่นยำคือ Accuracy} = \frac{TP + TN}{TP + FP + FN + TN} \quad (1)$$

ค่า Accuracy มีค่าระหว่าง 0 ถึง 1 โดยค่าที่มากที่สุดคือ 1 หมายถึงโมเดลทำนายได้ถูกต้องทั้งหมด และค่าที่น้อยที่สุดคือ 0 หมายถึงโมเดลทำนายผิดทั้งหมด

โดยตัวอย่างคือ Accuracy = $\frac{55 + 30}{55 + 5 + 30 + 10} = 0.85$ และความแม่นยำเป็นเปอร์เซ็นต์จะเป็น 85%

2.15.3 Precision

Precision (ความถูกต้อง) [35] เป็นตัวชี้วัดที่ใช้ในการประเมินประสิทธิภาพของแบบจำลองการจำแนกประเภท (Classification Model) โดยเฉพาะในงานจำแนกประเภท Binary Classification หรือ Multi-Class Classification ความถูกต้องวัดสัดส่วนของการจำแนกที่ถูกต้องเฉพาะคลาสที่น่าสนใจ (Positive Class) โดยคำนวณจากสมการที่ 2

$$\text{สมการคำนวณความถูกต้องคือ Precision} = \frac{TP}{TP + FP} \quad (2)$$

ความถูกต้องวัดความสามารถของแบบจำลองในการทำนายคลาสบวกโดยไม่สนใจว่าแบบจำลองมีการพลาดในการไม่ทำนายคลาสลบหรือคลาสอื่น ๆ ตัวชี้วัดนี้มีความสำคัญในงานที่ความผิดพลาดของการจำแนกคลาสบวกอาจมีผลกระทบที่ร้ายแรง เช่น งานการตรวจสอบโรคหรือการตรวจจับความผิดพลาดทางการเงิน ในกรณีเช่นนี้อาจต้องการความถูกต้องสูงเพื่อลดความเสี่ยงที่เกิดจากการทำนายคลาสบวกเมื่อไม่จริง

$$\text{โดยตัวอย่างคือ Precision} = \frac{30}{30 + 5} = 0.857$$

2.15.4 Recall

Recall (ความครอบคลุม) [35] เป็นตัวชี้วัดที่ใช้ในการประเมินประสิทธิภาพของแบบจำลองเพื่อตรวจจับค่า Positive โดยค่า Recall สามารถคำนวณได้จากสัดส่วนของจำนวน True Positive (TP) ต่อจำนวนที่จริงๆ เป็น Positive (TP + FN) โดยคำนวณโดยใช้สมการที่ 3

$$\text{สมการคำนวณความครอบคลุมคือ Recall} = \frac{TP}{TP + FN} \quad (3)$$

ค่า Recall มีค่าระหว่าง 0 ถึง 1 โดยค่าที่สูงที่สุดคือ 1 หมายถึงโมเดลสามารถตรวจจับ Positive ทั้งหมดที่เป็นไปได้ และค่าที่น้อยที่สุดคือ 0 หมายถึงโมเดลไม่สามารถตรวจจับ Positive ได้

$$\text{โดยตัวอย่างคือ Precision} = \frac{30}{30 + 10} = 0.75$$

2.15.5 F1 Score

F1 Score [35] เป็นค่าที่เอามาวัดความสมดุลระหว่าง Precision และ Recall ของโมเดล โดย F1 Score คำนวณจากสูตรเฉพาะที่คูณค่า Precision กับ Recall โดยคำนวณโดยใช้สมการที่ 4

$$\text{สมการคำนวณคือ F1 Score} = \frac{2 * (\text{Precision} * \text{Recall})}{\text{Precision} + \text{Recall}} \quad (4)$$

ค่า F1 Score จะอยู่ระหว่าง 0 ถึง 1 โดยค่าที่สูงที่สุดคือ 1 หมายถึงโมเดลมีความสมดุลระหว่าง Precision และ Recall ที่ดีที่สุด ค่า F1 Score จะมีค่าสูงเมื่อ Precision และ Recall มีค่าสูงพร้อมกัน และจะมีค่าต่ำเมื่อมีค่า Precision หรือ Recall ที่ต่ำลง

$$\text{โดยตัวอย่างคือ Precision} = \frac{2*(0.857+0.75)}{0.857 + 0.75} = 0.799$$

2.16 Operating Characteristics of Tests or Procedures

Operating Characteristics of Tests or Procedures เป็นคอนเซ็ปต์ทางการแพทย์และวิทยาการที่ใช้ในการประเมินและอธิบายประสิทธิภาพของการทดสอบทางการแพทย์หรือกระบวนการทางการแพทย์ เช่น การทดสอบการตรวจหาโรค การวินิจฉัยการรักษา หรือกระบวนการการผ่าตัด คอนเซ็ปต์นี้มักถูกใช้เพื่อให้ความเข้าใจเกี่ยวกับประสิทธิภาพของการทดสอบหรือกระบวนการนั้น ๆ โดยใช้ตัวชี้วัดหลาย ๆ อย่าง เพื่อให้ได้ภาพรวมที่ครอบคลุม

คุณสมบัติหรือลักษณะการทำงานที่สำคัญสำหรับการทดสอบหรือกระบวนการเหล่านี้รวมถึงความน่าเชื่อถือ (Reliability) คือความน่าเชื่อถือในผลลัพธ์ของการทดสอบหรือกระบวนการ, ความแม่นยำ (Accuracy) คือความเป็นไปได้ในการให้ผลลัพธ์ที่ถูกต้องและเที่ยงตรง, ความแม่นยำในการตรวจจับ (Sensitivity) คือความสามารถในการตรวจจับสิ่งที่สนใจ และความแม่นยำในการตรวจจับสิ่งที่ไม่สนใจ (Specificity) คือความสามารถในการตรวจจับสิ่งที่ไม่สนใจโดยไม่มีการตรวจจับผิดพลาด การประเมินและเข้าใจคุณสมบัติหรือลักษณะการทำงานเหล่านี้จะช่วยให้เลือกใช้หรือปรับปรุงการทดสอบหรือกระบวนการให้เหมาะสมกับการใช้งานและความต้องการได้

2.16.1 Positive and Negative Class

Positive Class และ Negative Class [36] เป็นคำศัพท์ที่ใช้ในงานวิทยาการข้อมูล (Data Science) และการเรียนรู้เชิงลึก (Deep Learning) เพื่ออธิบายคลาสหรือกลุ่มข้อมูลที่สนใจในการจำแนกหรือตรวจจับ

Positive Class คือคลาสหรือกลุ่มข้อมูลที่สนใจหรือต้องการจำแนกว่าเป็นผลลัพธ์บวก (Positive Outcome) หรือเป็นคลาสที่สนใจในการตรวจจับเหตุการณ์หรือปัญหาที่ต้องการค้นหา ในการวิเคราะห์ข้อมูลหรือการสร้างโมเดล

Negative Class คือคลาสหรือกลุ่มข้อมูลที่ไม่ใช่ Positive Class หรือไม่เป็นผลลัพธ์บวกที่สนใจหรือต้องการจำแนก ในบางกรณี Negative Class อาจเป็นคลาสที่สนใจว่าเป็นผลลัพธ์ลบ (Negative Outcome) หรือเป็นคลาสที่ไม่สนใจในการตรวจจับหรือตรวจสอบ

2.16.2 Sensitivity

Sensitivity (ความไวต่อการตรวจจับ) [37] เป็นค่าที่วัดความสามารถของการตรวจจับสิ่งที่สนใจโดยไม่พลาด (True Positive) หรือความสามารถในการตรวจจับผลบวกที่เป็นจริง โดยคำนวณโดยสัดส่วนของจำนวนผลบวกที่ถูกต้องต่อจำนวนผลบวกทั้งหมดที่เป็นจริง โดยคำนวณโดยใช้สมการที่ 1

$$\text{สมการคำนวณคือ Sensitivity} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \quad (1)$$

ค่า Sensitivity หรือ True Positive Rate (TPR) บ่งบอกถึงความสามารถของแบบจำลองในการตรวจจับคลาสบวก (Positive Class) อย่างมีประสิทธิภาพ หากค่า Sensitivity สูงแสดงว่าแบบจำลองมีความสามารถในการตรวจจับคลาสบวกได้ดีและมีความน่าเชื่อถือ หากค่า Sensitivity ต่ำอาจเกิดความพลาดในการตรวจจับสิ่งที่สนใจได้มากขึ้น

สรุป Sensitivity คือ ความน่าจะเป็นหรือโอกาสที่การตรวจได้ผลบวกเมื่อผู้ป่วยเป็นโรค True Positive Rate การตรวจที่มีความไวสูงจะเป็นเครื่องมือที่เหมาะสมสำหรับคัดกรองโรค Screening Test

2.16.3 Specificity

Specificity (ความไวต่อการตรวจไม่พบ) [37] คือค่าที่ใช้ในการวัดความแม่นยำของระบบหรือโมเดลในการตรวจจับคลาสลบ (Negative Class) หรือคลาสที่ไม่สนใจ ซึ่งสามารถคำนวณได้จากจำนวนของตัวอย่างที่ถูกตรวจจับว่าเป็นคลาสลบและตรวจจับผิด (False Positives) และจำนวนของตัวอย่างที่ถูกตรวจจับว่าเป็นคลาสบวก (Positive Class) และตรวจจับถูกต้อง (True Negatives) โดยคำนวณโดยใช้สมการที่ 2

$$\text{สมการคำนวณคือ Specificity} = \frac{\text{True Negatives}}{\text{True Negatives} + \text{False Positives}} \quad (2)$$

ค่า Specificity หรือ True Negative Rate (TNR) บ่งบอกถึงความสามารถของแบบจำลอง ในการระบุคลาสลบ (Negative Class) อย่างมีประสิทธิภาพ หรือระบุว่าสถานะที่ไม่สนใจไม่มีอยู่ (เช่น การระบุว่าคนไม่มีโรคมะเร็ง) หากค่า Specificity สูงแปลว่าแบบจำลองมีความสามารถในการ ระบุคลาสลบ (Negative Class) ได้ดีและมีความน่าเชื่อถือ ในขณะที่ค่า Specificity ต่ำแสดงว่าระบบ หรือโมเดลไม่สามารถตรวจจับคลาสบวก (Positive Class) ได้

สรุป Specificity คือ ความน่าจะเป็นหรือโอกาสที่การตรวจได้ผลลบเมื่อผู้ป่วยไม่เป็นโรค (True Negative Rate) การตรวจที่มีความจำเพาะสูงจะเป็นเครื่องมือที่เหมาะสมสำหรับวินิจฉัยโรค (Diagnostic Test)

2.17 การแปลผลค่า Sensitivity and Specificity

การแปลผลค่า Sensitivity และ Specificity อาจมีหลายวิธีขึ้นอยู่กับบริบทและการใช้งานที่ เกี่ยวข้อง ดังนี้

2.17.1 Sensitivity

ค่า Sensitivity สูงแสดงถึงความไวในการตรวจจับผลบวกที่เป็นจริง หรือความสามารถใน การตรวจจับสิ่งที่สนใจโดยไม่พลาด

ค่า Sensitivity 100% หมายถึงว่าการตรวจจับสิ่งที่สนใจทุกกรณีที่เป็นจริง

ค่า Sensitivity ต่ำสามารถแสดงถึงความไม่แม่นยำในการตรวจจับสิ่งที่สนใจหรือมีความ ผิดพลาดในการตรวจจับสิ่งที่สนใจ

ในงานการแพทย์ Sensitivity แปลว่าความสามารถของการทดสอบในการตรวจจับโรคหรือ สภาวะที่สนใจ (เช่น การตรวจหาโรคมะเร็ง) หาก Sensitivity สูง แปลว่าการทดสอบมีความสามารถ ในการตรวจจับโรคได้ดีและมีความน่าเชื่อถือ

2.17.2 Specificity

ค่า Specificity สูงแสดงถึงความแม่นยำในการตรวจจับผลลบที่ไม่สนใจ หรือความไม่มีความ ผิดพลาดในการตรวจจับสิ่งที่ไม่สนใจ

ค่า Specificity 100% หมายถึงว่าการตรวจจับสิ่งที่ไม่สนใจทุกกรณีที่เป็นจริง

ค่า Specificity ต่ำสามารถแสดงถึงความไม่แม่นยำในการตรวจจับสิ่งที่ไม่สนใจ หรือมีความ ผิดพลาดในการตรวจจับสิ่งที่ไม่สนใจ

ในงานการแพทย์ Specificity แปลว่าความสามารถของการทดสอบในการระบุภาวะที่ไม่สนใจไม่มีอยู่ (เช่น การระบุว่าคุณไม่มีโรคมะเร็ง) หาก Specificity สูง แปลว่าการทดสอบมีความสามารถในการระบุภาวะนี้ได้ดีและมีความน่าเชื่อถือ

สรุปได้ว่า Sensitivity มีประสิทธิภาพในการตรวจจับคลาสิก ในขณะที่ Specificity มีประสิทธิภาพในการระบุคลาสิก การแปลผลค่า Sensitivity และ Specificity ควรพิจารณาในบริบทของงานและการตัดสินใจที่เกี่ยวข้องกับการใช้งานแบบจำลองหรือการทดสอบที่เป็นไปได้



บทที่ 3

การดำเนินงาน

การดำเนินการวิจัยจะแบ่งออกเป็น 2 ส่วน ส่วนแรกคือการพัฒนาเว็บไซต์เพื่อเก็บข้อมูลภาพและเสียงจากอาสาสมัคร ส่วนที่สองคือการพัฒนาแบบจำลองการเรียนรู้เชิงลึกโดยใช้เสียงของมนุษย์เพื่อบ่งบอกและระบุภาวะซึมเศร้า

3.1 จริยธรรมการวิจัยในมนุษย์และการเก็บข้อมูลอาสาสมัคร

ทางผู้วิจัยได้ดำเนินการขอจริยธรรมจริยธรรมการวิจัยในมนุษย์กับทั้งสองสถาบัน ได้แก่

1. สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ผ่านการประเมิน
 2. คณะแพทยศาสตร์โรงพยาบาลรามาธิบดี มหาวิทยาลัยมหิดล ผ่านการประเมิน
- ตัวอย่างเอกสารการรับรอง

Human Research Ethics Committee, Faculty of Medicine Ramathobdi Hospital, Mahidol University
270 Rama 6 Rd. Phrayatal Ratchathewi Bangkok 10400 Tel.(660)2012175, 2011544, 2010388
Website: <https://med.mahidol.ac.th/research/ethics>
Email: hec.mahidol@gmail.com

COA. MURA2022/242

Title of Project (English)	Depression screening system using data-driven audio and video by deep learning model
Title of Project (Thai)	ระบบคัดกรองภาวะซึมเศร้าโดยใช้เสียงจากภาพวิดีโอที่เรียนรู้เชิงลึกด้วยรูปแบบเสียง
Type of Review	Expedited
Principal Investigator	Treesukon Treebupachatsakul
Official Address	Biomedical Engineering, School of Engineering, King Mongkut's Institute of Technology Ladkrabang
Co-Investigator(s)	1. Suvit Poommittigul 2. Kornsan Kiatnugrit 3. Phansorn Homsiang
Approval includes	1. Submission form protocol version 2 date 24/1/2022 2. Information sheet and consent form version 1 date 5/4/2022 3. Assent form aged 13 years-less than 18 years version 4 date 7/4/2022 4. Questionnaire 5. Data sharing agreement 6. Certificate in ethics training

Institutional Review Boards in Mahidol University are in full compliance with International Guidelines for Human Research Protection such as Declaration of Helsinki, The Belmont Report, CIOMS Guidelines and the International Conference on Harmonization in Good Clinical Practice (ICH-GCP)

Date of Approval April 24, 2022
Date of Expiration April 23, 2023

Signature of Chair.....
(Asst. Prof. Chusak Okascharoen, M.D., Ph.D.)

ภาพที่ 3.1 เอกสารการรับรองผ่านการประเมินจริยธรรมในมนุษย์ คณะแพทยศาสตร์โรงพยาบาลรามาธิบดี มหาวิทยาลัยมหิดล

3.2 การเก็บข้อมูล

3.2.1 กลุ่มตัวอย่างของงานวิจัย

- กลุ่มที่ 1 ผู้ป่วยที่ได้รับการวินิจฉัยเป็นภาวะซีมีเศร้า ที่มารับการรักษาในแผนกตรวจผู้ป่วยนอกภาควิชาจิตเวชศาสตร์ คณะแพทยศาสตร์ โรงพยาบาลรามาธิบดี มหาวิทยาลัยมหิดล คุณสมบัติต้องอายุตั้งแต่ 13 ปีขึ้นไปและสามารถอ่านและตอบแบบสอบถามได้ด้วยตนเอง และทางผู้วิจัยได้เก็บข้อมูลอาสาสมัครด้วยวิธีเก็บในห้องตรวจของภาควิชาจิตเวชศาสตร์ คณะแพทยศาสตร์ โรงพยาบาลรามาธิบดี



ภาพที่ 3.2 ตัวอย่างอาสาสมัครที่เก็บข้อมูลในห้องตรวจของภาควิชาจิตเวชศาสตร์ คณะแพทยศาสตร์ โรงพยาบาลรามาธิบดี

- กลุ่มที่ 2 บุคคลทั่วไป คุณสมบัติต้องอายุตั้งแต่ 13 ปีขึ้นไป สามารถอ่านและตอบแบบสอบถามได้ด้วยตนเองและสามารถบันทึกเสียงและวิดีโอได้ด้วยตนเอง ผ่านเว็บไซต์ที่ผู้วิจัยพัฒนาขึ้น

- จำนวนอาสาสมัครได้แก่ กลุ่มที่ 1 จำนวน 50 ท่าน กลุ่มที่ 2 จำนวน 50 ท่าน รวมเป็น 100 ท่าน

3.2.2 การเก็บรักษาข้อมูล

- ความปลอดภัยของข้อมูล

การเก็บข้อมูล Audio และ Video ไว้ใน Server สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง โดย Server นั้นจะมีเพียงผู้วิจัยเท่านั้นที่มีรหัสผ่านที่จะสามารถเข้าไปใช้งาน

Server ได้และยังมีการป้องกันว่าจะสามารถเข้าถึง Server ได้ต้องใช้งาน Internet ของสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังเท่านั้น วิธีเข้าถึง Server คือ 1.ต้องใช้งาน Internet ของสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง 2.ต้องมี IP-ADDRESS ของ Server 3. การจะเข้า Server ต้องได้รับสิทธิ์จากทางสำนักงานคอมพิวเตอร์ของสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง และ 4.การเข้าถึง Server ต้องมีรหัสผ่านเท่านั้น

- การเข้าถึงข้อมูล

มีเพียงนาย พนมกร หอมเสียง (ผู้วิจัย) ที่สามารถเข้าถึงข้อมูลได้

- ระยะเวลาการเก็บข้อมูล

1 ปี หรือเมื่องานวิจัยเสร็จสิ้น

- การทำลายข้อมูล

ข้อมูลจะถูกทำลายหลังจาก 1 ปี หรืองานวิจัยเสร็จสิ้นด้วยการลบข้อมูลทั้งหมดออกจาก Server และผู้วิจัยจะปิด Server นั้นไม่ให้อ่านการใช้งานได้ จากนั้นทางผู้วิจัยจะทำเรื่องไปยังสำนักงานคอมพิวเตอร์ของสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ให้ลบ Server นี้ ออกจากระบบ

3.2.3 การจัดการข้อมูล

- บันทึกข้อมูลเป็นไฟล์ภาพและเสียง จากการบันทึกและอัปโหลดวิดีโอลง database และประมวลผลแบบอัตโนมัติในการให้คะแนนของผู้ทดสอบ และแสดงผลการประเมินผลตามเกณฑ์กับผู้ทดสอบ โดยเก็บข้อมูลภาพและเสียงของอาสาสมัครที่อ่าน Consent Form จำนวน 1 ไฟล์และ ภาพและเสียงของอาสาสมัครที่ตอบคำถาม 9 ข้อของ PHQ-9 จำนวน 9 ไฟล์รวมเป็น 10 ไฟล์ต่อ 1 อาสาสมัคร ความยาวของแต่ละไฟล์โดยเฉลี่ยไม่เกิน 1 นาทีโดยรวมความยาวของวิดีโอคือ 10 นาทีต่ออาสาสมัคร 1 ท่าน การเก็บรักษาคือเก็บไฟล์วิดีโอไว้บน Server ของทางสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังโดยทำลายทิ้งหลังเสร็จสิ้นโครงการ

- การเก็บข้อมูลจะเก็บโดยตัวของผู้วิจัยเอง และจัดเก็บใน Hard disk ของผู้วิจัยเองโดยไม่เก็บไว้บน Cloud เพื่อไม่ให้ข้อมูลรั่วไหล จะทำเป็นรหัสล็อกอินให้อาสาสมัครเพื่อ login เข้าระบบโดยไม่ระบุตัวตน

3.3 แบบทดสอบภาวะซึมเศร้า Patient Health Questionnaire – 9 Thai Version (PHQ-9)

เป็นแบบสอบถามเกี่ยวกับอาการโรคซึมเศร้าในช่วง 2 สัปดาห์ที่ผ่านมา มีจำนวน 9 ข้อ แต่ละข้อมี 4 ตัวเลือกได้แก่ ไม่เลย (0 คะแนน) มีบางวันไม่บ่อย (1 คะแนน) มีค่อนข้างบ่อย (2 คะแนน) มีเกือบทุกวัน (3 คะแนน) โดยการแปลผลคะแนนมีเกณฑ์ดังต่อไปนี้

คะแนน 0 – 4 หมายถึง อยู่ในระดับไม่มีอาการซึมเศร้า

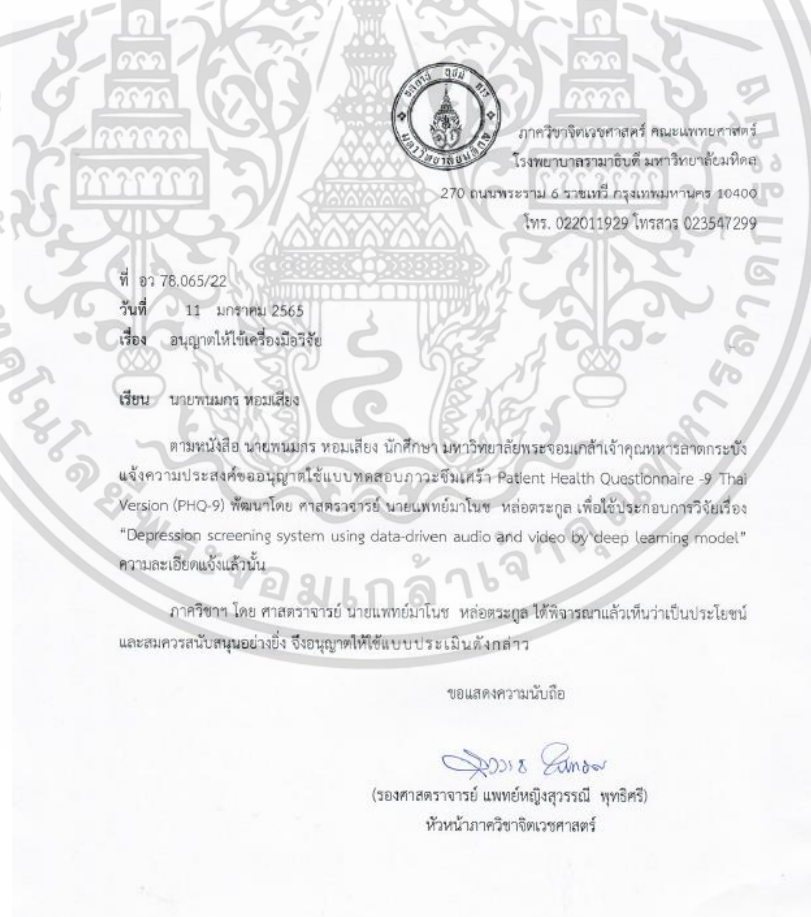
คะแนน 5 – 8 หมายถึง มีภาวะซึมเศร้าอยู่ในระดับน้อย

คะแนน 9 – 14 หมายถึง มีภาวะซึมเศร้าอยู่ในระดับปานกลาง

คะแนน 15 – 19 หมายถึง มีภาวะซึมเศร้าอยู่ในระดับมาก

คะแนน 20 – 27 หมายถึง มีภาวะซึมเศร้าอยู่ในระดับรุนแรง

และทางผู้วิจัยได้ผ่านการรับรองให้สามารถใช้งานแบบสอบถามภาวะซึมเศร้า PHQ-9 เวอร์ชันภาษาไทยได้



ภาพที่ 3.3 เอกสารอนุญาตให้ใช้เครื่องมือ แบบทดสอบภาวะซึมเศร้า Patient Health Questionnaire – 9 Thai Version (PHQ-9)

3.4 การพัฒนาโปรแกรมเพื่อใช้เป็นแบบทดสอบ PHQ-9 และเก็บข้อมูลของอาสาสมัคร

3.4.1 เครื่องมือที่ใช้พัฒนาเว็บไซต์

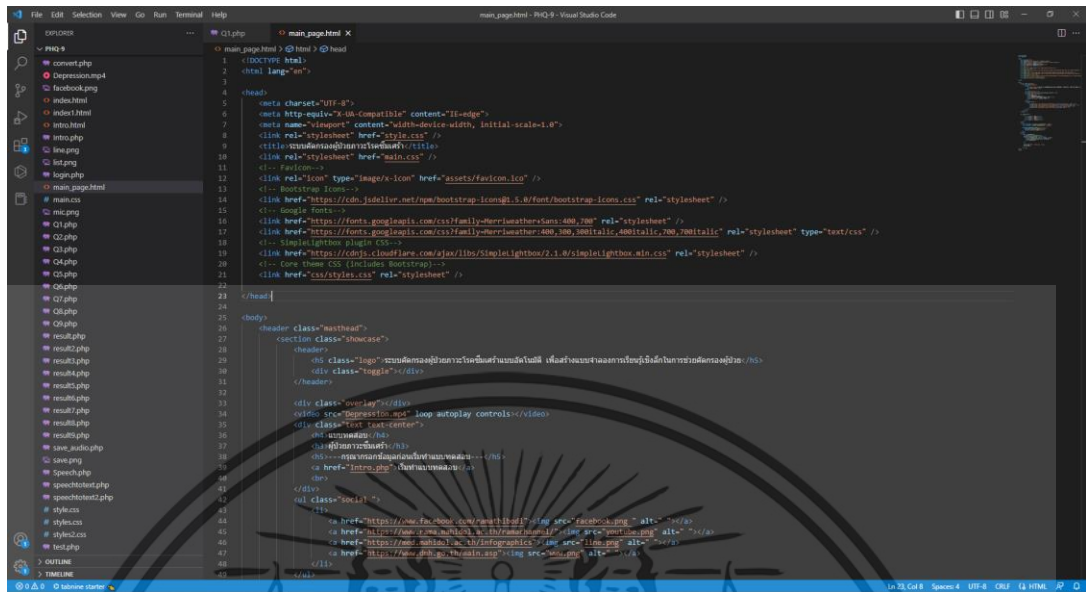
ผู้วิจัยได้ใช้เครื่องมือในการพัฒนาเว็บไซต์โดยแบบออกเป็น 4 ประเภทคือ 1. Front End 2. Back End และได้เก็บข้อมูลคำตอบของอาสาสมัครอยู่ใน 3. Database ของผู้วิจัย โดยทั้งหมดทำงานบน 4. Windows Server ของทางสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง มีรายละเอียด ดังนี้

3.4.1.1 Front-End

Front-End (หรือ Frontend) [38] คือส่วนหนึ่งของแอปพลิเคชันหรือเว็บไซต์ที่เป็นส่วนที่ผู้ใช้เห็นและทำงานกับตรงกับส่วนหน้าหลัก ซึ่งใช้สำหรับการแสดงข้อมูลและอินเตอร์แอคทีฟ (Interaction) ระหว่างผู้ใช้และแอปพลิเคชันหรือเว็บไซต์นั้น นี่เป็นส่วนที่ผู้ใช้สามารถมองเห็นและปฏิสัมพันธ์กับข้อมูลหรือบริการที่เว็บไซต์หรือแอปพลิเคชันนั้นให้บริการ

Front-End ประกอบด้วย

- HTML (Hypertext Markup Language) [38] ใช้ในการกำหนดโครงสร้างและเนื้อหาของหน้าเว็บ เป็นภาษาที่ใช้ในการสร้างโครงสร้างของหน้าเว็บ
- CSS (Cascading Style Sheets) [38] ใช้ในการกำหนดรูปแบบและสไตล์สำหรับเนื้อหาที่กำหนดด้วย HTML เช่น การกำหนดสีพื้นหลัง, ขนาดและรูปร่างของข้อความ, และการจัดวาง.
- JavaScript [38] เป็นภาษาโปรแกรมที่ใช้สำหรับเพิ่มความแอคทีฟ (Interactivity) ในหน้าเว็บ โดย JavaScript สามารถใช้ในการจัดการกับข้อมูลในเว็บ, การตรวจสอบปุ่มกด, การสร้างเมนูแบบแสดงผลตั้งแต่การสร้างกราฟและการประมวลผลข้อมูลเมื่อผู้ใช้กระทำการบนหน้าเว็บ
- Framework [38] และ Library มีหลาย Framework และ Library ที่ช่วยในการพัฒนา Front-End อย่างรวดเร็วและมีประสิทธิภาพ เช่น React, Node.js, Angular, Vue.js, Bootstrap, และอื่น ๆ ซึ่งช่วยในการสร้างส่วนหน้าของเว็บไซต์หรือแอปพลิเคชันได้อย่างมีประสิทธิภาพ
- โดยผู้วิจัยเลือกใช้ HTML, CSS, JavaScript เป็น Front-End ของเว็บไซต์ ดังภาพที่ 3.4



ภาพที่ 3.4 ตัวอย่าง Source code ของเว็บไซต์ฝั่ง Front-End

3.4.1.2 Back-End

Back-End (หรือ Backend) [38] คือส่วนหนึ่งของระบบแอปพลิเคชันหรือเว็บไซต์ที่ทำงานอยู่ที่ฝั่งเซิร์ฟเวอร์ ซึ่งไม่สามารถมองเห็นหรือปฏิสัมพันธ์ได้โดยตรงกับผู้ใช้ แต่มีหน้าที่สำคัญในการดำเนินการกับข้อมูล, การประมวลผล, และการจัดการกับการสื่อสารระหว่างคลิเอนต์ (Clients) และเซิร์ฟเวอร์ (Server)

Back-End ประกอบด้วย

เซิร์ฟเวอร์ (Server) เป็นคอมพิวเตอร์หรือเครื่องคอมพิวเตอร์ที่ทำหน้าที่ให้บริการแก่ผู้ใช้ซึ่งรับคำขอ (Requests) จากคลิเอนต์ผ่านทางเครือข่าย (เช่น HTTP Requests) และจัดการประมวลผลคำขอนั้นตามที่ได้รับกำหนด

ฐานข้อมูล (Database) ส่วนนี้ใช้ในการเก็บข้อมูลและข้อมูลที่สำคัญสำหรับแอปพลิเคชันหรือเว็บไซต์ เช่น ข้อมูลของผู้ใช้, ข้อมูลสินค้า, ข้อมูลการทำธุรกรรม, และอื่น ๆ การจัดการข้อมูลและการเรียกข้อมูลจากฐานข้อมูลเป็นหน้าที่สำคัญของ Back-End

การประมวลผล (Processing) โดย Back-End รับข้อมูลจากคลิเอนต์ (Clients), ดำเนินการประมวลผลตามคำขอ, และส่งผลลัพธ์กลับให้กับคลิเอนต์ การประมวลผลอาจเป็นการคำนวณ, การตรวจสอบสิทธิ์การเข้าถึงข้อมูล, การสร้างรายงาน, หรือการสร้างข้อมูลในรูปแบบที่สามารถนำไปแสดงให้กับผู้ใช้

การจัดการระบบและรักษาความปลอดภัย (System Management and Security) โดย Back-End รับผิดชอบในการจัดการระบบ, การรักษาความปลอดภัยของข้อมูลและการปกป้องตัวเซิร์ฟเวอร์จากการโจมตี

APIs (Application Programming Interfaces) โดย Back-End อาจมี APIs ที่ช่วยให้แอปพลิเคชันหรือบริการอื่น ๆ สามารถเชื่อมต่อและใช้งานข้อมูลหรือบริการที่ Back-End ให้บริการ

Back-End มีบทบาทสำคัญในการจัดการและประมวลผลข้อมูลเพื่อให้แอปพลิเคชันหรือเว็บไซต์ทำงานอย่างถูกต้องและมีประสิทธิภาพ ส่วน Front-End และ Back-End ทำงานร่วมกันเพื่อสร้างประสบการณ์การใช้งานที่ดีให้กับผู้ใช้

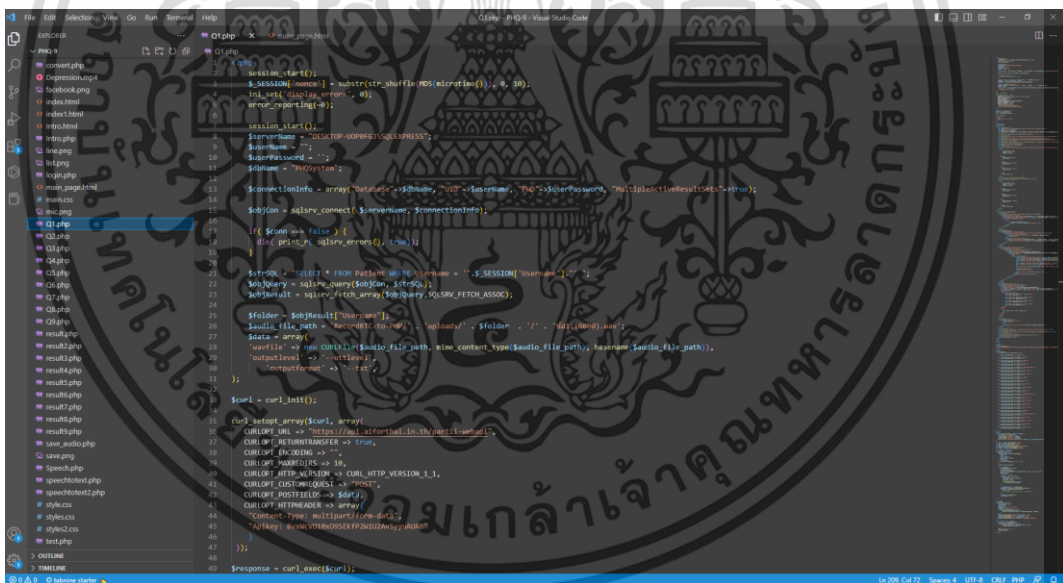
โดยผู้วิจัยเลือกใช้ PHP, NODE JS เป็น Front-End ของเว็บไซต์โดยมีรายละเอียดดังนี้

- PHP [39] หรือ PHP Hypertext Preprocessor เป็นภาษาโปรแกรมมิ่งที่ใช้สำหรับพัฒนาเว็บไซต์และแอปพลิเคชันเว็บ โดย PHP ทำงานร่วมกับเซิร์ฟเวอร์เพื่อสร้างและประมวลผลเนื้อหาที่ถูกส่งไปยังเว็บเบราว์เซอร์ของผู้ใช้ ภาษา PHP เป็นภาษาที่มีความยืดหยุ่นสูง สามารถผสานกับ HTML และ CSS ได้ง่าย และสามารถเชื่อมต่อกับฐานข้อมูลได้ง่ายด้วย ซึ่งทำให้ PHP เป็นภาษาที่ได้รับความนิยมในการพัฒนาเว็บไซต์และแอปพลิเคชันเว็บมากที่สุดในโลก โดย PHP มักใช้ในการพัฒนาเว็บไซต์และแอปพลิเคชันที่มีความต้องการในการประมวลผลข้อมูลที่เป็นการส่งผ่านเว็บ โดยเฉพาะเว็บไซต์ที่มีเนื้อหาและการโต้ตอบกับผู้ใช้ ตัวอย่างที่นิยมของแอปพลิเคชัน PHP รวมถึง WordPress, Joomla, Drupal, Magento, และ Laravel ซึ่งเป็นระบบจัดการเว็บครบวงจรและเฟรมเวิร์กที่ใช้ PHP ในการพัฒนา

- Node.js [40] เป็นแพลตฟอร์มการเขียนโปรแกรมที่ใช้ภาษา JavaScript และรันบนเครื่องเซิร์ฟเวอร์ โดยสามารถใช้งานได้ทั้งในระบบปฏิบัติการ Windows, macOS, และ Linux โดย Node.js ใช้ JavaScript เพื่อพัฒนาแอปพลิเคชันที่ทำงานบนเว็บและเซิร์ฟเวอร์ เช่น การสร้างเว็บแอปพลิเคชัน, หรือเขียนสคริปต์ที่ทำงานบนเครื่องเซิร์ฟเวอร์ โดย Node.js มีความสามารถในการทำงานแบบเหมือนภาษาโปรแกรมเชิงอ็อบเจกต์ และมีความสามารถในการทำงานแบบไม่ระบุเหตุการณ์ ทำให้สามารถจัดการกับการทำงานหลายๆ อย่างพร้อมกันได้ และยังสนับสนุนการทำงานแบบไม่ระบุเหตุการณ์ด้วยการใช้งาน Non-blocking I/O ที่ช่วยให้การทำงานไม่ถูกบล็อก และทำให้การรอคอยของโปรแกรมมีประสิทธิภาพมากขึ้น โดย Node.js มีความสำคัญในการพัฒนาแอปพลิเคชันเว็บที่ต้องการประสิทธิภาพและการปรับตัวได้ง่าย ๆ และมักถูกนำมาใช้ในการสร้างแอปพลิเคชันที่

เกี่ยวข้องกับการสื่อสารแบบ Real-Time และการประมวลผลข้อมูลแบบสด ๆ ที่ต้องใช้การทำงานแบบ Non-Blocking

- API [41] ย่อมาจาก Application Programming Interface เป็นชุดของคำสั่ง และรูปแบบการเข้าถึงและใช้งานฟังก์ชันหรือบริการที่ให้โดยระบบหรือแอปพลิเคชันอื่น ๆ ซึ่งช่วยให้สามารถแลกเปลี่ยนข้อมูลและทำงานร่วมกับระบบหรือแอปพลิเคชันนั้น ๆ โดย API ทำหน้าที่เป็นตัวกลางในการสื่อสารระหว่างระบบหรือแอปพลิเคชันที่แตกต่างกัน โดยอนุญาตให้แอปพลิเคชันหรือโปรแกรมอื่น ๆ เรียกใช้งานฟังก์ชันหรือบริการที่ระบบหรือแอปพลิเคชันนั้น ๆ มีให้ใช้ ซึ่งสามารถเป็นการรับหรือส่งข้อมูล การทำงานกับฐานข้อมูล การประมวลผลข้อมูล หรือการเข้าถึงแหล่งข้อมูลอื่น ๆ ตามที่ระบบหรือแอปพลิเคชันนั้น ๆ ได้กำหนดไว้ และ API มีรูปแบบการเรียกใช้งานที่กำหนดไว้ ซึ่งรวมถึงการระบุพารามิเตอร์ที่ต้องส่งเข้าไป และรูปแบบข้อมูลที่ต้องส่งไปหรือรับกลับจากระบบหรือแอปพลิเคชันนั้น ๆ นอกจากนี้ API ยังสามารถกำหนดการรับส่งข้อมูลในรูปแบบต่าง ๆ เช่น JSON, XML, หรือไฟล์ภาพ ดังภาพที่ 3.5



ภาพที่ 3.5 ตัวอย่าง Source code ของเว็บไซต์ฝั่ง Back-End

3.4.1.3 Server

ทางผู้วิจัยเลือกใช้ SQL Server เพื่อเก็บ 2 ข้อมูลสำคัญ ๆ ของอาสาสมัครได้แก่

1. Username ที่ใช้สำหรับ Login
2. คำตอบของอาสาสมัครที่ทำแบบทดสอบภาวะซึมเศร้า PHQ-9 แล้ว

SQL Server - Structured Query Language (SQL) เป็นระบบจัดการฐานข้อมูลที่พัฒนาโดย Microsoft ซึ่งมีความสามารถในการจัดการและบริหารจัดการฐานข้อมูลที่มีขนาดใหญ่และซับซ้อน โดย SQL Server ใช้ภาษา SQL (Structured Query Language) เป็นภาษาในการสร้างแก้ไข และค้นหาข้อมูลในฐานข้อมูล นอกจากนี้ SQL Server ยังมีฟีเจอร์ที่สำคัญอื่น ๆ เช่น การสร้างเว็บแอปพลิเคชัน การจัดการความปลอดภัยของข้อมูล การสำรองข้อมูล การวิเคราะห์ข้อมูล และอื่น ๆ ที่เกี่ยวข้องกับการจัดการฐานข้อมูล

Microsoft SQL Server - ระบบการจัดการฐานข้อมูลเชิงสัมพันธ์ ที่พัฒนาโดย Microsoft (ไมโครซอฟท์) เป็นซอฟต์แวร์ที่มีหน้าที่หลักในการจัดการฐานข้อมูล Server เป็นผลิตภัณฑ์ที่มีหน้าที่หลักในการจัดเก็บและเรียกข้อมูลตามคำขอของแอปพลิเคชัน-ซอฟต์แวร์อื่น ๆ ซึ่งอาจทำงานได้บนคอมพิวเตอร์เครื่องเดียวกันหรือบนคอมพิวเตอร์เครื่องอื่นในเครือข่าย ดังภาพที่ 3.6

result_id	name	one	two	three	four	five	six	seven	eight	nine
12	PHQ001	1	0	1	1	1	1	2	1	0
13	PHQ002	1	1	1	0	1	0	1	1	0
14	PHQ150	2	2	3	3	1	2	2	1	1
15	PHQ151	1	1	1	1	2	2	0	1	0
16	PHQ007	1	1	2	1	1	2	1	1	1
17	PHQ205	3	3	3	1	0	2	0	0	1
18	PHQ006	0	0	0	0	0	0	0	0	0
19	PHQ152	3	3	3	3	3	2	1	1	3
20	PHQ153	3	2	3	3	3	2	3	2	2
1012	PHQ003	0	1	0	0	0	1	0	0	0
1013	PHQ004	1	1	0	0	0	0	2	0	1
1014	PHQ011	1	0	2	1	0	1	0	0	0
1015	PHQ013	1	0	2	3	2	0	2	1	0
1016	PHQ009	0	0	1	0	0	0	0	0	0
1017	PHQ014	1	0	1	0	0	0	0	0	0
1018	PHQ019	0	1	2	2	1	1	0	0	1
1019	PHQ154	1	1	3	2	3	3	3	2	1
1020	PHQ030	1	2	0	1	3	2	3	1	0
1023	PHQ155	1	1	0	0	0	2	2	1	0
1025	PHQ021	1	0	0	0	1	0	0	0	0
1026	PHQ156	1	1	2	3	1	2	1	0	0
1028	PHQ157	1	1	2	2	2	0	1	0	1
1029	PHQ158	1	1	1	1	2	2	0	0	1
1030	PHQ159	1	1	2	1	1	1	2	2	1
1031	PHQ160	3	2	3	3	1	2	2	1	1
1032	PHQ161	2	1	3	3	2	1	2	1	1
1033	PHQ025	1	1	1	1	1	1	0	0	0
1034	PHQ162	1	1	2	2	2	3	1	0	2
1035	PHQ163	2	3	3	0	2	2	2	2	3
1036	PHQ164	2	3	2	3	1	3	2	3	3
1037	PHQ165	1	2	2	2	2	2	1	2	2
1038	PHQ167	1	1	2	2	3	2	1	0	1
1039	PHQ022	1	1	1	1	0	0	1	1	0
1040	PHQ166	1	2	2	1	0	1	0	1	0
1040	PHQ168	1	2	1	2	1	2	1	0	1
2041	PHQ169	1	0	0	0	0	0	0	0	0
2042	PHQ169	0	0	0	0	1	0	0	0	0
2043	PHQ170	0	0	1	1	2	3	0	0	0

ภาพที่ 3.6 ตัวอย่าง Database ของเว็บไซต์

3.4.1.4 Windows Server

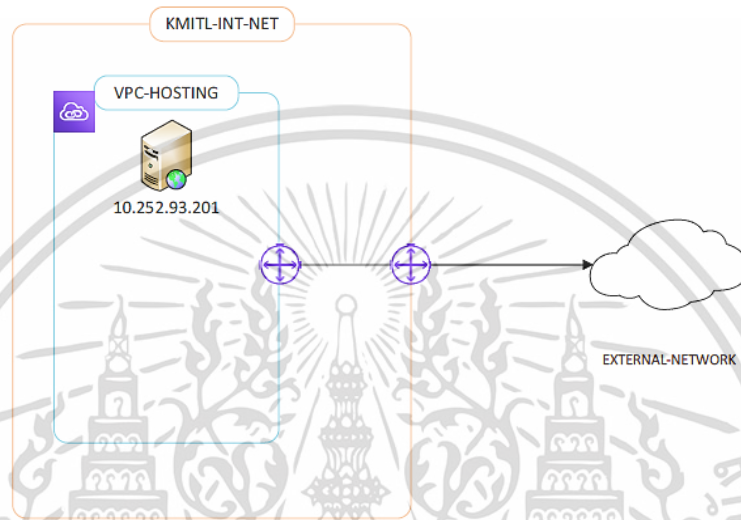
ทางผู้วิจัยเลือกใช้ Windows Server 2019 เป็น Server เพื่อ Run เว็บไซต์โดยทางผู้วิจัยได้ทำเรื่องขอเปิด Server จากทางสำนักงานคอมพิวเตอร์ของสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง โดยมีรายละเอียด ดังนี้

Requestor: Mr. Treesukorn Treebupachatsakul

Organization: ENG-BME-PROJECT

Request Date: 27-10-2022

Project name: ENG-BME-PROJECT



ภาพที่ 3.7 Server and Network diagram

No.	Server Name	OS Version	Hardware Spec.	Software Spec
1	bme	Windows 2019 STD	CPU : 4vCPU MEM : 16 GB SysDisk : 100GB	

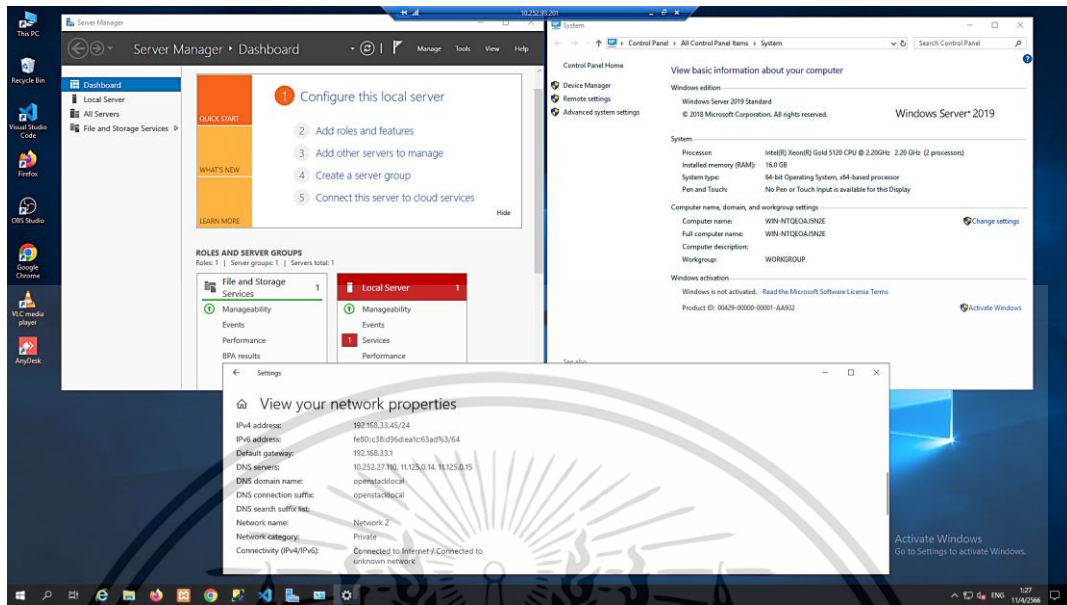
ภาพที่ 3.8 OS, Hardware and software Specification

No.	Server Name	VPC-IP	KMITL-INT-NET	EXTERNAL-NETWORK
1	bme	192.168.33.45	10.252.93.201	161.246.127.201

ภาพที่ 3.9 IP Assigned

No.	Server Name	Service	UserName	Password	Privileges
1	bme	RDP	treesukorn phanomkorn	B8CbQ4rQ64 63601218@kmitl.ac.th	Administrators Administrators

ภาพที่ 3.10 Access Matrix



ภาพที่ 3.11 ตัวอย่าง Windows Server ของเว็บไซต์

3.5 วิธีการใช้งานเว็บไซต์และตัวอย่างหน้าตาของเว็บไซต์

3.5.1 วิธีใช้งานเว็บไซต์แบ่งเป็นสองประเภทคือ 1. Online 2. Onsite

1. Online ผู้วิจัยจะนำส่งเอกสารการขอเป็นอาสาสมัครให้กับอาสาสมัครแบบ Online โดยอาสาสมัครต้องเซ็นยินยอมการเป็นอาสาสมัครก่อนถึงจะเริ่มทำแบบทดสอบได้เมื่ออาสาสมัครเซ็นใบยินยอมแล้วผู้วิจัยจะส่ง Link ของเว็บไซต์แบบทดสอบ PHQ-9 พร้อมทั้ง Username, Password ร่วมถึงคู่มือการใช้งานเว็บไซต์และอาสาสมัครสามารถติดต่อผู้วิจัยได้ทุกเมื่อ

2. Onsite ผู้วิจัยจะอธิบายการรายละเอียดการใช้งานเว็บไซต์ให้อาสาสมัครฟังรวมถึง Login เข้าสู่เว็บไซต์ให้อาสาสมัครก่อนเริ่มทำแบบทดสอบ PHQ-9 โดยอาสาสมัครทั้งหมดจะอยู่ในความดูแลของแพทย์ที่ปรึกษาของผู้วิจัย ผศ.นพ.คมสันต์ เกียรติรุ่งฤทธิ์ การเก็บข้อมูลกลุ่มอาสาสมัครกลุ่มนี้คือการไปเก็บข้อมูลตัวต่อตัวที่ ภาควิชาจิตเวชศาสตร์ คณะแพทยศาสตร์ โรงพยาบาลรามาริบัติ มหาวิทยาลัยมหิดล

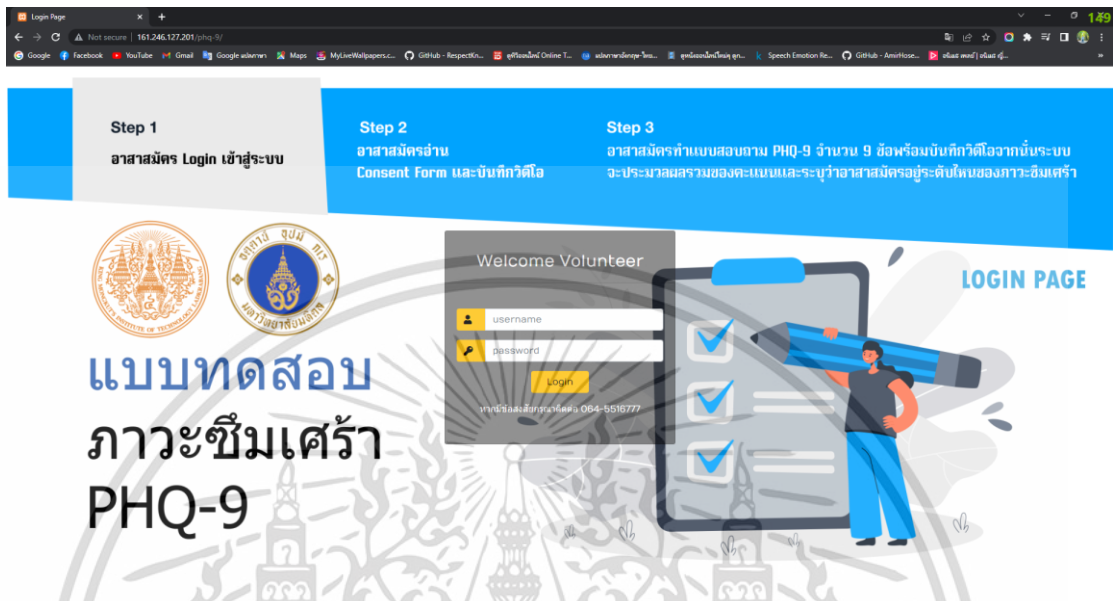
3.5.2 หน้าตาของเว็บไซต์และขั้นตอนใช้งานของแต่ละหน้า

- แผนภาพการใช้งานระบบ ดังภาพที่ 3.12



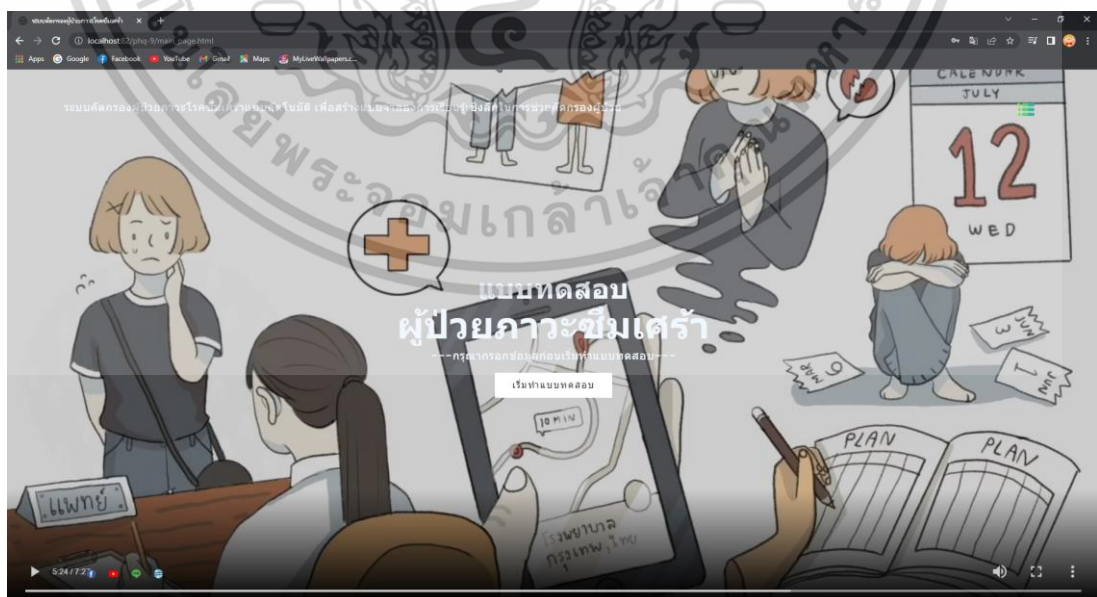
ภาพที่ 3.12 แผนภาพการใช้งานระบบ

ขั้นตอนที่ 1 ในการ Login เข้าสู่ระบบ (Username และ Password จะได้รับก็ต่อเมื่ออาสาสมัครทำแบบฟอร์มยินยอมการเป็นอาสาสมัครที่ทางผู้วิจัยจะจัดส่งให้ ดังภาพที่ 3.13



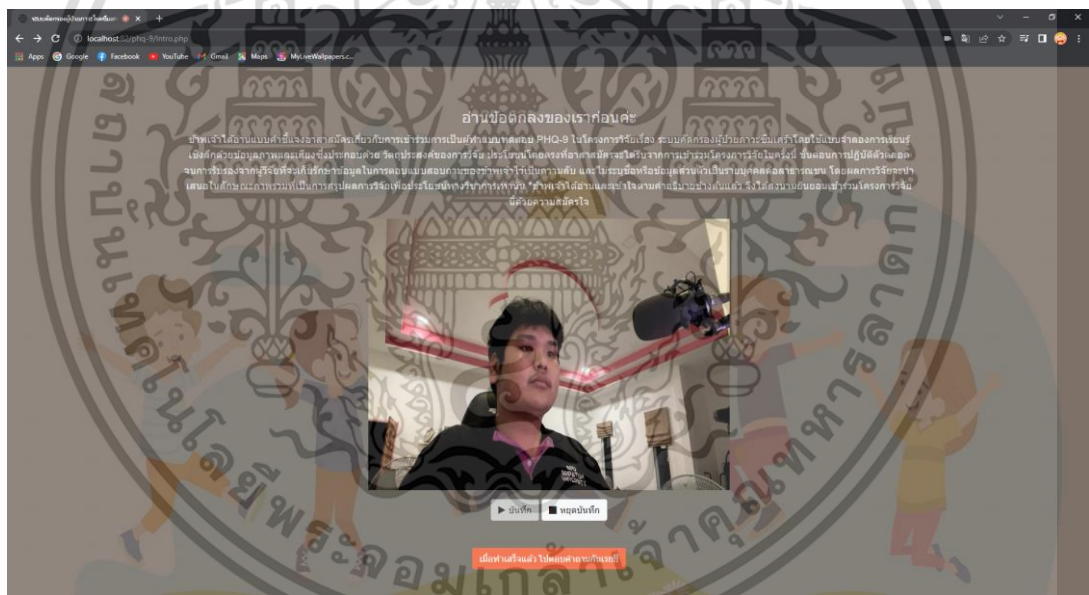
ภาพที่ 3.13 ตัวอย่าง Login Page

ขั้นตอนที่ 2 เมื่ออาสาสมัคร Login เสร็จทางระบบจะพาผู้สมัครไปยังหน้า Main ของระบบ ดังภาพที่ 3.14 หลังจากนั้นระบบจะให้อาสาสมัครรับชมวิดีโอเกี่ยวกับโรคซึมเศร้าเมื่ออาสาสมัครรับชมเสร็จ อาสาสมัครจะต้องกดปุ่มเริ่มทำแบบทดสอบเพื่อทำแบบทดสอบต่อไป



ภาพที่ 3.14 ตัวอย่าง Main Page

ขั้นตอนที่ 3 พุดคุยกันสักนิด ก่อนทำแบบสอบถามระบบจะให้อาสาสมัครอ่าน consent form ดังนี้ “ข้าพเจ้าได้อ่านแบบคำชี้แจงอาสาสมัครเกี่ยวกับการเข้าร่วมเป็นผู้ทำแบบทดสอบ PHQ-9 ในโครงการวิจัยเรื่องระบบคัดกรองผู้ป่วยภาวะซึมเศร้าโดยใช้แบบจำลองการเรียนรู้เชิงลึกด้วยข้อมูลภาพและเสียง ซึ่งประกอบด้วยวัตถุประสงค์ของการวิจัย ประโยชน์โดยตรงที่อาสาสมัครจะได้รับจากการเข้าร่วมโครงการวิจัย ขั้นตอนการปฏิบัติตัวตลอดจนการรับรองจากผู้วิจัยที่จะเก็บรักษาข้อมูลในการตอบแบบสอบถามของข้าพเจ้าไว้เป็นความลับโดยไม่ระบุชื่อหรือข้อมูลส่วนตัวต่อสาธารณชน ผลการวิจัยจะนำเสนอในลักษณะภาพรวมที่เป็นการสรุปผลการวิจัยเพื่อประโยชน์ทางวิชาการเท่านั้น” เพื่อเก็บข้อมูลภาพและเสียงของอาสาสมัครเพิ่มเติม เมื่ออาสาสมัครบันทึกวิดีโอเสร็จเรียบร้อยให้อาสาสมัครกดปุ่มหยุดบันทึกจากนั้นระบบจะปิดการบันทึกวิดีโอและให้อาสาสมัครกดปุ่มสีส้ม “เมื่อทำเสร็จแล้ว ไปตอบคำถามกันเลย!!” เพื่อเริ่มตอบแบบสอบถาม PHQ-9 โดยหน้าจอของ Intro page แสดงในภาพที่ 3.15



ภาพที่ 3.15 ตัวอย่าง Consent Form Page

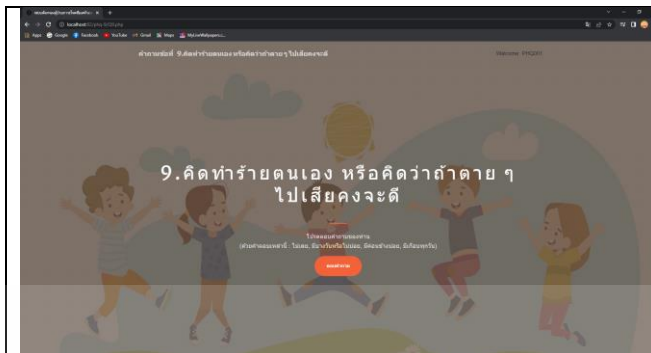
ขั้นตอนที่ 4 จากนั้นระบบจะพาอาสาสมัครมายังหน้าการตอบคำถามโดยมีทั้งหมด 9 ข้อตามแบบทดสอบภาวะซึมเศร้า PHQ-9 โดยระบบจะให้อาสาสมัครอ่านและตอบคำถามในรูปแบบการบันทึกวิดีโอ โดยหน้าตอบแบบสอบถามจะแบ่งเป็น 3 ส่วนคือ 1. ส่วนของคำถาม PHQ-9 2. ส่วนของการบันทึกวิดีโอ และ 3. ส่วนการยืนยันคำตอบ โดยทั้งหมดจะอยู่ในหน้า Page เดียว โดยส่วนที่ 1 คำถาม PHQ-9 ดังแสดงในตาราง 3.1

ตารางที่ 3.1 คำถาม PHQ-9

ตัวอย่างหน้าจอ	แบบทดสอบ PHQ-9
	<p>1. เบื่อทำอะไร ๆ ก็ไม่เพลิดเพลิน</p>
	<p>2. ไม่สบายใจ ซึมเศร้า หรือท้อแท้</p>
	<p>3. หลับยาก หรือหลับ ๆ ตื่น ๆ หรือหลับมากไป</p>
	<p>4. เหนื่อยง่าย หรือไม่ค่อยมีแรง</p>

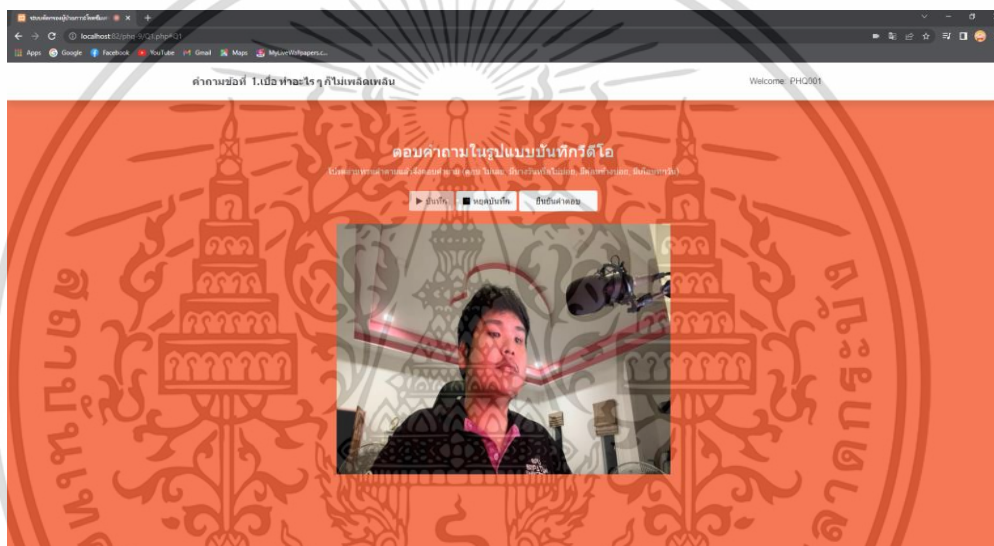
	<p>5. เบื่ออาหาร หรือกินมากเกินไป</p>
	<p>6. รู้สึกไม่ดีกับตัวเอง คิดว่าตัวเองล้มเหลว หรือเป็นคนทำให้ตัวเอง หรือครอบครัวผิดหวัง</p>
	<p>7. สมาธิไม่ดีเวลาทำอะไร เช่น ดูโทรทัศน์ ฟังวิทยุ หรือทำงานที่ต้องใช้ความตั้งใจ</p>
	<p>8. พูดหรือทำอะไรซ้ำจนคนอื่นมองเห็น หรือกระสับกระส่ายจนท่านอยู่ไม่นิ่งเหมือนเคย</p>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



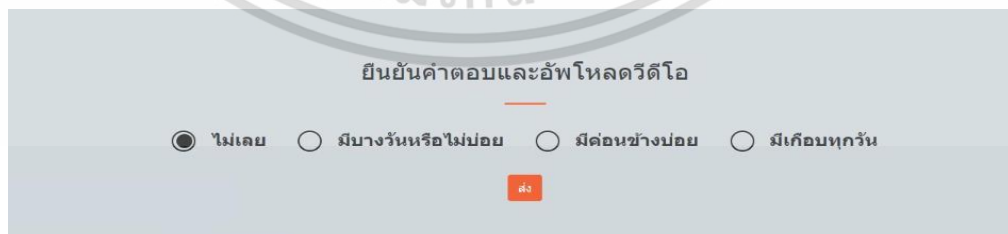
9. คิดทำร้ายตนเอง หรือคิดว่า
ถ้าตาย ๆ ไปเสียคงจะดี

- ส่วนที่ 2 ตอบแบบสอบถามด้วยการบันทึกวิดีโอแสดงในภาพที่ 3.16



ภาพที่ 3.16 ตอบแบบสอบถามด้วยการบันทึกวิดีโอ

- ส่วนที่ 3 การยืนยันคำตอบของอาสาสมัครโดยเว็บไซต์จะจับคำตอบของอาสาสมัครที่ได้บันทึกเสียงไว้และเลือกคำตอบให้อัตโนมัติแสดงในภาพที่ 3.17



ภาพที่ 3.17 ยืนยันคำตอบ

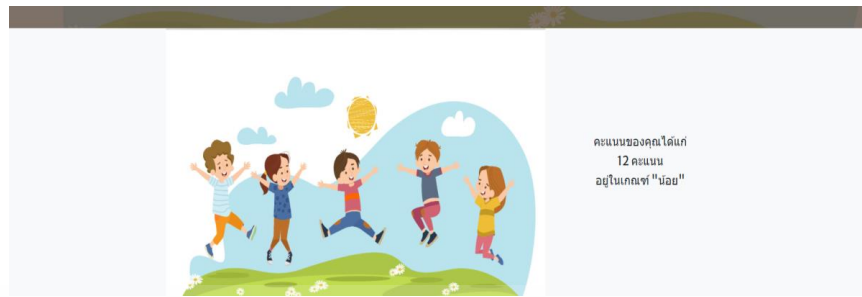


ภาพที่ 3.18 หน้าจอคำถามและตอบคำถาม PHQ-9

ขั้นตอนที่ 5 เมื่ออาสาสมัครตอบแบบสอบถามทั้งหมด 9 ข้อแล้วเว็บไซต์จะรวบรวมคะแนนของอาสาสมัครจากนั้นเว็บไซต์จะประมวลผลคะแนนรวมของอาสาสมัครรวมถึงบอกเกณฑ์ของอาสาสมัคร ดังภาพที่ 3.19 และ 3.20



ภาพที่ 3.19 หน้าจอแสดงคำอธิบายเกณฑ์คะแนน



ภาพที่ 3.20 หน้าจอแสดงผลการประเมิน

3.6 การวัดคุณภาพของ API บนเว็บไซต์

- ผู้วิจัยได้วัดคุณภาพของการเลือกคำตอบอัตโนมัติของ API โดยหลังบ้านของเว็บไซต์จะตรวจสอบคำพูดของอาสาสมัครโดยอัตโนมัติ และเว็บไซต์จะเลือกคำตอบของอาสาสมัครและเติมลงช่องคำตอบอัตโนมัติ โดยที่อาสาสมัครไม่ต้องกดเลือกเอง ทางผู้วิจัยได้เก็บค่าความแม่นยำในการเลือกคำตอบโดยอัตโนมัติของเว็บไซต์เพื่อทดสอบคุณภาพของ API โดยมีรายละเอียดดังนี้ 1. จำนวนอาสาสมัคร 100 ท่าน 2. จำนวนคำถาม 9 ข้อ 3. จำนวนคำตอบที่เลือกถูกต้อง 4. จำนวนคำตอบที่เลือกผิด 5. คิดเป็นกิบเปอร์เซ็นต์ ดังตารางที่ 3.2

ตารางที่ 3.2 คุณภาพของเว็บไซต์ PHQ-9

อาสาสมัคร	ทายถูก (ข้อ)	ทายผิด (ข้อ)	คิดเป็น %
PHQ001	9	0	100%
PHQ002	9	0	100%
PHQ003	3	6	40%
PHQ004	7	2	80%
PHQ005	5	4	60%
PHQ006	9	0	100%
PHQ007	7	2	80%
PHQ008	9	0	100%
PHQ009	8	1	90%
PHQ010	8	1	90%
PHQ011	9	0	100%
PHQ012	6	3	70%

PHQ013	7	2	80%
PHQ014	9	0	100%
PHQ015	8	1	90%
PHQ016	6	3	70%
PHQ017	9	0	100%
PHQ018	7	2	80%
PHQ019	7	2	80%
PHQ020	8	1	90%
PHQ021	3	6	40%
PHQ022	8	1	90%
PHQ023	9	0	100%
PHQ024	9	0	100%
PHQ025	7	2	80%
PHQ026	7	2	80%
PHQ027	9	0	100%
PHQ028	7	2	80%
PHQ029	9	0	100%
PHQ030	9	0	100%
PHQ031	9	0	100%
PHQ032	8	1	90%
PHQ033	8	1	90%
PHQ034	9	0	100%
PHQ035	9	0	100%
PHQ036	9	0	100%
PHQ037	8	1	90%
PHQ038	9	0	100%
PHQ039	7	2	80%
PHQ040	7	2	80%
PHQ041	9	0	100%

PHQ042	9	0	100%
PHQ043	9	0	100%
PHQ044	9	0	100%
PHQ045	9	0	100%
PHQ046	9	0	100%
PHQ047	7	2	80%
PHQ048	8	1	90%
PHQ049	6	3	70%
PHQ050	9	0	100%
PHQ051	9	0	100%
PHQ052	8	1	100%
PHQ053	9	0	100%
PHQ054	9	0	100%
PHQ055	9	0	100%
PHQ056	9	0	100%
PHQ057	8	1	90%
PHQ058	3	6	40%
PHQ059	8	1	90%
PHQ060	9	0	100%
PHQ061	6	3	70%
PHQ062	7	2	80%
PHQ063	9	0	100%
PHQ064	9	0	100%
PHQ065	9	0	100%
PHQ066	9	0	100%
PHQ067	9	0	100%
PHQ068	7	2	80%
PHQ069	8	1	90%
PHQ070	9	0	100%

PHQ071	9	0	100%
PHQ072	9	0	100%
PHQ073	9	0	100%
PHQ074	9	0	100%
PHQ075	9	0	100%
PHQ076	9	0	100%
PHQ077	9	0	100%
PHQ078	8	1	90%
PHQ079	9	0	100%
PHQ080	9	0	100%
PHQ081	9	0	100%
PHQ082	9	0	100%
PHQ083	9	0	100%
PHQ084	8	1	90%
PHQ085	8	1	90%
PHQ086	8	1	90%
PHQ087	9	0	100%
PHQ088	8	1	90%
PHQ089	7	2	80%
PHQ090	9	0	100%
PHQ091	6	3	70%
PHQ092	9	0	100%
PHQ093	6	3	70%
PHQ094	9	0	100%
PHQ095	9	0	100%
PHQ096	9	0	100%
PHQ097	9	0	100%
PHQ098	9	0	100%
PHQ099	6	3	70%

PHQ100	9	0	100%
--------	---	---	------

โดยค่าเฉลี่ยของเว็บไซต์เลือกคำตอบตามที่อาสาสมัครยืนยันอยู่ที่ 91% ถือว่ายอดเยี่ยม

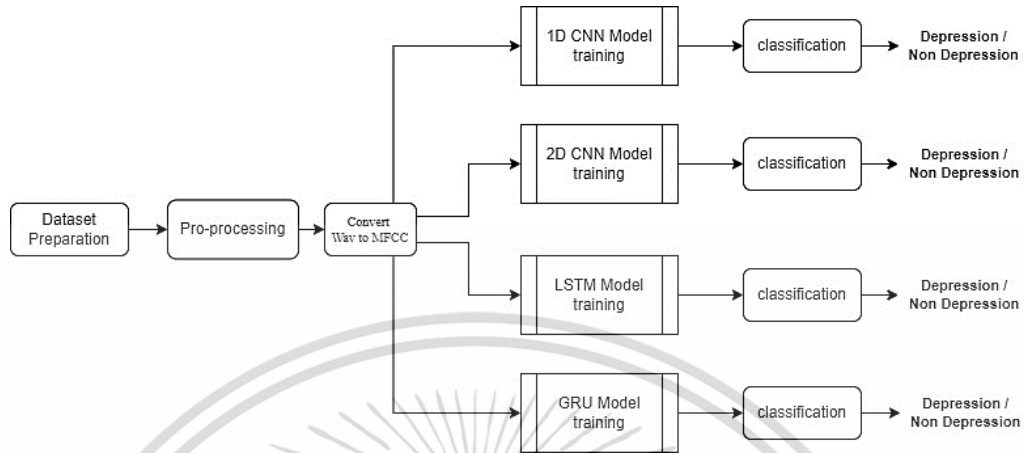
3.7 การพัฒนาโมเดลการเรียนรู้เชิงลึก (Deep Learning)

ผู้วิจัยได้สร้างแบบจำลองการเรียนรู้เชิงลึกออกเป็น 4 โมเดล ได้แก่ 1D CNN, 2D CNN, LSTM และ GRU โดยสถาปัตยกรรมการเรียนรู้เชิงลึกของ CNN, LSTM และ GRU เป็นที่นิยมสำหรับชุดข้อมูลคลื่นเสียง CNN แบบ 1 มิติและ 2 มิติคือ Convolutional Neural Network (CNN) ซึ่ง CNN จำลองการมองเห็นของมนุษย์ด้วยการดึงคุณสมบัติอัตโนมัติ โมเดลโครงข่ายประสาทเทียมแบบวนซ้ำ (RNN) สามารถประมวลผลได้ไม่เพียงแค่จุดข้อมูลเดียว (เช่น รูปภาพ) แต่ยังรวมถึงลำดับข้อมูลทั้งหมด (เช่น เสียงพูดหรือวิดีโอ) LSTM มีการเชื่อมต่อข้อเสนอแนะ GRU เป็นเหมือนหน่วยความจำระยะสั้นระยะยาวที่มีเกทการลืม แต่มีพารามิเตอร์น้อยกว่า LSTM เนื่องจากไม่มีเกทเอาต์พุต ประสิทธิภาพของ GRU ในงานบางอย่างของการสร้างแบบจำลองดนตรีโพลีโฟนิค การสร้างแบบจำลองสัญญาณเสียงพูด และภาษาธรรมชาติ

3.7.1 การพัฒนาแบบจำลองการเรียนรู้เชิงลึกเพื่อแยกแยะระหว่างมีภาวะซึมเศร้าและไม่มีภาวะซึมเศร้าหรือระดับของภาวะซึมเศร้า

เมื่อผู้วิจัยได้ข้อมูลเสียงจากอาสาสมัครเป็นที่เรียบร้อยแล้วทางผู้วิจัยจะนำข้อมูลที่ได้ไปจัดทำ Dataset เพื่อสร้างแบบจำลองการเรียนรู้เชิงลึกโดยทางผู้วิจัยจะแบ่ง Dataset ออกเป็นสองรูปแบบคือ 1. ผู้มีภาวะซึมเศร้าและไม่มีภาวะซึมเศร้า 2. แบ่งออกเป็น 5 กลุ่มตามเกณฑ์ของ PHQ-9 โดยทั้งสองรูปแบบมีพื้นฐานมาจากผลของคะแนนของ PHQ-9 ดังภาพที่ 3.21

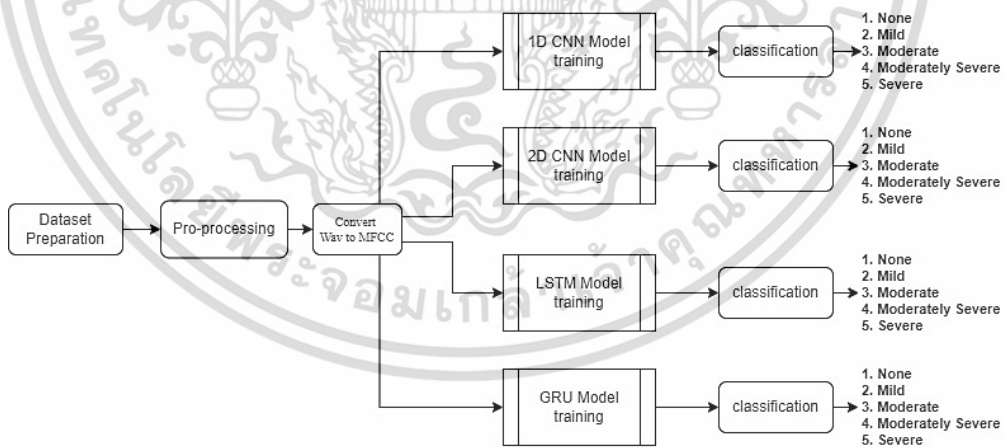
Workflow Diagram ของรูปแบบที่ 1 ผู้มีภาวะซึมเศร้าและไม่มีภาวะซึมเศร้า



ภาพที่ 3.21 Workflow Diagram ของรูปแบบที่ 1 ผู้มีภาวะซึมเศร้าและไม่มีภาวะซึมเศร้า

จาก Workflow Diagram จะเห็นว่าการพัฒนาโมเดลการเรียนรู้เชิงลึก (Deep Learning) แบ่งออกเป็น 5 ขั้นตอนได้แก่ 1. การเตรียมข้อมูลผู้มีภาวะซึมเศร้าและไม่มีภาวะซึมเศร้า 2. การประมวลผลข้อมูลผู้มีภาวะซึมเศร้าและไม่มีภาวะซึมเศร้า 3. แปลงคุณสมบัติไฟล์ WAV เป็น MFCC 4. Training โมเดลทั้ง 4 โมเดล 5. แยกแยะประเภทของมีภาวะซึมเศร้าและไม่มีภาวะซึมเศร้า ดังภาพที่ 3.22

Workflow Diagram ของรูปแบบที่ 2 แบ่งออกเป็น 5 ประเภทตามเกณฑ์ของ PHQ-9



ภาพที่ 3.22 Workflow Diagram ของรูปแบบที่ 2 แบ่งออกเป็น 5 ประเภทตามเกณฑ์ของ PHQ-9

จาก Workflow Diagram จะเห็นว่าการพัฒนาโมเดลการเรียนรู้เชิงลึก (Deep Learning) แบ่งออกเป็น 5 ขั้นตอนได้แก่ 1. การเตรียมข้อมูลผู้มีภาวะซึมเศร้าและไม่มีภาวะซึมเศร้า 2. การประมวลผลข้อมูลผู้มีภาวะซึมเศร้าและไม่มีภาวะซึมเศร้า 3. แปลงคุณสมบัติไฟล์ WAV เป็น MFCC 4. Training โมเดลทั้ง 4 โมเดล 5. แยกแยะประเภทตามเกณฑ์ทั้ง 5 ของ PHQ9

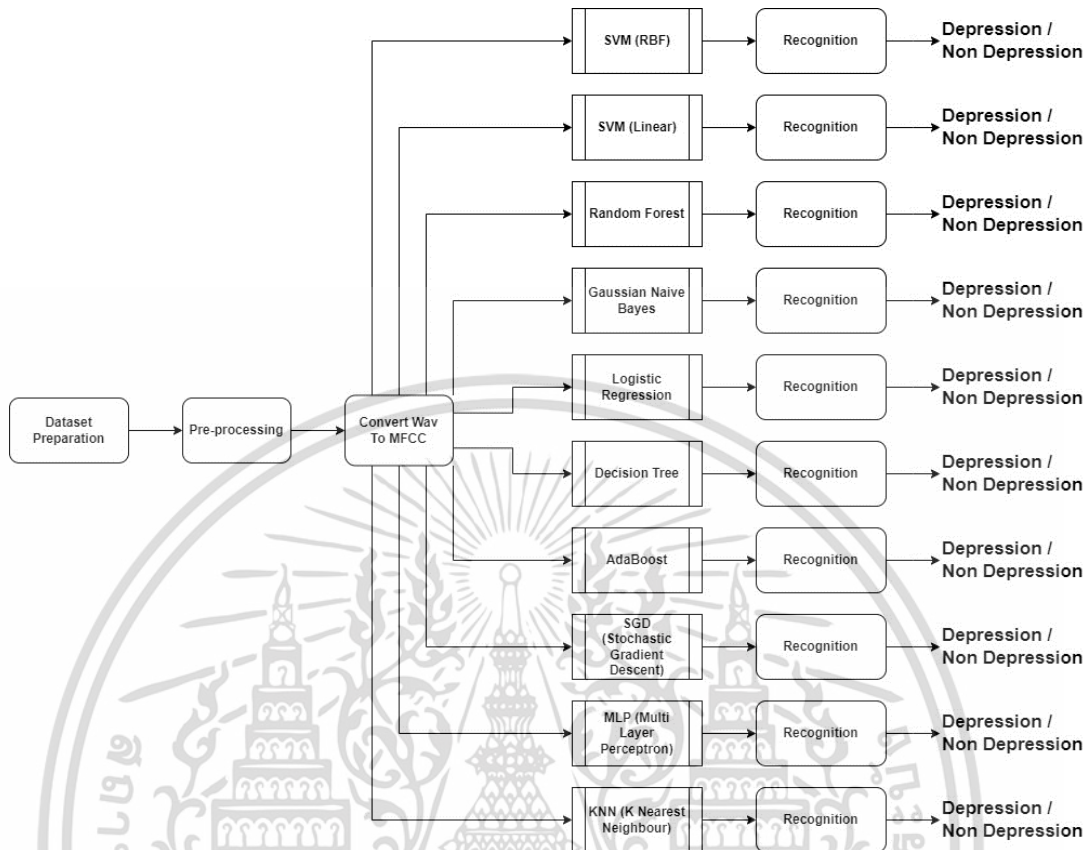
3.8 การสร้างโมเดลการเรียนรู้ของเครื่อง (Machine Learning)

ผู้วิจัยได้สร้างแบบจำลองการเรียนรู้ของเครื่องออกเป็น 10 โมเดล ได้แก่ Support Vector Machines (SVM (RBF)), Support Vector Machines (SVM (Linear)), Random Forest, Gaussian Naïve Bayes, Logistic Regression, Decision Tree, AdaBoost, SGD (Stochastic Gradient Descent), MLP (Multi-Layer Perceptron) และ KNN (K Nearest Neighbour) โดยทางผู้วิจัยได้เสนอการทดลองของการสร้างโมเดลการเรียนรู้เชิงลึกไปแล้วทางผู้วิจัยจึงอยากนำเสนอการสร้างโมเดลการเรียนรู้ของเครื่องว่าจะมีประสิทธิภาพในการแยกแยะผู้มีภาวะซึมเศร้าหรือไม่มีภาวะซึมเศร้า และแบ่งออกเป็น 5 ประเภทตามเกณฑ์ของ PHQ-9 ได้หรือไม่โดยมีรูปแบบดังต่อไปนี้

3.8.1 การพัฒนาแบบจำลองการเรียนรู้ของเครื่องเพื่อเรียนรู้เสียงของผู้มีภาวะซึมเศร้าและไม่มีภาวะซึมเศร้าหรือระดับของภาวะซึมเศร้า

เมื่อผู้วิจัยได้ข้อมูลเสียงจากอาสาสมัครเป็นที่เรียบร้อยแล้วทางผู้วิจัยจะนำข้อมูลที่ได้ไปจัดทำ Dataset เพื่อสร้างแบบจำลองการเรียนรู้ของเครื่องโดยทางผู้วิจัยจะแบ่ง Dataset ออกเป็นสองรูปแบบคือ 1. ผู้มีภาวะซึมเศร้าและไม่มีภาวะซึมเศร้า 2. แบ่งออกเป็น 5 กลุ่มตามเกณฑ์ของ PHQ-9 โดยทั้งสองรูปแบบมีพื้นฐานมาจากผลของคะแนนของ PHQ-9 ดังภาพที่ 3.23

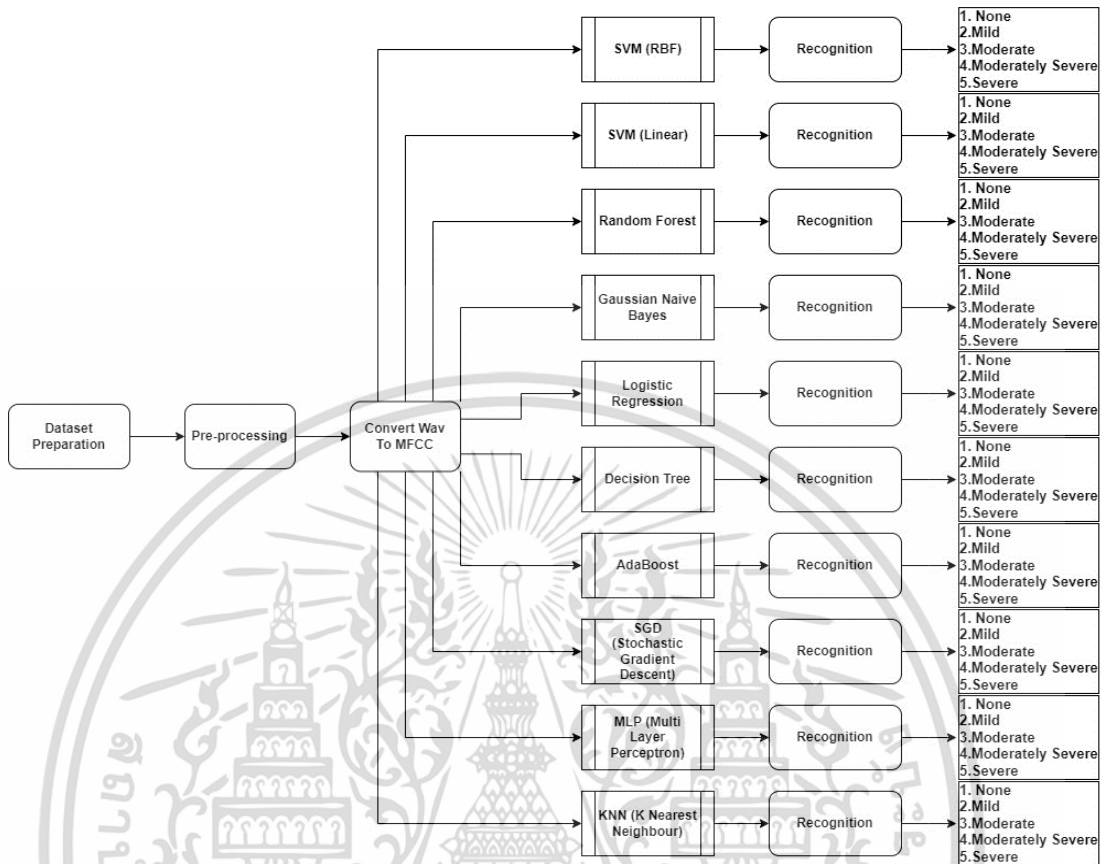
Workflow Diagram ของรูปแบบที่ 1 ผู้มีภาวะซึมเศร้าและไม่มีภาวะซึมเศร้าของ 10 Model



ภาพที่ 3.23 Workflow Diagram ของรูปแบบที่ 1 ผู้มีภาวะซึมเศร้าและไม่มีภาวะซึมเศร้าของ 10 Model

จาก Workflow Diagram จะเห็นว่าการพัฒนาโมเดลการเรียนรู้ของเครื่อง (Machine Learning) แบ่งออกเป็น 5 ขั้นตอนได้แก่ 1. การเตรียมข้อมูลผู้มีภาวะซึมเศร้าและไม่มีภาวะซึมเศร้า 2. การประมวลผลข้อมูลผู้มีภาวะซึมเศร้าและไม่มีภาวะซึมเศร้า 3. แปลงคุณสมบัติไฟล์ WAV เป็น MFCC 4. Training โมเดลทั้ง 10 โมเดล 5. แยกแยะประเภทของมีภาวะซึมเศร้าและไม่มีภาวะซึมเศร้า ดังภาพที่ 3.24

Workflow Diagram ของรูปแบบที่ 2 แบ่งออกเป็น 5 ประเภทตามเกณฑ์ของ PHQ-9 ของ 10 Model



ภาพที่ 3.34 Workflow Diagram ของรูปแบบที่ 2 แบ่งออกเป็น 5 ประเภทตามเกณฑ์ของ PHQ-9 ของ 10 Model

จาก Workflow Diagram จะเห็นว่าการพัฒนาโมเดลการเรียนรู้ของเครื่อง (Machine Learning) แบ่งออกเป็น 5 ขั้นตอนได้แก่ 1. การเตรียมข้อมูลผู้มีภาวะซึมเศร้าและไม่มีภาวะซึมเศร้า 2. การประมวลผลข้อมูลมีภาวะซึมเศร้าและไม่มีภาวะซึมเศร้า 3. แปลงคุณสมบัติไฟล์ WAV เป็น MFCC 4. Training โมเดลทั้ง 10 โมเดล 5. แยกแยะประเภทตามเกณฑ์ทั้ง 5 ของ PHQ9

3.9 ขั้นตอนการพัฒนาแบบจำลองการเรียนรู้เชิงลึก (Deep Learning) และแบบจำลองการเรียนรู้ของเครื่อง (Machine Learning)

3.9.1 Dataset Preparation

รูปแบบที่ 1 Dataset สามารถแบ่งเป็น 2 กลุ่มได้แก่ ผู้ที่มีภาวะซึมเศร้าและผู้ที่ไม่มีภาวะซึมเศร้า โดยมีรายละเอียดดังนี้ อาสาสมัครที่ทำคะแนนแบบทดสอบ PHQ-9 ได้ 0-14 คะแนนถือว่าเป็นไม่มีภาวะซึมเศร้า ส่วนอาสาสมัครที่มีคะแนน 15-27 คะแนนถือว่าเป็นมีภาวะซึมเศร้า ดังตารางที่ 3.3

ตารางที่ 3.3 แบ่งประเภทผู้มีภาวะซึมเศร้าและไม่มีภาวะซึมเศร้าโดยใช้พื้นฐานคะแนนของแบบทดสอบ

PHQ-9

PHQ-9 Score		
คะแนน	ระดับ	ประเภท
0 - 4	ไม่มีอาการ	ไม่มีภาวะซึมเศร้า
5 - 8	น้อย	
9 - 14	ปานกลาง	
15 - 19	มาก	มีภาวะซึมเศร้า
20 - 27	รุนแรง	

รูปแบบที่ 2 PHQ-9 จะมีเกณฑ์ทั้งหมด 5 ระดับผู้วิจัยได้ทำการจัดกลุ่มข้อมูลเสียงที่ได้จากอาสาสมัคร ซึ่งแบ่งออกเป็น 5 กลุ่มคือ

1. คะแนนตามเกณฑ์ PHQ-9 ในระดับ 0 – 4 ผลการประเมินอยู่ในระดับไม่มีอาการ
2. คะแนนตามเกณฑ์ PHQ-9 ในระดับ 5 – 8 ผลการประเมินอยู่ในระดับน้อย
3. คะแนนตามเกณฑ์ PHQ-9 ในระดับ 9 – 14 ผลการประเมินอยู่ในระดับปานกลาง
4. คะแนนตามเกณฑ์ PHQ-9 ในระดับ 15 – 19 ผลการประเมินอยู่ในระดับมาก
5. คะแนนตามเกณฑ์ PHQ-9 ในระดับ 20 – 27 ผลการประเมินอยู่ในระดับรุนแรง

ตารางที่ 3.4 แบ่งกลุ่มผู้มีภาวะซึมเศร้า 5 ระดับโดยใช้เกณฑ์คะแนนของแบบทดสอบ PHQ-9

PHQ-9 Score	
คะแนน	ระดับ
0 - 4	ไม่มีอาการ
5 - 8	น้อย
9 - 14	ปานกลาง
15 - 19	มาก
20 - 27	รุนแรง

3.9.2 Data Preprocessing

สำหรับข้อมูลเสียงเป็นกระบวนการที่ทำให้ข้อมูลเสียงพร้อมสำหรับการวิเคราะห์หรือการประมวลผลเสียง ดังนั้นมีขั้นตอนและเทคนิคที่เฉพาะเจาะจงที่เกี่ยวข้องกับข้อมูลเสียงในงานวิจัยดังนี้

- อ่านไฟล์เสียง (Audio File Reading) การเริ่มกระบวนการด้วยการอ่านไฟล์เสียงจากแหล่งที่เก็บข้อมูล เช่น WAV, MP3, FLAC, หรือไฟล์เสียงอื่น ๆ

- Sampling Rate ข้อมูลเสียงเป็นสัญญาณอนาล็อก ต้องถูกแอสตมป์ลงในรูปแบบตัวเลข โดยการสุ่มค่าที่จุดตัวอย่างเป็นระยะๆ (เช่น 44.1 kHz หรือ 48 kHz) ต่อวินาทีเพื่อสร้างแบบแยกตัวอย่าง (Samples) โดยงานวิจัยนี้ใช้ Sampling Rate อยู่ที่ 16 kHz

- การตัดเสียง (Audio Segmentation) หากมีการตัดเอาเสียงบางส่วนจากไฟล์เสียงหรือต้องการแบ่งช่วงของเสียงเพื่อวิเคราะห์ เช่น การตัดเสียงพูดจากพื้นหลังหรือตัดช่วงที่มีเสียงพัก

- การสกัดคุณลักษณะเสียง (Feature Extraction) การสกัดคุณลักษณะเสียงเป็นขั้นตอนสำคัญที่เป็นการแปลงเสียงให้กลายเป็นเวกเตอร์ของคุณลักษณะเช่น Mel-frequency cepstral coefficients (MFCCs)

- ตัวอย่าง Source Code

```
df = pd.DataFrame(columns=['feature'])
# loop feature extraction over the entire dataset
counter=0
for index,path in enumerate(ref.path):
    X, sample_rate = librosa.load(path,
                                   ,res_type='kaiser_fast'
                                   ,duration=16
                                   ,sr=16000
                                   ,offset=0.5
                                   )
    sample_rate = np.array(sample_rate)

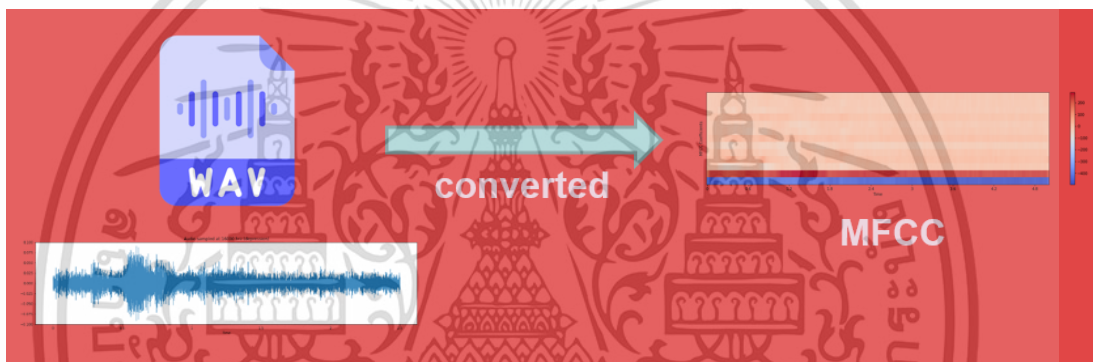
    # mean as the feature. Could do min and max etc as well.
    mfccs = np.mean(librosa.feature.mfcc(y=X,
                                         sr=sample_rate,
                                         n_mfcc=13),
                  axis=0)
    df.loc[counter] = [mfccs]
    counter=counter+1

# Check a few records to make sure its processed successfully
print(len(df))
```

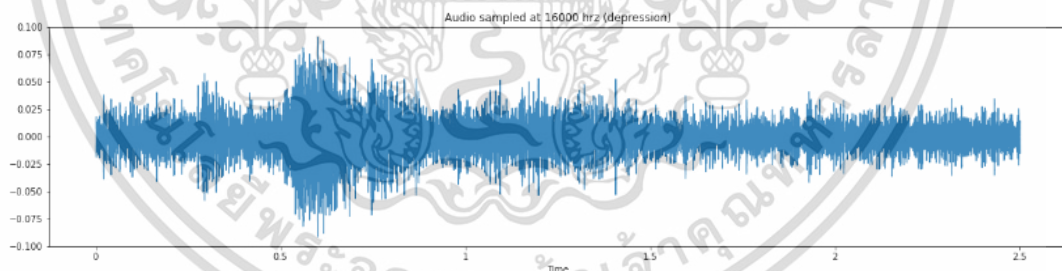
```
df.head()
-----
sampling_rate=16000
audio_duration=16
n_mfcc = 30
mfcc = prepare_data(ref, n = n_mfcc, aug = 0, mfcc = 1)
```

3.9.3 การแปลงไฟล์เสียง WAV เป็น MFCC

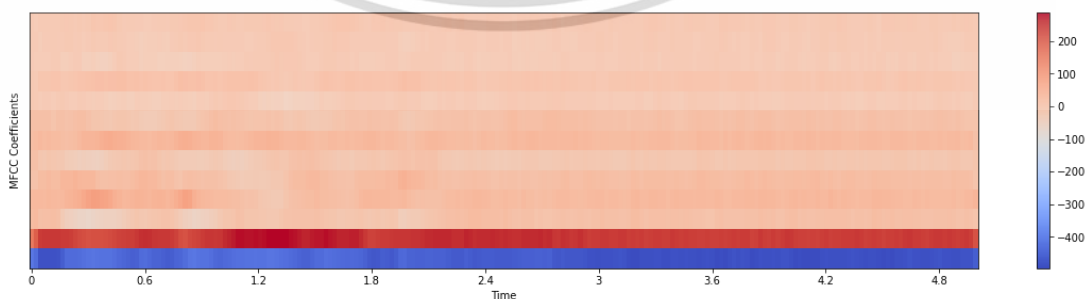
ผู้วิจัยได้นำ Dataset ที่เก็บรวบรวมได้ในรูปแบบ WAV จากนั้นนำไฟล์เข้ากระบวนการแปลงรูปแบบไฟล์ให้ออกมาอยู่ในรูปแบบของ MFCC โดยใช้ Python.Librosa เป็นตัวช่วย ดังภาพที่ 3.25, 3.26 และ 3.27



ภาพที่ 3.25 การแปลงคุณสมบัติจาก WAV เป็น MFCC



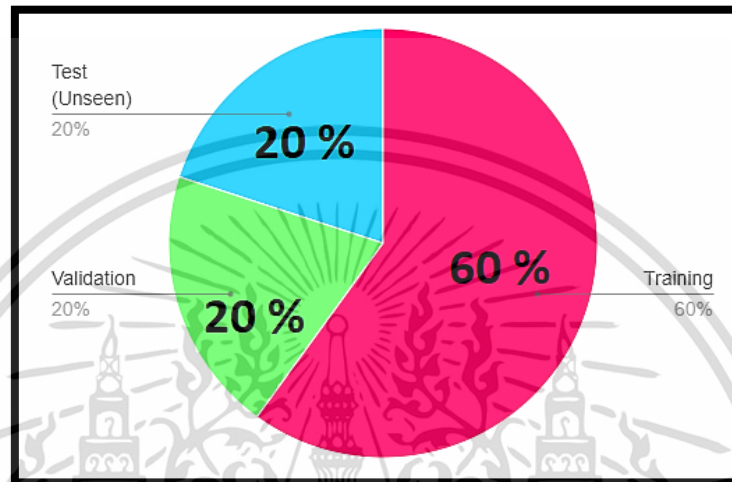
ภาพที่ 3.26 ตัวอย่างของการบันทึกในรูปแบบเสียงคลื่น (WAV)



ภาพที่ 3.27 ตัวอย่างของรูปแบบ MFCC ที่แปลงแล้ว

3.9.4 การแบ่งข้อมูลของงานวิจัย

โดยผู้วิจัยแบ่งข้อมูลออกเป็น 3 รูปแบบได้แก่ 1. Training set, 2. Validation set, 3. Unseen Test set ดังภาพที่ 3.28



ภาพที่ 3.28 การแบ่งข้อมูลทั้ง 3 รูปแบบ ผู้วิจัยแบ่งข้อมูลทั้งหมดออกเป็น Training set จำนวน 60% Validation set จำนวน 20% และ Unseen Test set จำนวน 20%

บทที่ 4

ผลการทดลอง

4.1 การเก็บข้อมูลภาพและเสียง

จากการเก็บข้อมูลทั้งภาพและเสียงทั้งหมดของอาสาสมัครได้ทั้งหมด 100 คนได้ไฟล์เสียงทั้งหมด 1000 ไฟล์ แบ่งข้อมูลออกเป็น 2 ประเภทคือ 1. การจำแนกประเภทระหว่างมีภาวะซึมเศร้าและไม่มีภาวะซึมเศร้า 2. การจำแนกประเภทที่แบ่งออกเป็น 5 ประเภทตามเกณฑ์ของ PHQ-9 ได้แก่ 1. ไม่มีอาการ 2. น้อย 3. ปานกลาง 4. มาก 5. รุนแรง

ประเภทที่ 1 การจำแนกประเภทระหว่างมีภาวะซึมเศร้าและไม่มีภาวะซึมเศร้า

ตารางที่ 4.1 ตารางแสดงจำนวนการเก็บข้อมูลอาสาสมัครประเภทที่ 1

ประเภท	อาสาสมัคร	จำนวนไฟล์
ไม่มีภาวะซึมเศร้า	75	742
มีภาวะซึมเศร้า	25	242

ประเภทที่ 2 การจำแนกประเภทที่แบ่งออกเป็น 5 ประเภทตามเกณฑ์ของ PHQ-9 ได้แก่

1. ไม่มีอาการ 2. น้อย 3. ปานกลาง 4. มาก 5. รุนแรง

ตารางที่ 4.2 ตารางแสดงจำนวนการเก็บข้อมูลอาสาสมัครประเภทที่ 2

ประเภท	อาสาสมัคร	จำนวนไฟล์
ไม่มีอาการ	17	145
มีภาวะน้อย	29	254
ปานกลาง	30	280
มาก	15	138
รุนแรง	9	74

ตารางต่อไปนี้เป็นตารางที่แบ่งตามเพศของอาสาสมัคร ชาย และ หญิง

ตารางที่ 4.3 ตารางแสดงจำนวนการเก็บข้อมูลอาสาสมัครแบ่งตามเพศ

ประเภท	อาสาสมัคร	จำนวนไฟล์
ชาย	31	310
หญิง	69	690

4.2 ผลการทดลองแบบจำลองการเรียนรู้เชิงลึกและการเรียนรู้ของเครื่อง

4.2.1 ประเภทที่ 1 การจำแนกประเภทระหว่างมีภาวะซึมเศร้าและไม่มีภาวะซึมเศร้า

โดยผู้วิจัยได้ทำการ Balance ข้อมูลเพื่อเพิ่มความแม่นยำของโมเดลจึงได้จำนวนไฟล์และ
อสาสมัครดังนี้

ตารางที่ 4.4 ตารางแสดงจำนวนการเก็บข้อมูลอาสาสมัครประเภทที่ 1 แบบ Balance ข้อมูล

ประเภท	อาสาสมัคร	จำนวนไฟล์
ไม่มีภาวะซึมเศร้า	24	210
มีภาวะซึมเศร้า	24	210

โดยผู้วิจัยได้แบ่งข้อมูลออกเป็น 1. Training set = 60% 2. Validation set = 20% 3.
Unseen Test set = 20% ดังตารางที่ 4.5

ตารางที่ 4.5 ตารางแสดงจำนวนการแบ่งข้อมูล Training set = 60% 2. Validation set = 20%
3. Unseen Test set = 20%

ประเภท (ไฟล์)	Training set	Validation set	Unseen test set
ไม่มีภาวะซึมเศร้า	134	34	42
มีภาวะซึมเศร้า	134	34	42

4.2.1.1 ตารางแสดงผลการทดลองแบบจำลองการเรียนรู้เชิงลึกและการเรียนรู้ของเครื่อง

ตารางแสดงค่า Precision, Recall, F1-Score และ Accuracy ของการจำแนกประเภท
ระหว่างมีภาวะซึมเศร้าและไม่มีภาวะซึมเศร้า

ตารางที่ 4.6 ผลของ Training Result, Validation Result, Test Result แสดงค่า Precision, Recall, F1-Score และ Accuracy ของการจำแนกประเภทระหว่างมีภาวะซึมเศร้าและไม่มีภาวะซึมเศร้าของทั้ง 4 Model (1D CNN, 2D CNN, LSTM, GRU)

Architecture	Training Result					Validation Result					Test Result				
	precision	recall	f1-score	support	accuracy	precision	recall	f1-score	support	accuracy	precision	recall	f1-score	support	accuracy
1D CNN															
depression	1.00	1.00	1.00	134	1.00	0.69	0.59	0.63	34	0.66	1.00	0.40	0.58	42	0.70
non-depression	1.00	1.00	1.00	134		0.64	0.74	0.68	34		0.63	1.00	0.77	42	
2D CNN															
depression	1.00	1.00	1.00	134	1.00	1.00	1.00	1.00	34	1.00	0.95	0.95	0.95	42	0.95
non-depression	1.00	1.00	1.00	134		1.00	1.00	1.00	34		0.95	0.95	0.95	42	
LSTM															
depression	0.92	0.08	0.15	134	0.54	0.00	0.00	0.00	34	0.49	0.33	0.02	0.04	42	0.49
non-depression	0.52	0.99	0.68	134		0.49	0.97	0.65	34		0.46	0.95	0.65	42	
GRU															
depression	0.51	0.92	0.66	134	0.52	0.54	1.00	0.70	34	0.57	0.50	0.05	0.09	42	0.50
non-depression	0.59	0.12	0.20	134		1.00	0.15	0.26	34		0.50	0.95	0.66	42	

จากตารางสรุปได้ดังนี้

1D CNN

ความแม่นยำในข้อมูลการฝึก (Training Result) คือ 1.00 หมายถึงระบบการจำแนกมีความแม่นยำที่ 100% ในการจำแนกข้อมูลในช่วงการฝึกนี้เป็นผลที่ค่อนข้างดีและบ่งชี้ว่าระบบได้เรียนรู้ข้อมูลการฝึกได้ดี

ความแม่นยำในข้อมูลการตรวจสอบ (Validation Result) คือ 0.66 หมายถึงระบบการจำแนกมีความแม่นยำที่ประมาณ 66% ในการจำแนกข้อมูลในช่วงการตรวจสอบ นี่เป็นค่าที่ต่ำกว่าความแม่นยำในข้อมูลการฝึกและอาจบ่งชี้ถึงปัญหาการถดถอยที่เรียนรู้ข้อมูลการฝึกอาจไม่สามารถใช้กับข้อมูลใหม่ได้อย่างดี

ความแม่นยำในข้อมูลการทดสอบ (Test Result) คือ 0.70 หมายถึงระบบการจำแนกมีความแม่นยำที่ประมาณ 70% ในการจำแนกข้อมูลในช่วงการทดสอบ นี่เป็นค่าที่สูงกว่าความแม่นยำในข้อมูลการตรวจสอบแต่ก็ยังไม่สูงมาก อาจจะต้องพิจารณาการปรับปรุงระบบการจำแนกหรือการใช้ชุดข้อมูลที่มากขึ้นในการทดสอบเพื่อปรับปรุงความแม่นยำ

2D CNN

ค่าความแม่นยำบนชุดข้อมูลการฝึกการเรียนรู้ (Training Result) คือ 1.00 หมายถึงโมเดลสามารถทำนายข้อมูลในชุดข้อมูลการฝึกได้ทั้งหมดถูกต้อง

ค่าความแม่นยำบนชุดข้อมูลการทดสอบ (Validation Result) คือ 1.00 หมายถึงโมเดลมีความแม่นยำในการทำนายทั้งหมด แสดงถึงประสิทธิภาพที่สูงสุดในการจำแนกคลาสในชุดข้อมูลนี้ และการทำนายมีความแม่นยำถูกต้องสูงสุดที่เป็นไปได้

ค่าความแม่นยำบนชุดข้อมูลการตรวจสอบ (Test Result) คือ 0.95 แสดงถึงประสิทธิภาพในการทำนายบนข้อมูลที่ไม่ได้ใช้ในการฝึกการเรียนรู้ ค่าความแม่นยำอยู่ 0.95 หรือ 95%

ข้อมูลนี้ชี้ให้เห็นว่าโมเดลมีความแม่นยำสูงมากในการทำนายทั้งคลาสในทุกชุดข้อมูล และค่าความแม่นยำบนชุดข้อมูลการทดสอบเป็นสถานะที่ดีและน่าเชื่อถือ เป็นสัญญาณที่ดีว่าโมเดลน่าจะสามารถทำนายข้อมูลในสถานการณ์จริงๆ ได้อย่างมีประสิทธิภาพ

LSTM

โมเดลมีความแม่นยำในการทำนายที่ต่ำบนทั้งชุดข้อมูลการฝึกการเรียนรู้ (Training Result) และชุดข้อมูลการตรวจสอบ (Validation Result) แสดงว่ามีความแม่นยำในการทำนายที่ต่ำกว่า 50% บนทั้งสองชุดข้อมูลนี้

ความแม่นยำบนชุดข้อมูลการทดสอบ (Test Result) ยิ่งต่ำกว่าทั้งชุดข้อมูลการฝึกการเรียนรู้และการตรวจสอบ แสดงว่าโมเดลมีประสิทธิภาพในการทำนายที่ไม่ดีและอาจจำเป็นต้องปรับปรุงหรือพัฒนาโมเดลให้ดีขึ้น

GRU

ความแม่นยำบนชุดข้อมูลการฝึกการเรียนรู้ (Training Result) คือ 0.52 (ประมาณ 52%) แสดงว่าโมเดลยังมีความผิดพลาดที่สูง

ความแม่นยำบนชุดข้อมูลการตรวจสอบ (Validation Result) คือ 0.57 (ประมาณ 57%) แสดงว่าโมเดลยังมีความผิดพลาดที่สูง

ความแม่นยำบนชุดข้อมูลการทดสอบ (Test Result) คือ 0.50 (ประมาณ 50%) แสดงว่าโมเดลมีความแม่นยำในการทำนายที่ต่ำกว่าทั้งชุดข้อมูลการฝึกการเรียนรู้และการตรวจสอบ และมีความผิดพลาดที่สูง

ข้อมูลชี้ให้เห็นว่าโมเดลที่ใช้ในงานนี้มีประสิทธิภาพในการทำนายที่ต่ำและอาจจะต้องมีการปรับปรุงหรือการพัฒนาเพื่อเพิ่มประสิทธิภาพในการจำแนกข้อมูลให้ดีขึ้น การปรับแต่งโมเดลหรือการสร้างคุณลักษณะอาจเป็นทางเลือกที่ดีในการพัฒนาโมเดลให้มีประสิทธิภาพมากขึ้น

สรุปทั้งหมด

- 2D CNN ดีที่สุดของทั้ง 4 โมเดลในการทำนายทั้งในกรณีภาวะซึมเศร้าและไม่มีภาวะซึมเศร้า
- 1D CNN มีผลลัพธ์ที่เป็นอันดับ 2 ของทั้ง 4 โมเดลทั้งในกรณีของภาวะซึมเศร้าและไม่มีภาวะซึมเศร้า
- LSTM และ GRU มีปัญหาในการทำนายในกรณีภาวะซึมเศร้า

ตารางที่ 4.7 ผลของ Training Result, Validation Result, Test Result แสดงค่า Precision, Recall, F1-Score และ Accuracy ของการจำแนกประเภทระหว่างมีภาวะซึมเศร้าและไม่มีภาวะซึมเศร้าของทั้ง 10 Model (KNN, MLP, Random Forest, Logistic Regression, SVM (Linear), SVM (RBF), GNB, Decision Tree, AdaBoost, SGD)

Architecture	Training Result					Validation Result					Test Result				
	precision	recall	f1-score	support	accuracy	precision	recall	f1-score	support	accuracy	precision	recall	f1-score	support	accuracy
SVM (RBF)															
depression	0.96	1.00	0.96	134	0.96	0.84	0.91	0.87	34	0.87	0.83	0.83	0.83	42	0.83
non-depression	1.00	0.91	0.95	134		0.90	0.82	0.86	34		0.83	0.83	0.83	42	
SVM (Linear)															
depression	1.00	1.00	1.00	134	1.00	0.82	0.94	0.88	34	0.87	0.88	0.90	0.89	42	0.89
non-depression	1.00	1.00	1.00	134		0.93	0.79	0.86	34		0.90	0.88	0.89	42	
Random Forest															
depression	1.00	1.00	1.00	134	1.00	0.91	0.91	0.91	34	0.91	0.95	0.95	0.95	42	0.95
non-depression	1.00	1.00	1.00	134		0.91	0.91	0.91	34		0.95	0.95	0.95	42	
GNB															
depression	0.77	0.82	0.79	134	0.79	0.69	0.65	0.67	34	0.68	0.76	0.83	0.80	42	0.79
non-depression	0.81	0.75	0.78	134		0.67	0.71	0.69	34		0.82	0.74	0.78	42	
Logistic Regression															
depression	1.00	1.00	1.00	134	1.00	0.86	0.88	0.87	34	0.87	0.82	0.88	0.85	42	0.85
non-depression	1.00	1.00	1.00	134		0.88	0.85	0.87	34		0.87	0.81	0.84	42	
Decision Tree															
depression	1.00	1.00	1.00	134	1.00	0.79	0.76	0.78	34	0.78	0.79	0.74	0.77	42	0.77
non-depression	1.00	1.00	1.00	134		0.77	0.79	0.78	34		0.76	0.81	0.78	42	

AdaBoost															
depression	1.00	1.00	1.00	134	1.00	0.91	0.88	0.90	34	0.90	0.81	0.83	0.82	42	0.82
non-depression	1.00	1.00	1.00	134		0.89	0.91	0.90	34		0.83	0.81	0.82	42	
SGD															
depression	0.83	0.75	0.79	134	0.80	0.77	0.59	0.67	34	0.71	0.81	0.83	0.82	42	0.82
non-depression	0.77	0.84	0.81	134		0.67	0.82	0.74	34		0.83	0.81	0.82	42	
MLP															
depression	1.00	1.00	1.00	134	1.00	0.89	0.91	0.90	34	0.90	0.95	0.95	0.95	42	0.95
non-depression	1.00	1.00	1.00	134		0.91	0.88	0.90	34		0.95	0.95	0.95	42	
KNN															
depression	1.00	1.00	1.00	134	1.00	0.91	0.94	0.93	34	0.93	0.95	0.95	0.95	42	0.95
non-depression	1.00	1.00	1.00	134		0.94	0.91	0.93	34		0.95	0.95	0.95	42	

จากตารางสรุปได้ดังนี้

จากผลการทดสอบทั้งหมด โมเดลที่มีประสิทธิภาพที่ดีที่สุดในการจำแนกคลาส Depression และ Non-depression ในชุดข้อมูลทดสอบคือ KNN (K Nearest Neighbour), MLP (Multi-Layer Perceptron) และ Random Forest ซึ่งมีความแม่นยำประมาณ 95% และเหมาะสมสำหรับงานนี้อย่างมีนัยสำคัญ SVM (Linear) ให้ผลลัพธ์ที่ดีในการทดสอบใกล้เคียง 90% SVM (RBF), Logistic Regression, AdaBoost, SGD (Stochastic Gradient Descent) มีค่าความแม่นยำมากกว่า 80% Decision Tree และ Gaussian Naive Bayes โดยรวมให้ผลลัพธ์ที่ต่ำกว่าโมเดลอื่นในทุกขั้นตอน

ตารางที่ 4.8 เปรียบเทียบผลรวมค่าความแม่นยำของประเภทที่ 1 การจำแนกประเภทระหว่างมีภาวะซึมเศร้าและไม่มีภาวะซึมเศร้าระหว่างแบบจำลองการเรียนรู้เชิงลึก และการเรียนรู้ของเครื่อง

แบบจำลองการเรียนรู้เชิงลึก		แบบจำลองการเรียนรู้ของเครื่อง	
Model	Accuracy	Model	Accuracy
1D CNN	70%	SVM (RBF)	83%
2D CNN	95%	SVM (Linear)	89%
LSTM	49%	Random Forest	95%
GRU	50%	Gaussian Naive Bayes	79%
		Logistic Regression	85%
		Decision Tree	77%
		AdaBoost	82%
		SGD (Stochastic Gradient Descent)	82%
		MLP (Multi-Layer Perceptron)	95%
		KNN (K Nearest Neighbour)	95%

สรุปผลได้ดังนี้

2D CNN มีความแม่นยำสูงสุดที่ประมาณ 95% ซึ่งแสดงให้เห็นว่าแบบจำลองนี้มีประสิทธิภาพที่ดีในการทำนายข้อมูลเมื่อถูกนำเข้าด้วยข้อมูลที่มีลักษณะเชิงพื้นที่ (Spatial Data) อย่างเช่น รูปภาพ 2 มิติ เป็นต้น

1D CNN มีความแม่นยำประมาณ 70% ซึ่งถือว่าความแม่นยำนี้สูงพอสมควรและเหมาะสมสำหรับข้อมูลที่มีลักษณะเชิงชั้นเวลา (Time Series Data)

LSTM และ GRU มีความแม่นยำที่น้อยกว่า 50% ซึ่งไม่สูงมาก แต่ยังคงมีความสามารถในการจัดการกับข้อมูลที่มีลักษณะเชิงชั้นเวลาได้

Random Forest, MLP (Multi-Layer Perceptron), และ KNN (K Nearest Neighbor) มีความแม่นยำสูงสุดที่ประมาณ 95% ซึ่งแสดงให้เห็นว่าแบบจำลองเหล่านี้มีประสิทธิภาพที่ดีในการทำนายข้อมูล

SVM (Linear) มีความแม่นยำประมาณ 89% และอยู่ในระดับสูง

Logistic Regression มีความแม่นยำประมาณ 85% ซึ่งก็ยังคงมีประสิทธิภาพดีในการทำนาย

SVM (RBF), AdaBoost, และ SGD มีความแม่นยำประมาณ 82-83%

Gaussian Naive Bayes มีความแม่นยำประมาณ 79% และ Decision Tree มีความแม่นยำประมาณ 77%

เมื่อเปรียบเทียบระหว่างการเรียนรู้เชิงลึกกับการเรียนรู้เรื่องเครื่องโมเดลที่มีค่าความแม่นยำสูงสุดได้เปอร์เซ็นต์ที่เท่ากันคือ 95% คือ 2D CNN, KNN ,MLP , Random Forest

4.2.1.2 Confusion Matrix และ Training loss and test loss graph ของแบบจำลองการเรียนรู้เชิงลึกและการเรียนรู้ของเครื่อง

1D CNN

จากข้อมูล Confusion Matrix ของ Test Result มีการวิเคราะห์เพื่อให้เห็นภาพรวมของผลการทดสอบโดยแบ่งเป็นกลุ่ม True Positive (เป็นกลุ่มที่ตรงกับเงื่อนไขที่ต้องการ) และ True Negative (เป็นกลุ่มที่ไม่ตรงกับเงื่อนไขที่ต้องการ)

True Positive: มีจำนวนผู้ทดสอบที่โมเดลจำแนกเป็น Depression อยู่ 17 กรณี

True Negative: มีจำนวนผู้ทดสอบที่โมเดลจำแนกไม่เป็น Depression อยู่ 42 กรณี

นอกจากนี้ยังมีผู้ทดสอบที่ถูกจำแนกผิดพลาดซึ่งสามารถแบ่งเป็น 2 กรณีดังนี้

False Positive: มีจำนวนผู้ทดสอบที่ถูกจำแนกว่าเป็น Depression แต่จริงๆ แล้วไม่เป็น Depression อยู่ 0 กรณี

False Negative: มีจำนวนผู้ทดสอบที่ถูกจำแนกว่าไม่เป็น Depression แต่จริงๆ แล้วเป็น Depression อยู่ 25 กรณี

โดยในภาพ % สีเขียว คือโมเดลทำนายถูกต้องส่วน % สีแดง คือโมเดลทำนายผิดพลาดดังภาพที่ 4.1

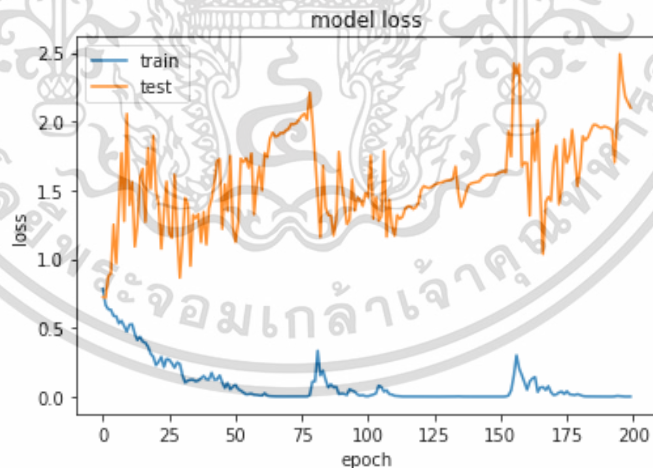
PHQ-9 (1-CNN)			
TARGET \ OUTPUT	depression	non-depression	SUM
depression	17 40.48% 59.52%	25 37.31% 62.69%	42 70.24% 29.76%
non-depression	0 0.00%	42 62.69% 37.31%	42 100.00% 0.00%
SUM	17 100.00% 0.00%	67 62.69% 37.31%	59 / 84 70.24% 29.76%

ภาพที่ 4.1 Confusion Matrix ของ Test Result (1D CNN)

ตารางที่ 4.9 Sensitivity และ Specificity (1D CNN)

Sensitivity (ความไวต่อการตรวจพบ)	Specificity (ความไวต่อการตรวจปฏิเสธ)
Sensitivity หมายถึง ความสามารถในการทดสอบในการตรวจพบผู้ป่วยที่มีโรค (ในที่นี้คือ Depression) คำนวณได้โดยใช้สูตร	Specificity หมายถึง ความสามารถในการทดสอบในการระบุผู้ป่วยที่ไม่มีโรค (Non-Depression) คำนวณได้โดยใช้สูตร
Sensitivity = (จำนวน True Positive ที่ตรวจพบ) / (จำนวนทั้งหมดที่มีโรค)	Specificity = (จำนวน True Negative ที่ระบุ) / (จำนวนทั้งหมดที่ไม่มีโรค)
Sensitivity = $17 / (17 + 0) = 17 / 17 \approx 1.0$	Specificity = $42 / (25 + 42) = 42 / 67 \approx 0.6269$
ดังนั้นค่า Sensitivity ในกรณีนี้คือ 1.0 หรือ 100% ซึ่งหมายความว่า การทดสอบมีความสามารถในการตรวจพบผู้ป่วยที่มีโรค Depression ทุกคนที่ถูกตรวจ	ดังนั้นค่า Specificity ในกรณีนี้คือ 0.6269 หรือประมาณ 62.69% ซึ่งหมายความว่า การทดสอบมีความสามารถในการระบุผู้ป่วยที่ไม่มีโรค Depression ประมาณ 62.69% ของกรณีที่ทดสอบนี้ใช้งาน

Training loss and test loss graph (1D CNN)



ภาพที่ 4.2 Training and test accuracy and loss (1D CNN)

2D CNN

จากข้อมูล Confusion Matrix ของ Test Result มีการวิเคราะห์เพื่อให้เห็นภาพรวมของผลการทดสอบโดยแบ่งเป็นกลุ่ม True Positive (เป็นกลุ่มที่ตรงกับเงื่อนไขที่ต้องการ) และ True Negative (เป็นกลุ่มที่ไม่ตรงกับเงื่อนไขที่ต้องการ)

True Positive: มีจำนวนผู้ทดสอบที่โมเดลจำแนกเป็น Depression อยู่ 40 กรณี

True Negative: มีจำนวนผู้ทดสอบที่โมเดลจำแนกไม่เป็น Depression อยู่ 40 กรณี

นอกจากนี้ยังมีผู้ทดสอบที่ถูกจำแนกผิดพลาดซึ่งสามารถแบ่งเป็น 2 กรณีดังนี้

False Positive: มีจำนวนผู้ทดสอบที่ถูกจำแนกว่าเป็น Depression แต่จริงๆ แล้วไม่เป็น Depression อยู่ 2 กรณี

False Negative: มีจำนวนผู้ทดสอบที่ถูกจำแนกว่าไม่เป็น Depression แต่จริงๆ แล้วเป็น Depression อยู่ 2 กรณี

โดยในภาพ % สีเขียว คือโมเดลทำนายถูกต้องส่วน % สีแดง คือโมเดลทำนายผิดพลาดดัง

ภาพที่ 4.3

PHQ-9 (2-CNN)			
TARGET \ OUTPUT	depression	non-depression	SUM
depression	40	2	42 95.24% 4.76%
non-depression	2	40	42 95.24% 4.76%
SUM	42 95.24% 4.76%	42 95.24% 4.76%	80 / 84 95.24% 4.76%

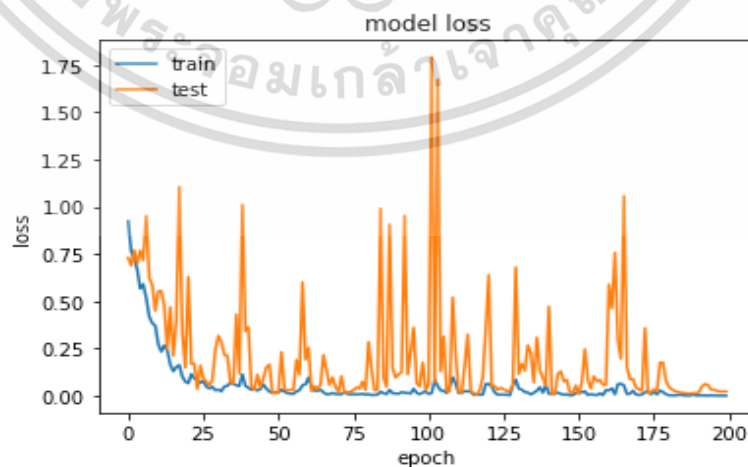
ภาพที่ 4.3 Confusion Matrix ของ Test Result (2D CNN)

ตารางที่ 4.10 Sensitivity และ Specificity (2D CNN)

Sensitivity (ความไวต่อการตรวจพบ)	Specificity (ความไวต่อการตรวจปฏิเสธ)
Sensitivity หมายถึง ความสามารถในการทดสอบในการตรวจพบผู้ป่วยที่มีโรค (ในที่นี้คือ Depression) คำนวณได้โดยใช้สูตร	Specificity หมายถึง ความสามารถในการทดสอบในการระบุผู้ป่วยที่ไม่มีโรค (Non-Depression) คำนวณได้โดยใช้สูตร
Sensitivity = (จำนวน True Positive ที่ตรวจพบ) / (จำนวนทั้งหมดที่มีโรค)	Specificity = (จำนวน True Negative ที่ระบุ) / (จำนวนทั้งหมดที่ไม่มีโรค)
Sensitivity = $40 / (40 + 2) = 40 / 42 \approx 0.9524$	Specificity = $40 / (2 + 40) = 40 / 42 \approx 0.9524$
ดังนั้นค่า Sensitivity ในกรณีนี้คือ 0.9524 หรือประมาณ 95.24% ซึ่งหมายความว่า การทดสอบมีความสามารถในการตรวจพบผู้ป่วยที่มีโรค depression ประมาณ 95.24% ของกรณีที่ทดสอบนี้ใช้งาน	ดังนั้นค่า Specificity ในกรณีนี้คือ 0.9524 หรือประมาณ 95.24% ซึ่งหมายความว่า การทดสอบมีความสามารถในการระบุผู้ป่วยที่ไม่มีโรค depression ประมาณ 95.24% ของกรณีที่ทดสอบนี้ใช้งาน

ดังนั้นค่า Sensitivity และ Specificity ในกรณีนี้มีค่าเท่ากันคือ 0.9524 หรือประมาณ 95.24% ซึ่งหมายความว่า การทดสอบมีความสามารถในการตรวจพบและระบุโรค Depression และ Non-Depression ในร้อยละ 95.24 ของกรณีที่ทดสอบ

Training loss and test loss graph (2D CNN)



ภาพที่ 4.4 Training and test accuracy and loss (2D CNN)

LSTM

จากข้อมูล Confusion Matrix ของ Test Result มีการวิเคราะห์เพื่อให้เห็นภาพรวมของผลการทดสอบโดยแบ่งเป็นกลุ่ม True Positive (เป็นกลุ่มที่ตรงกับเงื่อนไขที่ต้องการ) และ True Negative (เป็นกลุ่มที่ไม่ตรงกับเงื่อนไขที่ต้องการ)

True Positive: มีจำนวนผู้ทดสอบที่โมเดลจำแนกเป็น Depression อยู่ 1 กรณี

True Negative: มีจำนวนผู้ทดสอบที่โมเดลจำแนกว่าไม่เป็น Depression อยู่ 40 กรณี

นอกจากนี้ยังมีผู้ทดสอบที่ถูกจำแนกผิดพลาดซึ่งสามารถแบ่งเป็น 2 กรณีดังนี้

False Positive: มีจำนวนผู้ทดสอบที่ถูกจำแนกว่าเป็น Depression แต่จริงๆ แล้วไม่เป็น Depression อยู่ 2 กรณี

False Negative: มีจำนวนผู้ทดสอบที่ถูกจำแนกว่าไม่เป็น Depression แต่จริงๆ แล้วเป็น Depression อยู่ 41 กรณี

โดยในภาพ % สีเขียว คือโมเดลทำนายถูกต้องส่วน % สีแดง คือโมเดลทำนายผิดพลาดดัง

ภาพที่ 4.5

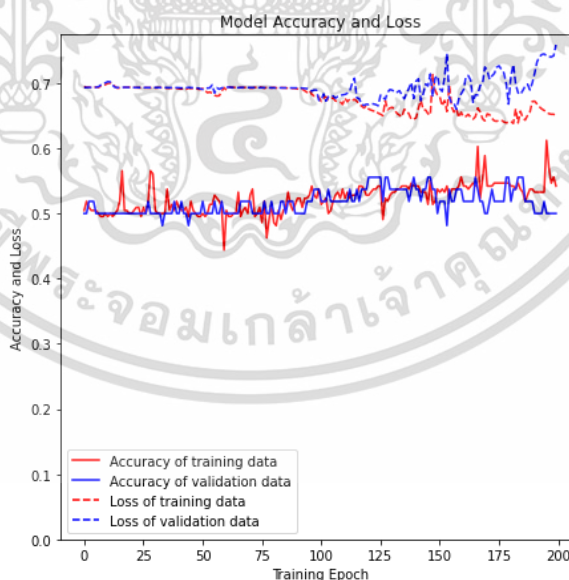
PHQ-9 (LSTM)			
TARGET \ OUTPUT	depression	non-depression	SUM
depression	1 2.38% 97.62%	41 49.38% 50.62%	42 48.81% 51.19%
non-depression	2 4.76%	40 49.38% 50.62%	42 48.81% 51.19%
SUM	3 33.33% 66.67%	81 49.38% 50.62%	41 / 84 48.81% 51.19%

ภาพที่ 4.5 Confusion Matrix ของ Test Result (LSTM)

ตารางที่ 4.11 Sensitivity และ Specificity (LSTM)

Sensitivity (ความไวต่อการตรวจพบ)	Specificity (ความไวต่อการตรวจปฏิเสธ)
Sensitivity หมายถึง ความสามารถของการทดสอบในการตรวจพบผู้ป่วยที่มีโรค (ในที่นี้คือ Depression) คำนวณได้โดยใช้สูตร	Specificity หมายถึง ความสามารถของการทดสอบในการระบุผู้ป่วยที่ไม่มีโรค (Non-Depression) คำนวณได้โดยใช้สูตร
Sensitivity = (จำนวน True Positive ที่ตรวจพบ) / (จำนวนทั้งหมดที่มีโรค)	Specificity = (จำนวน True Negative ที่ระบุ) / (จำนวนทั้งหมดที่ไม่มีโรค)
Sensitivity = $1 / (1 + 2) = 1 / 3 \approx 0.3333$	Specificity = $40 / (41 + 40) = 40 / 81 \approx 0.4938$
ดังนั้นค่า Sensitivity ในกรณีนี้คือ 0.3333 หรือประมาณ 33.33% ซึ่งหมายความว่าความสามารถในการตรวจพบผู้ป่วยที่มีโรค Depression เพียง 33.33% ของกรณีทดสอบนี้ ใช้งาน	ดังนั้นค่า Specificity ในกรณีนี้คือ 0.4938 หรือประมาณ 49.38% ซึ่งหมายความว่าความสามารถในการระบุผู้ป่วยที่ไม่มีโรค depression เพียง 49.38% ของกรณีทดสอบนี้ ใช้งาน

Training loss and test loss graph (LSTM)



ภาพที่ 4.6 Training and test accuracy and loss (LSTM)

GRU

จากข้อมูล Confusion Matrix ของ Test Result มีการวิเคราะห์เพื่อให้เห็นภาพรวมของผลการทดสอบโดยแบ่งเป็นกลุ่ม True Positive (เป็นกลุ่มที่ตรงกับเงื่อนไขที่ต้องการ) และ True Negative (เป็นกลุ่มที่ไม่ตรงกับเงื่อนไขที่ต้องการ)

True Positive: มีจำนวนผู้ทดสอบที่โมเดลจำแนกเป็น Depression อยู่ 2 กรณี

True Negative: มีจำนวนผู้ทดสอบที่โมเดลจำแนกว่าไม่เป็น Depression อยู่ 40 กรณี

นอกจากนี้ยังมีผู้ทดสอบที่ถูกจำแนกผิดพลาดซึ่งสามารถแบ่งเป็น 2 กรณีดังนี้

False Positive: มีจำนวนผู้ทดสอบที่ถูกจำแนกว่าเป็น Depression แต่จริงๆ แล้วไม่เป็น Depression อยู่ 2 กรณี

False Negative: มีจำนวนผู้ทดสอบที่ถูกจำแนกว่าไม่เป็น Depression แต่จริงๆ แล้วเป็น Depression อยู่ 40 กรณี

โดยในภาพ % สีเขียว คือโมเดลทำนายถูกต้องส่วน % สีแดง คือโมเดลทำนายผิดพลาดดังภาพที่ 4.7

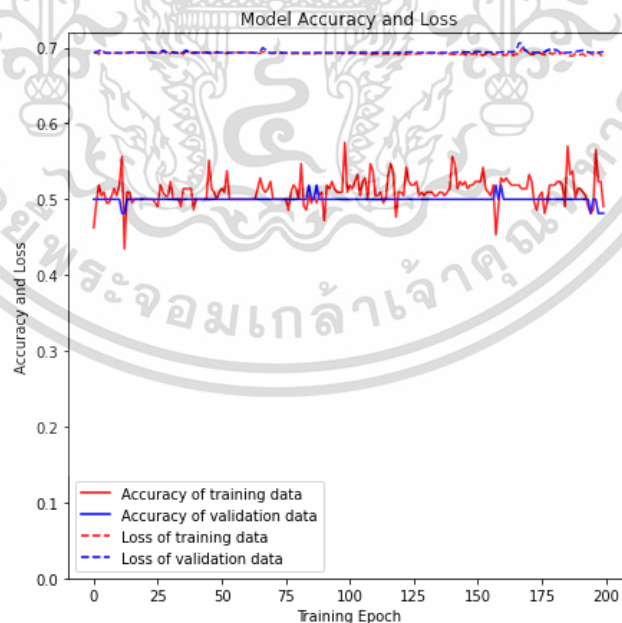
PHQ-9 (GRU)			
TARGET \ OUTPUT	depression	non-depression	SUM
depression	2 4.76%	40 95.24%	42
non-depression	2 95.24%	40 4.76%	42
SUM	4 50.00%	80 50.00%	42 / 84 50.00%

ภาพที่ 4.7 Confusion Matrix ของ Test Result (GRU)

ตารางที่ 4.12 Sensitivity และ Specificity (GRU)

Sensitivity (ความไวต่อการตรวจพบ)	Specificity (ความไวต่อการตรวจปฏิเสธ)
Sensitivity หมายถึง ความสามารถของการทดสอบในการตรวจพบผู้ป่วยที่มีโรค (ในที่นี้คือ Depression) คำนวณได้โดยใช้สูตร	Specificity หมายถึง ความสามารถของการทดสอบในการระบุผู้ป่วยที่ไม่มีโรค (Non-Depression) คำนวณได้โดยใช้สูตร
Sensitivity = (จำนวน True Positive ที่ตรวจพบ) / (จำนวนทั้งหมดที่มีโรค)	Specificity = (จำนวน True Negative ที่ระบุ) / (จำนวนทั้งหมดที่ไม่มีโรค)
Sensitivity = $2 / (2 + 2) = 2 / 4 \approx 0.50$	Specificity = $40 / (40 + 40) = 40 / 80 \approx 0.50$
ดังนั้นค่า Sensitivity ในกรณีนี้คือ 0.50 หรือประมาณ 50% ซึ่งหมายความว่าความสามารถในการตรวจพบผู้ป่วยที่มีโรค Depression เพียง 50% ของกรณีทดสอบนี้ใช้งาน	ดังนั้นค่า Specificity ในกรณีนี้คือ 0.50 หรือประมาณ 50% ซึ่งหมายความว่าความสามารถในการระบุผู้ป่วยที่ไม่มีโรค Depression ถึง 95.24% ของกรณีทดสอบนี้ใช้งาน

Training loss and test loss graph (GRU)



ภาพที่ 4.8 Training and test accuracy and loss (GRU)

KNN (K Nearest Neighbour)

จากข้อมูล Confusion Matrix ของ Test Result มีการวิเคราะห์เพื่อให้เห็นภาพรวมของผลการทดสอบดังต่อไปนี้

สำหรับคลาส Depression โมเดลทำนายถูกต้องใน 40 กรณี และทำนายผิด 2 กรณี

สำหรับคลาส Non-Depression โมเดลทำนายถูกต้องใน 40 กรณี และทำนายผิด 2 กรณี

โมเดลมีประสิทธิภาพดีในการจำแนกทั้งคลาส "Depression" และ "Non-depression" โดยมีจำนวน False Positive และ False Negative ที่น้อยมาก แสดงถึงความแม่นยำและประสิทธิภาพในการทำนายของโมเดลในการจำแนกกลุ่มข้อมูลนี้

โดยในภาพ % สีเขียว คือโมเดลทำนายถูกต้องส่วน % สีแดง คือโมเดลทำนายผิดพลาดดังภาพที่ 4.9

KNN (K Nearest Neighbour)			
TARGET \ OUTPUT	Depression	Non-Depression	SUM
Depression	40 95.24% 4.76%	2 4.76%	42 95.24% 4.76%
Non-Depression	2 4.76%	40 95.24% 4.76%	42 95.24% 4.76%
SUM	42 95.24% 4.76%	42 95.24% 4.76%	80 / 84 95.24% 4.76%

ภาพที่ 4.9 Confusion Matrix ของ Test Result KNN (K Nearest Neighbour) Model

ตารางที่ 4.13 Sensitivity และ Specificity KNN (K Nearest Neighbour)

Sensitivity (ความไวต่อการตรวจพบ)	Specificity (ความไวต่อการตรวจพบปฏิเสธ)
Sensitivity (True Positive Rate) หรือ ความสามารถในการจำแนกคลาส "Depression"	Specificity (True Negative Rate) หรือ ความสามารถในการจำแนกคลาส "Non-depression"

Sensitivity = $TP / (TP + FN) = 40 / (40 + 2) = 40 / 42 \approx 0.9524$	Specificity = $TN / (TN + FP) = 40 / (40 + 2) = 40 / 42 \approx 0.9524$
Sensitivity สำหรับคลาส "Depression" ประมาณ 95.24%	Specificity สำหรับคลาส "Non-Depression" ประมาณ 95.24%

นี่คือผลลัพธ์ที่แสดงให้เห็นว่าโมเดลมีความสามารถในการจำแนกทั้งคลาส Depression และ Non-depression อย่างดี เนื่องจาก Sensitivity และ Specificity มีค่าสูงและใกล้เคียงกัน

MLP (Multi-Layer Perceptron)

จากข้อมูล Confusion Matrix ของ Test Result มีการวิเคราะห์เพื่อให้เห็นภาพรวมของผลการทดสอบดังต่อไปนี้

สำหรับคลาส Depression โมเดลทำนายถูกต้องใน 40 กรณี และทำนายผิด 2 กรณี

สำหรับคลาส Non-Depression โมเดลทำนายถูกต้องใน 40 กรณี และทำนายผิด 2 กรณี

โมเดลมีประสิทธิภาพดีในการจำแนกทั้งคลาส Depression และ Non-depression โดยมีจำนวน False Positive และ False Negative ที่น้อยมาก แสดงถึงความแม่นยำและประสิทธิภาพในการทำนายของโมเดลในการจำแนกกลุ่มข้อมูลนี้

โดยในภาพ % สีเขียว คือโมเดลทำนายถูกต้องส่วน % สีแดง คือโมเดลทำนายผิดพลาดดังภาพที่ 4.10

MLP (Multi-Layer Perceptron)			
TARGET \ OUTPUT	Depression	Non-Depression	SUM
Depression	40 95.24%	2 4.76%	42 95.24%
Non-Depression	2 4.76%	40 95.24%	42 95.24%
SUM	42 95.24%	42 95.24%	80 / 84 95.24%

ภาพที่ 4.10 Confusion Matrix ของ Test Result MLP (Multi-Layer Perceptron) Model

ตารางที่ 4.14 Sensitivity และ Specificity MLP (Multi-Layer Perceptron)

Sensitivity (ความไวต่อการตรวจพบ)	Specificity (ความไวต่อการตรวจปฏิเสธ)
Sensitivity (True Positive Rate) หรือ ความสามารถในการจำแนกคลาส "Depression" $Sensitivity = TP / (TP + FN) = 40 / (40 + 2) =$ $40 / 42 \approx 0.9524$	Specificity (True Negative Rate) หรือ ความสามารถในการจำแนกคลาส "Non-depression" $Specificity = TN / (TN + FP) = 40 / (40 + 2) =$ $40 / 42 \approx 0.9524$
Sensitivity สำหรับคลาส "Depression" ประมาณ 95.24%	Specificity สำหรับคลาส "Non-Depression" ประมาณ 95.24%

นี่คือผลลัพธ์ที่แสดงให้เห็นว่าโมเดลมีความสามารถในการจำแนกทั้งคลาส "Depression" และ "Non-Depression" อย่างดี เนื่องจาก Sensitivity และ Specificity มีค่าสูงและใกล้เคียงกัน

Random Forest

จากข้อมูล Confusion Matrix ของ Test Result มีการวิเคราะห์เพื่อให้เห็นภาพรวมของผลการทดสอบดังต่อไปนี้

สำหรับคลาส Depression โมเดลทำนายถูกต้องใน 40 กรณี และทำนายผิด 2 กรณี

สำหรับคลาส Non-Depression โมเดลทำนายถูกต้องใน 40 กรณี และทำนายผิด 2 กรณี

โมเดลมีประสิทธิภาพดีในการจำแนกทั้งคลาส Depression และ Non-depression โดยมีจำนวน False Positive และ False Negative ที่น้อยมาก แสดงถึงความแม่นยำและประสิทธิภาพในการทำนายของโมเดลในการจำแนกกลุ่มข้อมูลนี้

โดยในภาพ % สีเขียว คือโมเดลทำนายถูกต้องส่วน % สีแดง คือโมเดลทำนายผิดพลาดดังภาพที่ 4.11

Random Forest			
TARGET \ OUTPUT	Depression	Non-Depression	SUM
Depression	40	2	42 95.24% 4.76%
Non-Depression	2	40	42 95.24% 4.76%
SUM	42 95.24% 4.76%	42 95.24% 4.76%	80 / 84 95.24% 4.76%

ภาพที่ 4.11 Confusion Matrix ของ Test Result Random Forest Model

ตารางที่ 4.15 Sensitivity และ Specificity (Random Forest)

Sensitivity (ความไวต่อการตรวจพบ)	Specificity (ความไวต่อการตรวจปฏิเสธ)
Sensitivity (True Positive Rate) หรือ ความสามารถในการจำแนกคลาส "Depression" $\text{Sensitivity} = \text{TP} / (\text{TP} + \text{FN}) = 40 / (40 + 2) = 40 / 42 \approx 0.9524$	Specificity (True Negative Rate) หรือ ความสามารถในการจำแนกคลาส "Non-depression" $\text{Specificity} = \text{TN} / (\text{TN} + \text{FP}) = 40 / (40 + 2) = 40 / 42 \approx 0.9524$
Sensitivity สำหรับคลาส "Depression" ประมาณ 95.24%	Specificity สำหรับคลาส "Non-Depression" ประมาณ 95.24%

นี่คือผลลัพธ์ที่แสดงให้เห็นว่าโมเดลมีความสามารถในการจำแนกทั้งคลาส "Depression" และ "Non-Depression" อย่างดี เนื่องจาก Sensitivity และ Specificity มีค่าสูงและใกล้เคียงกัน

Logistic Regression

จากข้อมูล Confusion Matrix ของ Test Result มีการวิเคราะห์เพื่อให้เห็นภาพรวมของผลการทดสอบดังต่อไปนี้

สำหรับคลาส Depression โมเดลทำนายถูกต้องใน 37 กรณี และทำนายผิด 5 กรณี

สำหรับคลาส Non-Depression โมเดลทำนายถูกต้องใน 34 กรณี และทำนายผิด 8 กรณี

โดยในภาพ % สีเขียว คือโมเดลทำนายถูกต้องส่วน % สีแดง คือโมเดลทำนายผิดพลาดดัง

ภาพที่ 4.12

Logistic Regression			
TARGET \ OUTPUT	Depression	Non-Depression	SUM
Depression	37 88.10% 11.90%	5	42
Non-Depression	8	34 80.95% 19.05%	42
SUM	45 82.22% 17.78%	39 87.18% 12.82%	71 / 84 84.52% 15.48%

ภาพที่ 4.12 Confusion Matrix ของ Test Result Logistic Regression Model

ตารางที่ 4.16 Sensitivity และ Specificity (Logistic Regression)

Sensitivity (ความไวต่อการตรวจพบ)	Specificity (ความไวต่อการตรวจปฏิเสธ)
Sensitivity (True Positive Rate) หรือ ความสามารถในการจำแนกคลาส "Depression"	Specificity (True Negative Rate) หรือ ความสามารถในการจำแนกคลาส "Non-depression"
$Sensitivity = TP / (TP + FN) = 37 / (37 + 5) = 37 / 42 \approx 0.881$	$Specificity = TN / (TN + FP) = 34 / (34 + 8) = 34 / 42 \approx 0.809$
Sensitivity สำหรับคลาส "Depression" ประมาณ 88.1%	Specificity สำหรับคลาส "Non-Depression" ประมาณ 80.9%

นี่คือผลลัพธ์ที่แสดงให้เห็นว่าโมเดลมีความสามารถในการจำแนกทั้งคลาส Depression และ Non-Depression อย่างดี เนื่องจาก Sensitivity และ Specificity มีค่าสูงและใกล้เคียงกัน

SVM (Linear)

จากข้อมูล Confusion Matrix ของ Test Result มีการวิเคราะห์เพื่อให้เห็นภาพรวมของผลการทดสอบดังต่อไปนี้

สำหรับคลาส Depression โมเดลทำนายถูกต้องใน 38 กรณี และทำนายผิด 4 กรณี

สำหรับคลาส Non-Depression โมเดลทำนายถูกต้องใน 37 กรณี และทำนายผิด 5 กรณี

โดยในภาพ % สีเขียว คือโมเดลทำนายถูกต้องส่วน % สีแดง คือโมเดลทำนายผิดพลาดดัง

ภาพที่ 4.13

		SVM (Linear)		
		TARGET Depression	Non-Depression	SUM
OUTPUT	Depression	38 90.48% 9.52%	4	42
	Non-Depression	5	37 88.10% 11.90%	42
SUM		43 88.37% 11.63%	41 90.24% 9.76%	75 / 84 89.29% 10.71%

ภาพที่ 4.13 Confusion Matrix ของ Test Result SVM (Linear) Model

ตารางที่ 4.17 Sensitivity และ Specificity (SVM (Linear))

Sensitivity (ความไวต่อการตรวจพบ)	Specificity (ความไวต่อการตรวจปฏิเสธ)
Sensitivity (True Positive Rate) หรือ ความสามารถในการจำแนกคลาส	Specificity (True Negative Rate) หรือ ความสามารถในการจำแนกคลาส

"Depression" Sensitivity = $TP / (TP + FN) = 38 / (38 + 4) = 38 / 42 \approx 0.904$	"Non-depression" Specificity = $TN / (TN + FP) = 37 / (37 + 5) = 37 / 42 \approx 0.881$
Sensitivity สำหรับคลาส "Depression" ประมาณ 90.4%	Specificity สำหรับคลาส "Non-Depression" ประมาณ 88.1%

นี่คือผลลัพธ์ที่แสดงให้เห็นว่าโมเดลมีความสามารถในการจำแนกทั้งคลาส Depression และ Non-Depression อย่างดี เนื่องจาก Sensitivity และ Specificity มีค่าสูงและใกล้เคียงกัน

SVM (RBF)

จากข้อมูล Confusion Matrix ของ Test Result มีการวิเคราะห์เพื่อให้เห็นภาพรวมของผลการทดสอบดังต่อไปนี้

สำหรับคลาส Depression โมเดลทำนายถูกต้องใน 35 กรณี และทำนายผิด 7 กรณี

สำหรับคลาส Non-Depression โมเดลทำนายถูกต้องใน 35 กรณี และทำนายผิด 7 กรณี

โดยในภาพ % สีเขียว คือโมเดลทำนายถูกต้องส่วน % สีแดง คือโมเดลทำนายผิดพลาดดัง

ภาพที่ 4.14

SVM - RBF			
TARGET \ OUTPUT	Depression	Non-Depression	SUM
Depression	35 83.33% 16.67%	7 16.67%	42 83.33% 16.67%
Non-Depression	7 16.67%	35 83.33%	42 83.33% 16.67%
SUM	42 83.33% 16.67%	42 83.33% 16.67%	70 / 84 83.33% 16.67%

ภาพที่ 4.14 Confusion Matrix ของ Test Result SVM (RBF) Model

ตารางที่ 4.18 Sensitivity และ Specificity (SVM (RBF))

Sensitivity (ความไวต่อการตรวจพบ)	Specificity (ความไวต่อการตรวจปฏิเสธ)
Sensitivity (True Positive Rate) หรือ ความสามารถในการจำแนกคลาส "Depression" $\text{Sensitivity} = \text{TP} / (\text{TP} + \text{FN}) = 35 / (35 + 7) = 35 / 42 \approx 0.833$	Specificity (True Negative Rate) หรือ ความสามารถในการจำแนกคลาส "Non-depression" $\text{TN} / (\text{TN} + \text{FP}) = 35 / (35 + 7) = 35 / 42 \approx 0.833$
Sensitivity สำหรับคลาส "Depression" ประมาณ 83.3%	Specificity สำหรับคลาส "Non-Depression" ประมาณ 83.3%

นี่คือผลลัพธ์ที่แสดงให้เห็นว่าโมเดลมีความสามารถในการจำแนกทั้งคลาส "Depression" และ "Non-Depression" อย่างดี เนื่องจาก Sensitivity และ Specificity มีค่าสูงและใกล้เคียงกัน

Gaussian Naive Bayes

จากข้อมูล Confusion Matrix ของ Test Result มีการวิเคราะห์เพื่อให้เห็นภาพรวมของผลการทดสอบดังต่อไปนี้

สำหรับคลาส Depression โมเดลทำนายถูกต้องใน 35 กรณี และทำนายผิด 7 กรณี

สำหรับคลาส Non-Depression โมเดลทำนายถูกต้องใน 31 กรณี และทำนายผิด 11 กรณี

โดยในภาพ % สีเขียว คือโมเดลทำนายถูกต้องส่วน % สีแดง คือโมเดลทำนายผิดพลาดดังภาพที่ 4.15

Gaussian Naive Bayes			
TARGET \ OUTPUT	Depression	Non-Depression	SUM
Depression	35	7	42 83.33% 16.67%
Non-Depression	11	31	42 73.81% 26.19%
SUM	46 76.09% 23.91%	38 81.58% 18.42%	66 / 84 78.57% 21.43%

ภาพที่ 4.15 Confusion Matrix ของ Test Result Gaussian Naive Bayes Model

ตารางที่ 4.19 Sensitivity และ Specificity (Gaussian Naive Bayes)

Sensitivity (ความไวต่อการตรวจพบ)	Specificity (ความไวต่อการตรวจปฏิเสธ)
Sensitivity (True Positive Rate) หรือ ความสามารถในการจำแนกคลาส "Depression" $Sensitivity = TP / (TP + FN) = 35 / (35 + 7) =$ $35 / 42 \approx 0.833$	Specificity (True Negative Rate) หรือ ความสามารถในการจำแนกคลาส "Non-depression" $Specificity = TN / (TN + FP) = 31 / (31 + 11) =$ $31 / 42 \approx 0.738$
Sensitivity สำหรับคลาส "Depression" ประมาณ 83.3%	Specificity สำหรับคลาส "Non-Depression" ประมาณ 73.8%

โมเดลมีความสามารถในการตรวจพบคลาส "Depression" ในระดับที่ดี โดยมีค่า Sensitivity ที่สูง และในการตรวจพบคลาส "Non-depression" ค่า Specificity ที่ค่อนข้างสูง แต่ยังคงมีความผิดพลาดในการจำแนกคลาส "Non-depression" บางส่วนของข้อมูลที่มีค่า FN และ FP สูงๆ อาจต้องพิจารณาความสำคัญของการตรวจพบแต่ละคลาสในงานหรือแอปพลิเคชันให้ดีขึ้นต่อไป

Decision Tree

จากข้อมูล Confusion Matrix ของ Test Result มีการวิเคราะห์เพื่อให้เห็นภาพรวมของผลการทดสอบดังต่อไปนี้

สำหรับคลาส Depression โมเดลทำนายถูกต้องใน 31 กรณี และทำนายผิด 11 กรณี

สำหรับคลาส Non-Depression โมเดลทำนายถูกต้องใน 34 กรณี และทำนายผิด 8 กรณี

โดยในภาพ % สีเขียว คือโมเดลทำนายถูกต้องส่วน % สีแดง คือโมเดลทำนายผิดพลาดดัง

ภาพที่ 4.16

Decision Tree			
TARGET \ OUTPUT	Depression	Non-Depression	SUM
Depression	31 73.81% 26.19%	11 26.19%	42 73.81% 26.19%
Non-Depression	8 19.05%	34 80.95%	42 80.95% 19.05%
SUM	39 79.49% 20.51%	45 75.56% 24.44%	65 / 84 77.38% 22.62%

ภาพที่ 4.16 Confusion Matrix ของ Test Result Decision Tree Model

ตารางที่ 4.20 Sensitivity และ Specificity (Decision Tree)

Sensitivity (ความไวต่อการตรวจพบ)	Specificity (ความไวต่อการตรวจปฏิเสธ)
Sensitivity (True Positive Rate) หรือ ความสามารถในการจำแนกคลาส "Depression"	Specificity (True Negative Rate) หรือ ความสามารถในการจำแนกคลาส "Non-depression"
$Sensitivity = TP / (TP + FN) = 31 / (31 + 11)$ $= 31 / 42 \approx 0.738$	$Specificity = TN / (TN + FP) = 34 / (34 + 8) =$ $34 / 42 \approx 0.810$

Sensitivity สำหรับคลาส "Depression" ประมาณ 73.8%	Specificity สำหรับคลาส "Non-Depression" ประมาณ 81.0%
---	---

โมเดลมีความสามารถในการตรวจพบคลาส "Depression" ในระดับที่ดี โดยมีค่า Sensitivity ที่ค่อนข้างสูง และในการตรวจพบคลาส "Non-depression" ค่า Specificity ที่ค่อนข้างสูง แสดงให้เห็นว่าโมเดลมีความสามารถในการจำแนกคลาสทั้งสองอย่างดี

AdaBoost

จากข้อมูล Confusion Matrix ของ Test Result มีการวิเคราะห์เพื่อให้เห็นภาพรวมของผลการทดสอบดังต่อไปนี้

สำหรับคลาส Depression โมเดลทำนายถูกต้องใน 35 กรณี และทำนายผิด 7 กรณี

สำหรับคลาส Non-Depression โมเดลทำนายถูกต้องใน 34 กรณี และทำนายผิด 8 กรณี

โดยในภาพ % สีเขียว คือโมเดลทำนายถูกต้องส่วน % สีแดง คือโมเดลทำนายผิดพลาดดัง

ภาพที่ 4.17

AdaBoost			
TARGET \ OUTPUT	Depression	Non-Depression	SUM
Depression	35 83.33% 16.67%	7	42
Non-Depression	8	34 80.95% 19.05%	42
SUM	43 81.40% 18.60%	41 82.93% 17.07%	69 / 84 82.14% 17.86%

ภาพที่ 4.17 Confusion Matrix ของ Test Result AdaBoost Model

ตารางที่ 4.21 Sensitivity และ Specificity (AdaBoost)

Sensitivity (ความไวต่อการตรวจพบ)	Specificity (ความไวต่อการตรวจปฏิเสธ)
Sensitivity (True Positive Rate) หรือ ความสามารถในการจำแนกคลาส "Depression" $\text{Sensitivity} = \text{TP} / (\text{TP} + \text{FN}) = 35 / (35 + 7) = 35 / 42 \approx 0.833$	Specificity (True Negative Rate) หรือ ความสามารถในการจำแนกคลาส "Non-depression" $\text{Specificity} = \text{TN} / (\text{TN} + \text{FP}) = 34 / (34 + 8) = 34 / 42 \approx 0.810$
Sensitivity สำหรับคลาส "Depression" ประมาณ 83.3%	Specificity สำหรับคลาส "Non-Depression" ประมาณ 81.0%

โมเดลมีความสามารถในการตรวจพบคลาส "Depression" ในระดับที่ดี โดยมีค่า Sensitivity ที่ค่อนข้างสูง และในการตรวจพบคลาส "Non-depression" ค่า Specificity ที่ค่อนข้างสูง แสดงให้เห็นว่าโมเดลมีความสามารถในการจำแนกคลาสทั้งสองอย่างดี

Stochastic Gradient Descent (SGD)

จากข้อมูล Confusion Matrix ของ Test Result มีการวิเคราะห์เพื่อให้เห็นภาพรวมของผลการทดสอบดังต่อไปนี้

สำหรับคลาส Depression โมเดลทำนายถูกต้องใน 0 กรณี และทำนายผิด 42 กรณี

สำหรับคลาส Non-Depression โมเดลทำนายถูกต้องใน 42 กรณี และทำนายผิด 0 กรณี

โดยในภาพ % สีเขียว คือโมเดลทำนายถูกต้องส่วน % สีแดง คือโมเดลทำนายผิดพลาดดัง

ภาพที่ 4.18

Stochastic Gradient Descent (SGD)			
TARGET \ OUTPUT	Depression	Non-Depression	SUM
Depression	0	42	42 0.00% 100.00%
Non-Depression	0	42	42 100.00% 0.00%
SUM	0 NaN% NaN%	84 50.00% 50.00%	42 / 84 50.00% 50.00%

ภาพที่ 4.18 Confusion Matrix ของ Test Result Stochastic Gradient Descent (SGD) Model

ตารางที่ 4.22 Sensitivity และ Specificity (SGD)

Sensitivity (ความไวต่อการตรวจพบ)	Specificity (ความไวต่อการตรวจปฏิเสธ)
<p>Sensitivity (True Positive Rate) หรือความสามารถในการจำแนกคลาส "Depression"</p> <p>จะเป็น 0% หรือ 0/0 (ไม่มีค่า TP หรือ FN ในกรณีนี้) และไม่สามารถคำนวณได้ด้วยวิธีการปกติ (ที่ใช้ TN และ FN) แต่ในทางปฏิบัติ Sensitivity ในกรณีนี้มักจะถูกตั้งเป็น 0% หรือไม่มีการตรวจพบคลาส "Depression" ในทุกกรณี</p>	<p>Specificity (True Negative Rate) หรือความสามารถในการจำแนกคลาส "Non-depression"</p> <p>$Specificity = TN / (TN + FP) = 42 / (42 + 0) = 42 / 42 \approx 1.000$</p>
Sensitivity สำหรับคลาส "Depression" ประมาณ 0%	Specificity สำหรับคลาส "Non-Depression" ประมาณ 100%

Sensitivity สำหรับคลาส "Depression" ในกรณีนี้ คือ 0% (ไม่มีการตรวจพบคลาส "Depression" ในข้อมูล)

Specificity สำหรับคลาส "Non-depression" คือ 100% (ความถูกต้องในการตรวจพบคลาส "Non-depression" ในทุกกรณี)

โมเดลมีความแม่นยำในการตรวจจับคลาส "Non-depression" แต่ไม่สามารถตรวจจับคลาส "Depression" ได้ในข้อมูลที่ใช้ในการทดสอบนี้และมีค่า Specificity สูง

4.2.2 ประเภทที่ 2 การจำแนกประเภทที่แบ่งออกเป็น 5 ประเภทตามเกณฑ์ของ PHQ-9 ได้แก่ 1. ไม่มีอาการ 2. น้อย 3. ปานกลาง 4. มาก 5. รุนแรง

โดยผู้วิจัยไม่สามารถ Balance ข้อมูลได้เนื่องจากข้อมูลของแต่ละประเภททั้งหมด 5 กลุ่มยังมีน้อยเกินไปทางผู้วิจัยจึงตัดสินใจไม่ Balance เพื่อให้ข้อมูลที่เก็บมาได้ใช้ประโยชน์สูงสุด

ตารางที่ 4.23 ตารางแสดงจำนวนการเก็บข้อมูลอาสาสมัครประเภทที่ 2 แบบไม่ Balance ข้อมูล

ประเภท	อาสาสมัคร	จำนวนไฟล์
ไม่มีอาการ	17	145
น้อย	29	254
ปานกลาง	30	280
มาก	15	138
รุนแรง	9	74

- โดยผู้วิจัยได้แบ่งข้อมูลออกเป็น 1. Training set = 60% 2. Validation set = 20% 3. Unseen Test set = 20% ดังตารางที่ 4.27

ตารางที่ 4.24 ตารางแสดงจำนวนการแบ่งข้อมูล Training set = 60% 2. Validation set = 20% 3. Unseen Test set = 20%

ประเภท (ไฟล์)	Training set	Validation set	Unseen test set
ไม่มีอาการ	97	20	28
น้อย	159	48	47
ปานกลาง	169	47	64
มาก	89	19	30
รุนแรง	54	9	11

4.2.2.1 ผลการทดลองแบบจำลองการเรียนรู้เชิงลึกและการเรียนรู้ของเครื่องแบ่งข้อมูลเป็น 5 ประเภท

ตารางแสดงค่า Precision, Recall, F1-Score และ Accuracy ของการจำแนกประเภทที่แบ่งออกเป็น 5 ประเภทตามเกณฑ์ของ PHQ-9 ได้แก่ 1. ไม่มีอาการ 2. น้อย 3. ปานกลาง 4. มาก 5. รุนแรง

ตารางที่ 4.25 ผลของ Training Result, Validation Result, Test Result แสดงค่า Precision, Recall, F1-Score และ Accuracy ของการจำแนกประเภทที่แบ่งออกเป็น 5 ประเภทตามเกณฑ์ของ PHQ-9 ได้แก่ 1. ไม่มีอาการ 2. น้อย 3. ปานกลาง 4. มาก 5. รุนแรงของทั้ง 4 Model การเรียนรู้เชิงลึก

Architecture	Training Result					Validation Result					Test Result				
	precision	recall	f1-score	support	accuracy	precision	recall	f1-score	support	accuracy	precision	recall	f1-score	support	accuracy
1D CNN															
None	0.98	0.97	0.97	97	0.99	0.43	0.45	0.44	20	0.44	0.20	0.96	0.33	28	0.25
Mild	0.97	0.98	0.98	159		0.48	0.42	0.44	48		0.41	0.28	0.33	47	
Moderate	1.00	1.00	1.00	169		0.41	0.49	0.45	47		0.50	0.03	0.06	64	
Moderately Severe	0.98	0.97	0.97	89		0.50	0.47	0.49	19		0.30	0.10	0.15	30	
Severe	1.00	0.96	0.98	54		0.33	0.22	0.27	9		0.00	0.00	0.00	11	
2D CNN															
None	1.00	1.00	1.00	97	1.00	0.66	0.95	0.78	20	0.80	0.74	0.82	0.78	28	0.85
Mild	1.00	1.00	1.00	159		0.91	0.65	0.76	48		0.93	0.79	0.85	47	
Moderate	0.99	1.00	0.99	169		0.76	0.89	0.82	47		0.82	0.91	0.86	64	
Moderately Severe	1.00	1.00	1.00	89		0.83	0.79	0.81	19		0.89	0.83	0.86	30	
Severe	1.00	0.96	0.98	54		1.00	0.78	0.88	9		1.00	0.91	0.95	11	

LSTM															
None	0.75	0.37	0.50	97	0.45	0.53	0.40	0.46	20	0.37	0.14	1.00	0.25	28	0.14
Mild	0.36	0.79	0.50	159		0.40	0.67	0.50	48		1.00	0.02	0.03	47	
Moderate	0.50	0.40	0.45	169		0.35	0.28	0.31	47		0.00	0.00	0.00	64	
Moderately Severe	0.61	0.21	0.32	89		0.00	0.00	0.00	19		0.00	0.00	0.00	30	
Severe	1.00	0.07	0.14	54		0.00	0.00	0.00	9		0.00	0.00	0.000	11	
GRU															
None	0.87	0.35	0.50	97	0.52	0.53	0.40	0.46	20	0.42	0.16	1.00	0.27	28	0.16
Mild	0.64	0.53	0.58	159		0.40	0.67	0.50	48		0.00	0.00	0.00	47	
Moderate	0.50	0.65	0.57	169		0.35	0.28	0.31	47		0.50	0.02	0.03	64	
Moderately Severe	0.41	0.40	0.41	89		0.00	0.00	0.00	19		0.00	0.00	0.00	30	
Severe	0.33	0.56	0.41	54		0.00	0.00	0.00	9		0.00	0.00	0.000	11	

จากตารางสรุปได้ดังนี้

1D CNN

ค่า Accuracy ใน (Training Result) สูงมาก (94%) แสดงถึงความสามารถในการทำนายข้อมูลที่ใช้ในการฝึกโมเดล แต่อาจแสดงการ overfitting หรือการเรียนรู้เกินไปกับข้อมูลฝึก

ค่า Accuracy ใน (Validation Result) ต่ำ (41%) แสดงถึงความสามารถในการทำนายข้อมูลที่ไม่เคยเห็นมาก่อน (out-of-sample data) และอาจชี้ชัดถึงปัญหา Overfitting

ค่า Accuracy ใน (Test Result) ต่ำกว่าทั้ง Training Result และ Validation Result (18%) แสดงถึงความสามารถในการทำนายข้อมูล

2D CNN

ค่า Accuracy ใน (Training Result) โมเดลมีความแม่นยำถึง 100% ในการทำนายชุดข้อมูลฝึกฝน ซึ่งมีความน่าเชื่อถือแต่อาจแสดงถึงโมเดลที่เรียนรู้จำความถูกต้องจากข้อมูลฝึกเกินไป (Overfitting)

ค่า Accuracy ใน (Validation Result) โมเดลมีความแม่นยำประมาณ 80% ในการทำนายชุดข้อมูลตรวจสอบ ซึ่งหมายความว่าโมเดลนั้นยังคงความแม่นยำและสามารถทำนายข้อมูลที่ไม่เคยเห็นมาก่อนได้ดี แต่ยังคงมีความคลาดเคลื่อนเล็กน้อย

ค่า Accuracy ใน (Test Result) โมเดลมีความแม่นยำประมาณ 85% ในการทำนายชุดข้อมูลทดสอบ ซึ่งเป็นการประเมินความสามารถในการใช้โมเดลในสภาวะการใช้งานจริง และมีความคลาดเคลื่อนน้อยกว่า Validation Result

LSTM

ผลรวมของความแม่นยำในชุดข้อมูลการฝึก (Training Result) คือ 0.44 (ประมาณ 44%)

ผลรวมของความแม่นยำในชุดข้อมูลการตรวจสอบความถูกต้อง (Validation Result) คือ 0.39 (ประมาณ 39%)

ผลรวมของความแม่นยำในชุดข้อมูลการทดสอบ (Test Result) คือ 0.16 (ประมาณ 16%)

โดยโมเดลอาจมีประสิทธิภาพที่ไม่ดีในการทำนายและความแม่นยำในการตรวจสอบความถูกต้องในชุดข้อมูลการตรวจสอบและการทดสอบมีค่าต่ำมาก ซึ่งอาจแสดงถึงปัญหาในการทำนายและการจำแนกข้อมูลของโมเดล ควรพิจารณาการปรับปรุงโมเดลหรือการเพิ่มข้อมูลเพื่อเพิ่มประสิทธิภาพในการทำนายและความแม่นยำในการตรวจสอบความถูกต้องของโมเดลในอนาคต

GRU

ในชุดข้อมูลการฝึก (Training Set) ความแม่นยำคือ 0.42 (42%)

ในชุดข้อมูลการทดสอบ (Validation Set) ความแม่นยำคือ 0.16 (16%)

ในชุดข้อมูลการทดสอบ (Test Set) ความแม่นยำคือ 0.52 (52%)

ความแม่นยำในชุดข้อมูลการฝึกสูงกว่าชุดข้อมูลการทดสอบและชุดข้อมูลการตรวจสอบ (Validation Set และ Test Set) ซึ่งอาจชี้ให้เห็นโมเดลมีประสิทธิภาพในการทำนายข้อมูลการฝึกมากกว่า

ความแม่นยำในชุดข้อมูลการทดสอบและชุดข้อมูลการตรวจสอบต่ำมาก แสดงถึงปัญหาในการทำนายคลาสเหล่านี้ในข้อมูลการทดสอบและการตรวจสอบ

สรุปคือโมเดลนี้มีประสิทธิภาพในการทำนายข้อมูลการฝึกมากที่สุดแต่ไม่สามารถทำนายคลาสอื่น ๆ ในข้อมูลการทดสอบและการตรวจสอบได้อย่างแม่นยำ แนวทางที่เป็นไปได้เพื่อปรับปรุงคือการเพิ่มข้อมูลการฝึก เปลี่ยนแปลงโมเดล หรือปรับพารามิเตอร์ของโมเดลให้เหมาะสมมากขึ้นในการทำนายคลาสอื่น ๆ ในข้อมูลการทดสอบและการตรวจสอบ และอาจต้องใช้เวลาทดลองและปรับปรุงเพิ่มเติมเพื่อปรับปรุงประสิทธิภาพของโมเดลในงานนี้ต่อไป

สรุปทั้งหมด

- 2D CNN ดีที่สุดของทั้ง 4 โมเดลในการทำนายทั้งใน 5 กรณี 1. ไม่มีอาการ 2. น้อย 3. ปานกลาง 4. มาก 5. รุนแรง

- 1D CNN, LSTM และ GRU มีปัญหาในการทำนายในทุกกรณีและค่าความแม่นยำน้อยมากและอาจต้องใช้เวลาทดลองและปรับปรุงเพิ่มเติมเพื่อปรับปรุงประสิทธิภาพของโมเดลในงานนี้ต่อไป

ตารางที่ 4.26 ผลของ Training Result, Validation Result, Test Result แสดงค่า Precision, Recall, F1-Score และ Accuracy ของการจำแนกประเภทที่แบ่งออกเป็น 5 ประเภทตามเกณฑ์ของ PHQ-9 ได้แก่ 1. ไม่มีอาการ 2. น้อย 3. ปานกลาง 4. มาก 5. รุนแรงของทั้ง 10 Model การเรียนรู้ของเครื่อง

Architecture	Training Result					Validation Result					Test Result				
	precision	recall	f1-score	support	accuracy	precision	recall	f1-score	support	accuracy	precision	recall	f1-score	support	accuracy
SVM (RBF)															
None	1.00	0.86	0.92	92	0.89	1.00	0.72	0.84	25	0.79	1.00	0.68	0.81	28	0.75
Mild	0.86	0.93	0.89	165		0.66	0.90	0.76	42		0.68	0.92	0.78	53	
Moderate	0.83	0.98	0.89	161		0.82	0.85	0.84	55		0.71	0.81	0.76	59	
Moderately Severe	0.98	0.86	0.91	99		1.00	0.78	0.88	9		0.93	0.46	0.62	28	
Severe	1.00	0.63	0.77	51		1.00	0.25	0.40	12		0.86	0.50	0.63	12	
SVM (Linear)															
None	1.00	1.00	1.00	92	0.97	0.84	0.84	0.84	25	0.70	0.81	0.61	0.69	28	0.61
Mild	0.95	0.98	0.96	165		0.73	0.57	0.64	42		0.56	0.55	0.55	53	
Moderate	0.97	0.95	0.96	161		0.77	0.75	0.76	55		0.59	0.63	0.61	59	
Moderately Severe	0.96	0.95	0.95	99		0.29	0.56	0.38	9		0.68	0.61	0.64	28	
Severe	0.98	0.98	0.98	51		0.60	0.75	0.67	12		0.53	0.83	0.65	12	
Random Forest															
None	1.00	1.00	1.00	92	1.00	1.00	0.76	0.86	25	0.87	0.92	0.86	0.89	28	0.84
Mild	1.00	1.00	1.00	165		0.78	0.93	0.85	42		0.81	0.91	0.86	53	
Moderate	1.00	1.00	1.00	161		0.88	0.93	0.90	55		0.77	0.90	0.83	59	
Moderately Severe	1.00	1.00	1.00	99		1.00	0.89	0.94	9		1.00	0.57	0.73	28	
Severe	1.00	1.00	1.00	51		1.00	0.67	0.80	12		0.73	0.83	0.91	12	

Gaussian Naive Bayes															
None	0.60	0.46	0.52	92	0.39	0.56	0.36	0.44	25	0.23	0.67	0.36	0.47	28	0.43
Mild	0.73	0.20	0.31	165		0.54	0.17	0.25	42		0.47	0.23	0.33	53	
Moderate	0.56	0.27	0.37	161		0.47	0.15	0.22	55		0.48	0.59	0.53	59	
Moderately Severe	0.26	0.96	0.41	99		0.08	0.89	0.15	9		0.33	0.32	0.33	28	
Severe	1.00	0.16	0.27	51		0.50	0.08	0.14	12		0.24	0.92	0.38	12	
Logistic Regression															
None	0.93	0.93	0.93	92	0.88	0.91	0.80	0.85	25	0.66	0.77	0.61	0.68	28	0.63
Mild	0.85	0.87	0.86	165		0.61	0.60	0.60	42		0.66	0.66	0.66	53	
Moderate	0.88	0.86	0.87	161		0.73	0.69	0.71	55		0.59	0.75	0.66	59	
Moderately Severe	0.87	0.87	0.87	99		0.28	0.56	0.37	9		0.50	0.29	0.36	28	
Severe	0.96	0.94	0.95	51		0.60	0.50	0.55	12		0.67	0.83	0.74	12	
Decision Tree															
None	1.00	1.00	1.00	92	1.00	0.65	0.60	0.63	25	0.60	0.66	0.68	0.67	28	0.59
Mild	1.00	1.00	1.00	165		0.64	0.71	0.67	42		0.60	0.62	0.61	53	
Moderate	1.00	1.00	1.00	161		0.70	0.58	0.63	55		0.59	0.64	0.62	59	
Moderately Severe	1.00	1.00	1.00	99		0.33	0.78	0.47	9		0.50	0.32	0.39	28	
Severe	1.00	1.00	1.00	51		0.33	0.17	0.22	12		0.57	0.67	0.62	12	
AdaBoost															
None	0.52	0.63	0.57	92	0.52	0.67	0.48	0.56	25	0.41	0.25	0.21	0.23	28	0.39
Mild	0.48	0.56	0.52	165		0.38	0.48	0.42	42		0.44	0.47	0.45	53	
Moderate	0.55	0.52	0.53	161		0.50	0.42	0.46	55		0.47	0.47	0.47	59	
Moderately Severe	0.53	0.48	0.51	99		0.12	0.33	0.18	9		0.27	0.21	0.24	28	

Severe	0.64	0.31	0.42	51		0.50	0.08	0.14	12		0.29	0.42	0.34	12	
SGD (Stochastic Gradient Descent)															
None	0.41	0.91	0.56	92	0.48	0.46	0.84	0.59	25	0.50	0.25	0.21	0.23	28	0.39
Mild	0.89	0.10	0.17	165		1.00	0.07	0.13	42		0.44	0.47	0.45	53	
Moderate	0.54	0.78	0.64	161		0.64	0.71	0.67	55		0.47	0.47	0.47	59	
Moderately Severe	0.69	0.11	0.19	99		0.50	0.11	0.18	9		0.27	0.21	0.24	28	
Severe	0.40	0.75	0.52	51		0.23	0.58	0.33	12		0.29	0.42	0.34	12	
MLP															
None	1.00	1.00	1.00	92	1.00	0.89	0.96	0.92	25	0.89	0.87	0.96	0.92	28	0.91
Mild	1.00	1.00	1.00	165		0.86	0.90	0.88	42		0.94	0.92	0.93	53	
Moderate	1.00	1.00	1.00	161		0.94	0.85	0.90	55		0.89	0.92	0.90	59	
Moderately Severe	1.00	1.00	1.00	99		0.89	0.96	0.92	9		1.00	0.82	0.90	28	
Severe	1.00	1.00	1.00	51		0.91	0.83	0.87	12		0.77	0.83	0.80	12	
KNN															
None	1.00	1.00	1.00	92	1.00	0.86	0.96	0.91	25	0.88	0.87	0.93	0.90	28	0.86
Mild	1.00	1.00	1.00	165		0.88	0.88	0.88	42		0.85	0.85	0.85	53	
Moderate	1.00	1.00	1.00	161		0.94	0.85	0.90	55		0.84	0.90	0.87	59	
Moderately Severe	1.00	1.00	1.00	99		0.80	0.89	0.84	9		1.00	0.71	0.83	28	
Severe	1.00	1.00	1.00	51		0.77	0.83	0.80	12		0.71	0.83	0.77	12	

จากตารางสรุปได้ดังนี้

จากผลการทดสอบทั้งหมด โมเดลที่มีประสิทธิภาพที่ดีที่สุดในการจำแนกคลาสทั้ง 5 คลาสในชุดข้อมูลทดสอบคือ MLP (Multi-Layer Perceptron) ซึ่งมีความแม่นยำประมาณ 91% และเหมาะสมสำหรับงานนี้อย่างมีนัยสำคัญ

KNN (K Nearest Neighbour) และ Random Forest มีความแม่นยำสูงที่สุดในการทดสอบ (Test) ที่ 86%, 84% ตามลำดับ

SVM (RBF) มีค่าความแม่นยำมากกว่า 70%

AdaBoost และ SGD (Stochastic Gradient Descent) โดยรวมให้ผลลัพธ์ที่ต่ำกว่าโมเดลอื่นในทุกขั้นตอนที่ 39%

ส่วน SVM (Linear), Gaussian Naive Bayes, Logistic Regression, Decision Tree อยู่ในระหว่าง 40% ถึง 65%

ตารางที่ 4.27 เปรียบเทียบผลรวมค่าความแม่นยำของประเภทที่ 2 การจำแนกประเภทที่แบ่งออกเป็น 5 ประเภทตามเกณฑ์ของ PHQ-9

ได้แก่ 1. ไม่มีอาการ 2. น้อย 3. ปานกลาง 4. มาก 5. รุนแรง

แบบจำลองการเรียนรู้เชิงลึก		แบบจำลองการเรียนรู้ของเครื่อง	
Model	Accuracy	Model	Accuracy
1D CNN	25%	SVM (RBF)	75%
2D CNN	85%	SVM (Linear)	61%
LSTM	14%	Random Forest	84%
GRU	16%	Gaussian Naive Bayes	43%
		Logistic Regression	63%
		Decision Tree	59%
		AdaBoost	39%
		SGD	39%
		MLP	91%
		KNN	86%

สรุปผลได้ดังนี้

2D CNN มีความแม่นยำสูงสุดที่ประมาณ 85% ซึ่งแสดงให้เห็นว่าแบบจำลองนี้มีประสิทธิภาพที่ดีในการทำนายข้อมูลในงานที่เกี่ยวข้องกับภาพหรือข้อมูล

สองมิติ

1D CNN, LSTM, และ GRU มีความแม่นยำต่ำมาก ซึ่งอาจจะไม่เหมาะสมในงานที่ต้องการความแม่นยำสูง MLP (Multi-Layer Perceptron) และ KNN (K Nearest Neighbour) มีความแม่นยำสูงสุดที่ประมาณ 91% และ 86% ตามลำดับ ซึ่งแสดงให้เห็นว่าแบบจำลองเหล่านี้มีประสิทธิภาพที่ดีในการทำนายข้อมูล

Random Forest มีความแม่นยำประมาณ 84% ซึ่งก็มีประสิทธิภาพไม่น้อยเมื่อเปรียบเทียบกับ MLP และ KNN SVM (RBF), Logistic Regression, และ SVM (Linear) มีความแม่นยำอยู่ในระดับประมาณ 60-75% ซึ่งยังคงเป็นแบบจำลองที่มีความแม่นยำในระดับยาก Gaussian Naive Bayes, Decision Tree, AdaBoost, และ SGD มีความแม่นยำต่ำที่ประมาณ 39-63% แสดงให้เห็นว่าแบบจำลองเหล่านี้อาจจะไม่เหมาะสมในงานที่ต้องการความแม่นยำสูง

เมื่อเปรียบเทียบระหว่างการเรียนรู้เชิงลึกกับการเรียนรู้เรื่องเครื่องโมเดลที่มีความแม่นยำสูงที่สุดของการเรียนรู้เชิงลึกคือ 2D CNN ได้เปอร์เซ็นต์ที่ 85% ส่วนการเรียนรู้ของเครื่องโมเดลที่ได้ค่าความแม่นยำสูงที่สุดคือ MLP ได้เปอร์เซ็นต์ที่ 91% โดยเหตุผลที่การเรียนรู้ของเครื่องได้ค่าความแม่นยำมากกว่าการเรียนรู้เชิงลึกอาจเป็นเพราะการเรียนรู้เชิงลึกเป็นสาขาย่อยของการเรียนรู้ของเครื่องที่เน้นการใช้โครงข่ายประสาทเทียมเพื่อเรียนรู้รูปแบบที่ซับซ้อนในข้อมูล การเรียนรู้เชิงลึกมีความแม่นยำสูงกว่าการเรียนรู้ของเครื่องโดยทั่วไป แต่ต้องใช้ข้อมูลมากกว่าและมีความซับซ้อนมากกว่า การเลือกระหว่างการเรียนรู้ของเครื่องและการเรียนรู้เชิงลึกขึ้นอยู่กับปัจจัยหลายประการ เช่น ลักษณะของงาน ความแม่นยำที่ต้องการ ปริมาณข้อมูลที่มีอยู่ และทรัพยากรที่มี หากงานต้องการความแม่นยำสูงและสามารถใช้ข้อมูลจำนวนมากได้ การเรียนรู้เชิงลึกอาจเป็นตัวเลือกที่ดีกว่า หากงานต้องการความรวดเร็วและสามารถใช้ข้อมูลจำนวนน้อยได้ การเรียนรู้ของเครื่องอาจเป็นตัวเลือกที่ดีกว่า โดย Dataset ของงานวิจัยนี้ในรูปแบบที่สองจะมีค่อนข้างจำกัดนี้อาจจะเป็นเหตุผลที่การเรียนรู้ของเครื่องได้ค่าความแม่นยำสูงกว่าการเรียนรู้เชิงลึก

4.2.2.2 Confusion Matrix และ Training loss and test loss graph ของแบบจำลองการเรียนรู้เชิงลึกและการเรียนรู้ของเครื่องของข้อมูล 5 ประเภท

1D CNN

จากข้อมูล Confusion Matrix ของ Test Result มีการวิเคราะห์เพื่อให้เห็นภาพรวมของผลการทดสอบดังต่อไปนี้

สำหรับคลาส "น้อย" โมเดลทำนายถูกต้องใน 5 กรณี และทำนายผิด 42 กรณี

สำหรับคลาส "ปานกลาง" โมเดลทำนายถูกต้องใน 0 กรณี และทำนายผิด 64 กรณี

สำหรับคลาส "มาก" โมเดลทำนายถูกต้องใน 0 กรณี และทำนายผิด 30 กรณี

สำหรับคลาส "ไม่มีอาการ" โมเดลทำนายถูกต้องใน 27 กรณี และทำนายผิด 1 กรณี

สำหรับคลาส "รุนแรง" โมเดลทำนายถูกต้องใน 0 กรณี และทำนายผิด 11 กรณี

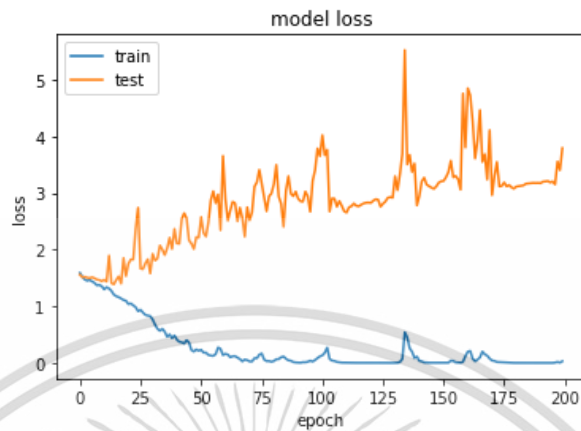
โดยในภาพ % สีเขียว คือโมเดลทำนายถูกต้องส่วน % สีแดง คือโมเดลทำนายผิดพลาดดัง

ภาพที่ 4.9

5 Class (1D-CNN)						
TARGET \ OUTPUT	Mild	Moderate	Moderately Severe	None	Severe	SUM
Mild	5 10.64% 89.36%	2	0	40	0	47
Moderate	1	0 0.00% 100.00%	0	62	1	64
Moderately Severe	3	0	0 0.00% 100.00%	27	0	30
None	1	0	0	27 96.43% 3.57%	0	28
Severe	0	0	1	10	0 0.00% 100.00%	11
SUM	10 50.00% 50.00%	2 100.00% 0.00%	1 100.00% 0.00%	166 16.27% 83.73%	1 100.00% 0.00%	32 / 180 17.78% 82.22%

ภาพที่ 4.19 Confusion Matrix ของ Test Result (1D CNN-5 Class)

Training loss and test loss graph (1D CNN-5 Class)



ภาพที่ 4.20 Training and test accuracy and loss (1D CNN-5 Class)

ตารางที่ 4.28 Sensitivity และ Specificity (1D CNN-5 Class)

Class	Sensitivity (ความไวต่อการตรวจพบ)	Specificity (ความไวต่อการตรวจปฏิเสธ)
สำหรับคลาส "ไม่มีอาการ"	ค่า Sensitivity ประมาณ 0%	Specificity คือ 100%
สำหรับคลาส "น้อย"	ค่า Sensitivity ประมาณ 10.6%	Specificity ประมาณ 90.7%
สำหรับคลาส "ปานกลาง"	ค่า Sensitivity ประมาณ 0%	Specificity คือ 77.1%
สำหรับคลาส "มาก"	ค่า Sensitivity ประมาณ 0%	Specificity คือ 100%
สำหรับคลาส "รุนแรง"	ค่า Sensitivity ประมาณ 0%	Specificity คือ 91.9%

สำหรับคลาส "น้อย" มีความแม่นยำสูงในการตรวจพบคลาสไม่มีอาการ อย่างไรก็ตาม Sensitivity ที่ต่ำกว่าหมายถึงโมเดลไม่ดีในการตรวจพบคลาสน้อย เป็นตัวหนึ่งที่ต้องพิจารณา

สำหรับคลาส "ปานกลาง" มี Sensitivity ที่เป็นศูนย์แสดงว่าโมเดลไม่สามารถตรวจพบคลาสนี้ได้ แต่มีความแม่นยำในการตรวจพบคลาสไม่มีอาการ

สำหรับคลาส "มาก" มี Sensitivity ที่เป็นศูนย์แสดงว่าโมเดลไม่สามารถตรวจพบคลาสนี้ได้ และมีความแม่นยำในการตรวจพบคลาสไม่มีอาการ

สำหรับคลาส "ไม่มีอาการ" มี Sensitivity ที่เป็นศูนย์แสดงว่าโมเดลไม่สามารถตรวจพบคลาสนี้ได้ และมีความแม่นยำในการตรวจพบคลาสไม่มีอาการ

สำหรับคลาส "รุนแรง" มี Sensitivity ที่เป็นศูนย์แสดงว่าโมเดลไม่สามารถตรวจพบคลาสนี้ได้ แต่มีความแม่นยำในการตรวจพบคลาสไม่มีอาการ

2D CNN

จากข้อมูล Confusion Matrix ของ Test Result มีการวิเคราะห์เพื่อให้เห็นภาพรวมของผลการทดสอบดังต่อไปนี้

สำหรับคลาส "น้อย" โมเดลทำนายถูกต้องใน 37 กรณี และทำนายผิด 10 กรณี

สำหรับคลาส "ปานกลาง" โมเดลทำนายถูกต้องใน 58 กรณี และทำนายผิด 6 กรณี

สำหรับคลาส "มาก" โมเดลทำนายถูกต้องใน 25 กรณี และทำนายผิด 5 กรณี

สำหรับคลาส "ไม่มีอาการ" โมเดลทำนายถูกต้องใน 23 กรณี และทำนายผิด 5 กรณี

สำหรับคลาส "รุนแรง" โมเดลทำนายถูกต้องใน 10 กรณี และทำนายผิด 1 กรณี

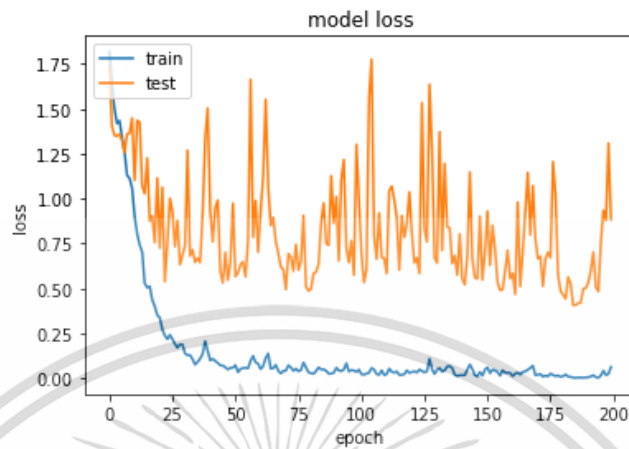
โดยในภาพ % สีเขียว คือโมเดลทำนายถูกต้องส่วน % สีแดง คือโมเดลทำนายผิดพลาดดัง

ภาพที่ 4.11

		5 Class (2D-CNN)					
TARGET \ OUTPUT	Mild	Moderate	Moderately Severe	None	Severe	SUM	
Mild	37	5	1	4	0	47	78.72% 21.28%
Moderate	1	58	2	3	0	64	90.63% 9.38%
Moderately Severe	0	4	25	1	0	30	83.33% 16.67%
None	2	3	0	23	0	28	82.14% 17.86%
Severe	0	1	0	0	10	11	90.91% 9.09%
SUM	40	71	28	31	10	153 / 180	92.50% 7.50% 81.69% 18.31% 89.29% 10.71% 74.19% 25.81% 100.00% 0.00% 85.00% 15.00%

ภาพที่ 4.21 Confusion Matrix ของ Test Result (2D CNN-5 Class)

Training loss and test loss graph (2D CNN-5 Class)



ภาพที่ 4.22 Training and test accuracy and loss (2D CNN-5 Class)

ตารางที่ 4.29 Sensitivity และ Specificity (2D CNN-5 Class)

Class	Sensitivity (ความไวต่อการตรวจพบ)	Specificity (ความไวต่อการตรวจปฏิเสธ)
สำหรับคลาส "ไม่มีอาการ"	ค่า Sensitivity ประมาณ 78.7%	Specificity คือ 77.7%
สำหรับคลาส "น้อย"	ค่า Sensitivity ประมาณ 90.6%	Specificity ประมาณ 100%
สำหรับคลาส "ปานกลาง"	ค่า Sensitivity ประมาณ 83.3%	Specificity คือ 100%
สำหรับคลาส "มาก"	ค่า Sensitivity ประมาณ 82.1%	Specificity คือ 100%
สำหรับคลาส "รุนแรง"	ค่า Sensitivity ประมาณ 90.9%	Specificity คือ 100%

ค่า Sensitivity แสดงความสามารถในการตรวจพบความจริงของคลาสบวก (Positive Class) ในขณะที่ค่า Specificity แสดงความสามารถในการตรวจพบความจริงของคลาสลบ (Negative Class) โดยโมเดลมีความแม่นยำสูงในการทำนายทุกคลาสและค่า Sensitivity และ Specificity สูงมากสำหรับทุกคลาส ซึ่งแสดงถึงความเป็นไปได้ที่โมเดลมีประสิทธิภาพสูงในการทำนายคลาสต่าง ๆ ในข้อมูลนี้

LSTM

จากข้อมูล Confusion Matrix ของ Test Result มีการวิเคราะห์เพื่อให้เห็นภาพรวมของผลการทดสอบดังต่อไปนี้

สำหรับคลาส "น้อย" โมเดลทำนายถูกต้องใน 0 กรณี และทำนายผิด 47 กรณี

สำหรับคลาส "ปานกลาง" โมเดลทำนายถูกต้องใน 0 กรณี และทำนายผิด 64 กรณี

สำหรับคลาส "มาก" โมเดลทำนายถูกต้องใน 0 กรณี และทำนายผิด 30 กรณี

สำหรับคลาส "ไม่มีอาการ" โมเดลทำนายถูกต้องใน 28 กรณี และทำนายผิด 0 กรณี

สำหรับคลาส "รุนแรง" โมเดลทำนายถูกต้องใน 0 กรณี และทำนายผิด 11 กรณี

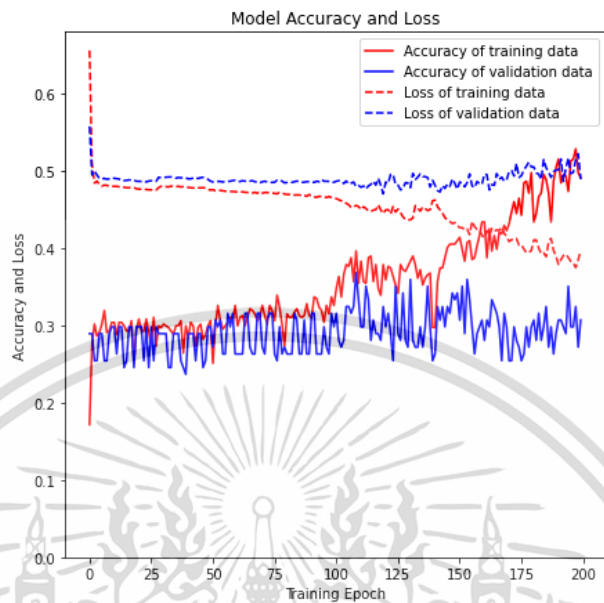
จาก Confusion Matrix นี้ จะเห็นว่าโมเดลมีปัญหาในการทำนายและความแม่นยำในการจำแนกคลาส เฉพาะคลาส "น้อย," "ปานกลาง," "มาก," และ "รุนแรง" มีค่า True Positive เพียงเล็กน้อยหรือไม่มีเลย ควรพิจารณาการปรับปรุงโมเดลหรือการเพิ่มข้อมูลเพื่อปรับปรุงประสิทธิภาพในการทำนายและการจำแนกคลาสในอนาคต

โดยในภาพ % สีเขียว คือโมเดลทำนายถูกต้องส่วน % สีแดง คือโมเดลทำนายผิดพลาดดังภาพที่ 4.13

		5 Class (LSTM)					
TARGET \ OUTPUT	Mild	Moderate	Moderately Severe	None	Severe	SUM	
Mild	0	0	0	47	0	47	0.00% 100.00%
Moderate	2	0	0	62	0	64	0.00% 100.00%
Moderately Severe	1	0	0	29	0	30	0.00% 100.00%
None	0	0	0	28	0	28	100.00% 0.00%
Severe	0	0	0	11	0	11	0.00% 100.00%
SUM	3 100.00% 0.00%	0 NaN% NaN%	0 NaN% NaN%	177 15.82% 84.18%	0 NaN% NaN%	28 / 180	15.56% 84.44%

ภาพที่ 4.23 Confusion Matrix ของ Test Result (LSTM-5 Class)

Training loss and test loss graph (LSTM-5 Class)



ภาพที่ 4.24 Training and test accuracy and loss (LSTM-5 Class)

ตารางที่ 4.30 Sensitivity และ Specificity (LSTM-5 Class)

Class	Sensitivity (ความไวต่อการตรวจพบ)	Specificity (ความไวต่อการตรวจปฏิเสธ)
สำหรับคลาส "ไม่มีอาการ"	ในกรณีนี้ โมเดลไม่ได้ทำนายคลาสใดๆ ให้ถูกต้องเลย (True Positive = 0) สำหรับทุกคลาส ดังนั้นไม่สามารถคำนวณ Sensitivity และ Specificity ได้ เนื่องจากไม่มีการทำนายใดๆ ที่ถูกต้องเกิดขึ้น ทั้งนี้เนื่องจากมีค่า True Negative (True Negative = 0) และ False Positive (False Positive = 0) ในทุกคลาสด้วย ดังนั้นไม่สามารถสร้างการแบ่งแยก (Discrimination) ได้ในกรณีนี้เนื่องจากไม่มีการทำนายที่ถูกต้องในทุกคลาส	
สำหรับคลาส "น้อย"		
สำหรับคลาส "ปานกลาง"		
สำหรับคลาส "มาก"		
สำหรับคลาส "รุนแรง"		

GRU

จากข้อมูล Confusion Matrix ของ Test Result มีการวิเคราะห์เพื่อให้เห็นภาพรวมของผลการทดสอบดังต่อไปนี้

สำหรับคลาส "น้อย" โมเดลทำนายถูกต้องใน 0 กรณี และทำนายผิด 47 กรณี

สำหรับคลาส "ปานกลาง" โมเดลทำนายถูกต้องใน 1 กรณี และทำนายผิด 63 กรณี

สำหรับคลาส "มาก" โมเดลทำนายถูกต้องใน 0 กรณี และทำนายผิด 30 กรณี

สำหรับคลาส "ไม่มีอาการ" โมเดลทำนายถูกต้องใน 28 กรณี และทำนายผิด 0 กรณี

สำหรับคลาส "รุนแรง" โมเดลทำนายถูกต้องใน 0 กรณี และทำนายผิด 11 กรณี

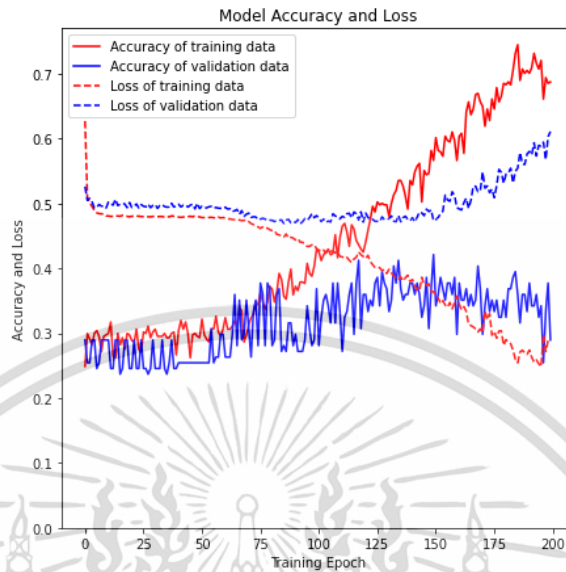
จาก Confusion Matrix นี้ จะเห็นว่าโมเดลมีปัญหาในการทำนายและความแม่นยำในการจำแนกคลาส เฉพาะคลาส "น้อย," "ปานกลาง," "มาก," และ "รุนแรง" มีค่า True Positive เพียงเล็กน้อย หรือไม่มีเลย ควรพิจารณาการปรับปรุงโมเดลหรือการเพิ่มข้อมูลเพื่อปรับปรุงประสิทธิภาพในการทำนาย และการจำแนกคลาสในอนาคต

โดยในภาพ % สีเขียว คือโมเดลทำนายถูกต้องส่วน % สีแดง คือโมเดลทำนายผิดพลาดดังภาพที่ 4.15

5 Class (GRU)						
TARGET \ OUTPUT	Mild	Moderate	Moderately Severe	None	Severe	SUM
Mild	0 100.00% 0.00%	0 50.00% 50.00%	0 NaN% NaN%	47 15.91% 84.09%	0 NaN% NaN%	47 0.00% 100.00%
Moderate	2 100.00% 0.00%	1 50.00% 50.00%	0 NaN% NaN%	61 15.91% 84.09%	0 NaN% NaN%	64 1.56% 98.44%
Moderately Severe	0 100.00% 0.00%	1 50.00% 50.00%	0 NaN% NaN%	29 15.91% 84.09%	0 NaN% NaN%	30 0.00% 100.00%
None	0 100.00% 0.00%	0 50.00% 50.00%	0 NaN% NaN%	28 15.91% 84.09%	0 NaN% NaN%	28 100.00% 0.00%
Severe	0 100.00% 0.00%	0 50.00% 50.00%	0 NaN% NaN%	11 15.91% 84.09%	0 NaN% NaN%	11 0.00% 100.00%
SUM	2 100.00% 0.00%	2 50.00% 50.00%	0 NaN% NaN%	176 15.91% 84.09%	0 NaN% NaN%	29 / 180 16.11% 83.89%

ภาพที่ 4.25 Confusion Matrix ของ Test Result (GRU-5 Class)

Training loss and test loss graph (GRU-5 Class)



ภาพที่ 4.26 Training and test accuracy and loss (GRU-5 Class)

ตารางที่ 4.31 Sensitivity และ Specificity (GRU-5 Class)

Class	Sensitivity (ความไวต่อการตรวจพบ)	Specificity (ความไวต่อการตรวจปฏิเสธ)
สำหรับคลาส "ไม่มีอาการ"	ในกรณีนี้ โมเดลไม่ได้ทำนายคลาสใดๆ ให้ถูกต้องเลย (True Positive = 0) สำหรับทุกคลาส ดังนั้นไม่สามารถคำนวณ Sensitivity และ Specificity ได้ เนื่องจากไม่มีการทำนายใดๆ ที่ถูกต้องเกิดขึ้น ทั้งนี้เนื่องจากมีค่า True Negative (True Negative = 0) และ False Positive (False Positive = 0) ในทุกคลาสด้วย ดังนั้นไม่สามารถสร้างการแบ่งแยก (Discrimination) ได้ในกรณีนี้เนื่องจากไม่มีการทำนายที่ถูกต้องในทุกคลาส	
สำหรับคลาส "น้อย"		
สำหรับคลาส "ปานกลาง"		
สำหรับคลาส "มาก"		
สำหรับคลาส "รุนแรง"		

KNN (K Nearest Neighbour)

จากข้อมูล Confusion Matrix ของ Test Result มีการวิเคราะห์เพื่อให้เห็นภาพรวมของผลการทดสอบดังต่อไปนี้

สำหรับคลาส "น้อย" โมเดลทำนายถูกต้องใน 45 กรณี และทำนายผิด 8 กรณี

สำหรับคลาส "ปานกลาง" โมเดลทำนายถูกต้องใน 53 กรณี และทำนายผิด 6 กรณี

สำหรับคลาส "มาก" โมเดลทำนายถูกต้องใน 20 กรณี และทำนายผิด 8 กรณี

สำหรับคลาส "ไม่มีอาการ" โมเดลทำนายถูกต้องใน 26 กรณี และทำนายผิด 2 กรณี

สำหรับคลาส "รุนแรง" โมเดลทำนายถูกต้องใน 10 กรณี และทำนายผิด 2 กรณี

โดยในภาพ % สีเขียว คือโมเดลทำนายถูกต้องส่วน % สีแดง คือโมเดลทำนายผิดพลาดตั้ง

ภาพที่ 4.27

		KNN					
TARGET \ OUTPUT	Mild	Moderate	Moderately severe	None	Severe	SUM	
Mild	45	3	0	3	2	53	84.91% 15.09%
Moderate	5	53	0	0	1	59	89.83% 10.17%
Moderately severe	2	5	20	1	0	28	71.43% 28.57%
None	0	1	0	26	1	28	92.86% 7.14%
Severe	1	1	0	0	10	12	83.33% 16.67%
SUM	53	63	20	30	14	154 / 180	85.56% 14.44%

ภาพที่ 4.27 Confusion Matrix ของ Test Result KNN Model (5 Class)

ตารางที่ 4.32 Sensitivity และ Specificity (KNN-5 Class)

Class	Sensitivity (ความไวต่อการตรวจพบ)	Specificity (ความไวต่อการตรวจปฏิเสธ)
สำหรับคลาส "ไม่มีอาการ"	ค่า Sensitivity คือ 96.30%	Specificity คือ 92.03%
สำหรับคลาส "น้อย"	ค่า Sensitivity คือ 93.75%	Specificity คือ 90.17%
สำหรับคลาส "ปานกลาง"	ค่า Sensitivity คือ 91.38%	Specificity คือ 90.91%
สำหรับคลาส "มาก"	ค่า Sensitivity คือ 95.24%	Specificity คือ 92.48%
สำหรับคลาส "รุนแรง"	ค่า Sensitivity คือ 90.91%	Specificity คือ 92.90%

สรุปได้ว่าโมเดลมีความแม่นยำในการทำนายคลาส "น้อย" และ "ปานกลาง" ด้วยค่า Sensitivity และ Specificity ที่สูง ซึ่งหมายความว่ามีความสามารถในการตรวจพบและตรวจไม่พบคลาสเหล่านี้ได้ดี ในขณะที่คลาส "มาก" และ "ไม่มีอาการ" ยังคงมีความแม่นยำสูง แต่ Sensitivity มีค่าสูงกว่า Specificity ซึ่งหมายความว่ามีโอกาสตรวจพบคลาสเหล่านี้ได้ดีกว่าการตรวจไม่พบ ส่วนคลาส "รุนแรง" มีค่า Sensitivity และ Specificity ที่ใกล้เคียงกัน

MLP (Multi-Layer Perceptron)

จากข้อมูล Confusion Matrix ของ Test Result มีการวิเคราะห์เพื่อให้เห็นภาพรวมของผลการทดสอบดังต่อไปนี้

สำหรับคลาส "น้อย" โมเดลทำนายถูกต้องใน 49 กรณี และทำนายผิด 4 กรณี

สำหรับคลาส "ปานกลาง" โมเดลทำนายถูกต้องใน 54 กรณี และทำนายผิด 5 กรณี

สำหรับคลาส "มาก" โมเดลทำนายถูกต้องใน 23 กรณี และทำนายผิด 5 กรณี

สำหรับคลาส "ไม่มีอาการ" โมเดลทำนายถูกต้องใน 27 กรณี และทำนายผิด 1 กรณี

สำหรับคลาส "รุนแรง" โมเดลทำนายถูกต้องใน 10 กรณี และทำนายผิด 2 กรณี

โดยในภาพ % สีเขียว คือโมเดลทำนายถูกต้องส่วน % สีแดง คือโมเดลทำนายผิดพลาดดังภาพที่ 4.28

MLP						
TARGET \ OUTPUT	Mild	Moderate	Moderately severe	None	Severe	SUM
Mild	49 92.45% 7.55%	2	0	1	1	53
Moderate	2	54 91.53% 8.47%	0	1	2	59
Moderately severe	0	3	23 82.14% 17.86%	2	0	28
None	0	1	0	27 96.43% 3.57%	0	28
Severe	1	1	0	0	10 83.33% 16.67%	12
SUM	52 94.23% 5.77%	61 88.52% 11.48%	23 100.00% 0.00%	31 87.10% 12.90%	13 76.92% 23.08%	163 / 180 90.56% 9.44%

ภาพที่ 4.28 Confusion Matrix ของ Test Result MLP Model (5 Class)

ตารางที่ 4.33 Sensitivity และ Specificity (MLP-5 Class)

Class	Sensitivity (ความไวต่อการตรวจพบ)	Specificity (ความไวต่อการตรวจปฏิเสธ)
สำหรับคลาส "ไม่มีอาการ"	ค่า Sensitivity คือ 96%	Specificity คือ 79%
สำหรับคลาส "น้อย"	ค่า Sensitivity คือ 98%	Specificity คือ 95%
สำหรับคลาส "ปานกลาง"	ค่า Sensitivity คือ 95%	Specificity คือ 72%
สำหรับคลาส "มาก"	ค่า Sensitivity คือ 82%	Specificity คือ 79%
สำหรับคลาส "รุนแรง"	ค่า Sensitivity คือ 83%	Specificity คือ 86%

สรุปโมเดลเหมาะสมในการตรวจจับคลาส "น้อย" และ "รุนแรง" มากที่สุด ในขณะที่คลาส "ปานกลาง" และ "มาก" ค่า Sensitivity น้อยกว่าค่า Specificity ซึ่งหมายความว่าโมเดลอาจจะมีโอกาสตรวจไม่พบคลาสเหล่านี้ได้ดีกว่าการตรวจพบ และสุดท้ายคลาส "ไม่มีอาการ" มีค่า Sensitivity สูงและค่า

Specificity น้อยกว่า ซึ่งแสดงว่าโมเดลอาจจะตรวจพบคลาส "ไม่มีอาการ" ได้ดีกว่าการตรวจไม่พบ คือ คลาสอื่น ๆ และอาจจะมีการตรวจพบเทียบเพิ่มขึ้นกว่าที่ต้องการในบางกรณี

Random Forest

จากข้อมูล Confusion Matrix ของ Test Result มีการวิเคราะห์เพื่อให้เห็นภาพรวมของผลการทดสอบดังต่อไปนี้

สำหรับคลาส "น้อย" โมเดลทำนายถูกต้องใน 48 กรณี และทำนายผิด 5 กรณี

สำหรับคลาส "ปานกลาง" โมเดลทำนายถูกต้องใน 53 กรณี และทำนายผิด 6 กรณี

สำหรับคลาส "มาก" โมเดลทำนายถูกต้องใน 16 กรณี และทำนายผิด 12 กรณี

สำหรับคลาส "ไม่มีอาการ" โมเดลทำนายถูกต้องใน 24 กรณี และทำนายผิด 4 กรณี

สำหรับคลาส "รุนแรง" โมเดลทำนายถูกต้องใน 10 กรณี และทำนายผิด 2 กรณี

โดยในภาพ % สีเขียว คือโมเดลทำนายถูกต้องส่วน % สีแดง คือโมเดลทำนายผิดพลาดดัง

ภาพที่ 4.29

Random Forest						
TARGET \ OUTPUT	Mild	Moderate	Moderately severe	None	Severe	SUM
Mild	48 90.57% 9.43%	3	0	2	0	53
Moderate	6	53 89.83% 10.17%	0	0	0	59
Moderately severe	3	9	16 57.14% 42.86%	0	0	28
None	2	2	0	24 85.71% 14.29%	0	28
Severe	0	2	0	0	10 83.33% 16.67%	12
SUM	59 81.36% 18.64%	69 76.81% 23.19%	16 100.00% 0.00%	26 92.31% 7.69%	10 100.00% 0.00%	151 / 180 83.89% 16.11%

ภาพที่ 4.29 Confusion Matrix ของ Test Result Random Forest Model (5 Class)

ตารางที่ 4.34 Sensitivity และ Specificity (Random Forest-5 Class)

Class	Sensitivity (ความไวต่อการตรวจพบ)	Specificity (ความไวต่อการตรวจปฏิเสธ)
สำหรับคลาส "ไม่มีอาการ"	ค่า Sensitivity คือ 86%	Specificity คือ 69%
สำหรับคลาส "น้อย"	ค่า Sensitivity คือ 96%	Specificity คือ 84%
สำหรับคลาส "ปานกลาง"	ค่า Sensitivity คือ 77%	Specificity คือ 53%
สำหรับคลาส "มาก"	ค่า Sensitivity คือ 100%	Specificity คือ 73%
สำหรับคลาส "รุนแรง"	ค่า Sensitivity คือ 100%	Specificity คือ 76%

สรุปโมเดลเหมาะสมในการตรวจพบคลาส "น้อย", "มาก", "ไม่มีอาการ", และ "รุนแรง" แต่มีความยากลำบากในการตรวจพบคลาส "ปานกลาง" เนื่องจากค่า Sensitivity และ Specificity ของคลาสนี้ต่ำกว่าค่าของคลาสนอื่น ๆ ที่มีค่าใกล้เคียงกัน

Logistic Regression

จากข้อมูล Confusion Matrix ของ Test Result มีการวิเคราะห์เพื่อให้เห็นภาพรวมของผลการทดสอบดังต่อไปนี้

สำหรับคลาส "น้อย" โมเดลทำนายถูกต้องใน 35 กรณี และทำนายผิด 18 กรณี

สำหรับคลาส "ปานกลาง" โมเดลทำนายถูกต้องใน 44 กรณี และทำนายผิด 15 กรณี

สำหรับคลาส "มาก" โมเดลทำนายถูกต้องใน 8 กรณี และทำนายผิด 20 กรณี

สำหรับคลาส "ไม่มีอาการ" โมเดลทำนายถูกต้องใน 17 กรณี และทำนายผิด 11 กรณี

สำหรับคลาส "รุนแรง" โมเดลทำนายถูกต้องใน 10 กรณี และทำนายผิด 2 กรณี

โดยในภาพ % สีเขียว คือโมเดลทำนายถูกต้องส่วน % สีแดง คือโมเดลทำนายผิดพลาดดัง

ภาพที่ 4.30

Logistic Regression						
TARGET \ OUTPUT	Mild	Moderate	Moderately severe	None	Severe	SUM
Mild	35	10	3	4	1	53 66.04% 33.96%
Moderate	8	44	4	0	3	59 74.58% 25.42%
Moderately severe	4	14	8	1	1	28 28.57% 71.43%
None	5	5	1	17	0	28 60.71% 39.29%
Severe	1	1	0	0	10	12 83.33% 16.67%
SUM	53 66.04% 33.96%	74 59.46% 40.54%	16 50.00% 50.00%	22 77.27% 22.73%	15 66.67% 33.33%	114 / 180 63.33% 36.67%

ภาพที่ 4.30 Confusion Matrix ของ Test Result Logistic Regression Model (5 Class)

ตารางที่ 4.35 Sensitivity และ Specificity (Logistic Regression-5 Class)

Class	Sensitivity (ความไวต่อการตรวจพบ)	Specificity (ความไวต่อการตรวจปฏิเสธ)
สำหรับคลาส "ไม่มีอาการ"	ค่า Sensitivity คือ 77%	Specificity คือ 87%
สำหรับคลาส "น้อย"	ค่า Sensitivity คือ 66%	Specificity คือ 80%
สำหรับคลาส "ปานกลาง"	ค่า Sensitivity คือ 71%	Specificity คือ 47%
สำหรับคลาส "มาก"	ค่า Sensitivity คือ 50%	Specificity คือ 91%
สำหรับคลาส "รุนแรง"	ค่า Sensitivity คือ 67%	Specificity คือ 100%

สรุปโมเดลเหมาะสมในการตรวจคลาส "น้อย" และ "ไม่มีอาการ" แต่มีความยากลำบากในการตรวจคลาส "ปานกลาง" เนื่องจากค่า Sensitivity และ Specificity ของคลาสนี้ต่ำกว่าค่าของคลาสนอื่น ๆ ที่มีค่าใกล้เคียงกัน

SVM (Linear)

จากข้อมูล Confusion Matrix ของ Test Result มีการวิเคราะห์เพื่อให้เห็นภาพรวมของผลการทดสอบดังต่อไปนี้

สำหรับคลาส "น้อย" โมเดลทำนายถูกต้องใน 29 กรณี และทำนายผิด 24 กรณี

สำหรับคลาส "ปานกลาง" โมเดลทำนายถูกต้องใน 37 กรณี และทำนายผิด 22 กรณี

สำหรับคลาส "มาก" โมเดลทำนายถูกต้องใน 17 กรณี และทำนายผิด 11 กรณี

สำหรับคลาส "ไม่มีอาการ" โมเดลทำนายถูกต้องใน 17 กรณี และทำนายผิด 11 กรณี

สำหรับคลาส "รุนแรง" โมเดลทำนายถูกต้องใน 10 กรณี และทำนายผิด 2 กรณี

โดยในภาพ % สีเขียว คือโมเดลทำนายถูกต้องส่วน % สีแดง คือโมเดลทำนายผิดพลาดดัง

ภาพที่ 4.31

		SVM - Linear					
TARGET \ OUTPUT	Mild	Moderate	Moderately severe	None	Severe	SUM	
Mild	29 54.72% 45.28%	15	4	2	3	53	
Moderate	15	37 62.71% 37.29%	3	1	3	59	
Moderately severe	4	5	17 60.71% 39.29%	1	1	28	
None	3	5	1	17 60.71% 39.29%	2	28	
Severe	1	1	0	0	10 83.33% 16.67%	12	
SUM	52 55.77% 44.23%	63 58.73% 41.27%	25 68.00% 32.00%	21 80.95% 19.05%	19 52.63% 47.37%	110 / 180 61.11% 38.89%	

ภาพที่ 4.31 Confusion Matrix ของ Test Result SVM (Linear) Model (5 Class)

ตารางที่ 4.36 Sensitivity และ Specificity (SVM (Linear-5 Class))

Class	Sensitivity (ความไวต่อการตรวจพบ)	Specificity (ความไวต่อการตรวจปฏิเสธ)
สำหรับคลาส "ไม่มีอาการ"	ค่า Sensitivity คือ 65%	Specificity คือ 83%
สำหรับคลาส "น้อย"	ค่า Sensitivity คือ 55%	Specificity คือ 76%
สำหรับคลาส "ปานกลาง"	ค่า Sensitivity คือ 63%	Specificity คือ 55%
สำหรับคลาส "มาก"	ค่า Sensitivity คือ 74%	Specificity คือ 78%
สำหรับคลาส "รุนแรง"	ค่า Sensitivity คือ 83%	Specificity คือ 77%

สรุปโมเดลดูเหมาะสมในการตรวจคลาส "มาก" และ "ไม่มีอาการ" และค่า Sensitivity สูงสำหรับคลาส "รุนแรง" แต่ค่า Specificity ต่ำกว่าสำหรับคลาส "น้อย" และ "ปานกลาง"

SVM (RBF)

จากข้อมูล Confusion Matrix ของ Test Result มีการวิเคราะห์เพื่อให้เห็นภาพรวมของผลการทดสอบดังต่อไปนี้

สำหรับคลาส "น้อย" โมเดลทำนายถูกต้องใน 49 กรณี และทำนายผิด 4 กรณี

สำหรับคลาส "ปานกลาง" โมเดลทำนายถูกต้องใน 48 กรณี และทำนายผิด 11 กรณี

สำหรับคลาส "มาก" โมเดลทำนายถูกต้องใน 13 กรณี และทำนายผิด 15 กรณี

สำหรับคลาส "ไม่มีอาการ" โมเดลทำนายถูกต้องใน 19 กรณี และทำนายผิด 9 กรณี

สำหรับคลาส "รุนแรง" โมเดลทำนายถูกต้องใน 6 กรณี และทำนายผิด 6 กรณี

โดยในภาพ % สีเขียว คือโมเดลทำนายถูกต้องส่วน % สีแดง คือโมเดลทำนายผิดพลาดดัง

ภาพที่ 4.32

SVM - RBF						
TARGET \ OUTPUT	Mild	Moderate	Moderately severe	None	Severe	SUM
Mild	49	4	0	0	0	53 92.45% 7.55%
Moderate	10	48	0	0	1	59 81.36% 18.64%
Moderately severe	5	10	13	0	0	28 46.43% 53.57%
None	6	3	0	19	0	28 67.86% 32.14%
Severe	2	3	1	0	6	12 50.00% 50.00%
SUM	72 68.06% 31.94%	68 70.59% 29.41%	14 92.86% 7.14%	19 100.00% 0.00%	7 85.71% 14.29%	135 / 180 75.00% 25.00%

ภาพที่ 4.32 Confusion Matrix ของ Test Result SVM (RBF) Model (5 Class)

ตารางที่ 4.37 Sensitivity และ Specificity (SVM (RBF-5 Class))

Class	Sensitivity (ความไวต่อการตรวจพบ)	Specificity (ความไวต่อการตรวจปฏิเสธ)
สำหรับคลาส "ไม่มีอาการ"	ค่า Sensitivity คือ 76%	Specificity คือ 104% (ไม่สามารถคำนวณได้)
สำหรับคลาส "น้อย"	ค่า Sensitivity คือ 66%	Specificity คือ 77%
สำหรับคลาส "ปานกลาง"	ค่า Sensitivity คือ 71%	Specificity คือ 85%
สำหรับคลาส "มาก"	ค่า Sensitivity คือ 54%	Specificity คือ 126% (ไม่สามารถคำนวณได้)
สำหรับคลาส "รุนแรง"	ค่า Sensitivity คือ 86%	Specificity คือ 145% (ไม่สามารถคำนวณได้)

ค่า Specificity มีค่าที่เกิน 100% ในบางคลาส เนื่องจากมีค่า False Positive ที่น้อยกว่าค่า True Negative ใน Confusion Matrix ที่เป็นไปได้ ทำให้ค่า Specificity นั้นเกิน 100% ควรตรวจสอบว่า ข้อมูลหรือการคำนวณอาจมีข้อผิดพลาดหรือไม่ หากข้อมูลถูกต้องและการคำนวณถูกต้องค่า Specificity จะไม่เกิน 100% แต่ในกรณีที่เกิน 100% อาจจะเป็นผลลัพธ์ของข้อมูลที่น้อยลง

ค่า Sensitivity แสดงความสามารถในการตรวจพบคลาสที่มีอยู่จริง ส่วนค่า Specificity แสดงความสามารถในการตรวจไม่พบคลาสที่ไม่มีอยู่จริง ค่า Specificity ที่เกิน 100% อาจเกิดขึ้นได้เมื่อข้อมูลมีจำนวนความผิดพลาด (False Positive และ True Negative) น้อยมากหรือข้อมูลน้อยลง และสัดส่วน False Positive ในแต่ละคลาสสูงกว่า True Negative

Gaussian Naive Bayes

จากข้อมูล Confusion Matrix ของ Test Result มีการวิเคราะห์เพื่อให้เห็นภาพรวมของผลการทดสอบดังต่อไปนี้

สำหรับคลาส "น้อย" โมเดลทำนายถูกต้องใน 12 กรณี และทำนายผิด 41 กรณี

สำหรับคลาส "ปานกลาง" โมเดลทำนายถูกต้องใน 35 กรณี และทำนายผิด 24 กรณี

สำหรับคลาส "มาก" โมเดลทำนายถูกต้องใน 9 กรณี และทำนายผิด 19 กรณี

สำหรับคลาส "ไม่มีอาการ" โมเดลทำนายถูกต้องใน 10 กรณี และทำนายผิด 18 กรณี

สำหรับคลาส "รุนแรง" โมเดลทำนายถูกต้องใน 11 กรณี และทำนายผิด 1 กรณี

โดยในภาพ % สีเขียว คือโมเดลทำนายถูกต้องส่วน % สีแดง คือโมเดลทำนายผิดพลาดดัง

ภาพที่ 4.33

Gaussian Naive Bayes						
TARGET \ OUTPUT	Mild	Moderate	Moderately severe	None	Severe	SUM
Mild	12	18	10	2	11	53 22.64% 77.36%
Moderate	6	35	3	1	14	59 59.32% 40.68%
Moderately severe	1	7	9	2	9	28 32.14% 67.86%
None	0	12	5	10	1	28 35.71% 64.29%
Severe	0	1	0	0	11	12 91.67% 8.33%
SUM	19 63.16% 36.84%	73 47.95% 52.05%	27 33.33% 66.67%	15 66.67% 33.33%	46 23.91% 76.09%	77 / 180 42.78% 57.22%

ภาพที่ 4.33 Confusion Matrix ของ Test Result Gaussian Naive Bayes Model (5 Class)

ตารางที่ 4.38 Sensitivity และ Specificity (Gaussian Naive Bayes-5 Class)

Class	Sensitivity (ความไวต่อการตรวจพบ)	Specificity (ความไวต่อการตรวจปฏิเสธ)
สำหรับคลาส "ไม่มีอาการ"	ค่า Sensitivity คือ 11%	Specificity คือ 20%
สำหรับคลาส "น้อย"	ค่า Sensitivity คือ 63%	Specificity คือ 60%
สำหรับคลาส "ปานกลาง"	ค่า Sensitivity คือ 39%	Specificity คือ 42%
สำหรับคลาส "มาก"	ค่า Sensitivity คือ 13%	Specificity คือ 39%
สำหรับคลาส "รุนแรง"	ค่า Sensitivity คือ 12%	Specificity คือ 16%

สรุปได้ว่าโมเดลมีประสิทธิภาพในการจำแนกคลาส "น้อย" และ "ปานกลาง" ในระดับที่ดีกว่าคลาสอื่น ๆ แต่ควรปรับปรุงเพื่อเพิ่มประสิทธิภาพในการจำแนกคลาส "มาก", "ไม่มีอาการ", และ "รุนแรง" เนื่องจากมีค่า Sensitivity และ Specificity ที่ต่ำกว่ามาก ๆ สำหรับคลาสเหล่านี้

Decision Tree

จากข้อมูล Confusion Matrix ของ Test Result มีการวิเคราะห์เพื่อให้เห็นภาพรวมของผลการทดสอบดังต่อไปนี้

สำหรับคลาส "น้อย" โมเดลทำนายถูกต้องใน 33 กรณี และทำนายผิด 20 กรณี

สำหรับคลาส "ปานกลาง" โมเดลทำนายถูกต้องใน 38 กรณี และทำนายผิด 21 กรณี

สำหรับคลาส "มาก" โมเดลทำนายถูกต้องใน 9 กรณี และทำนายผิด 19 กรณี

สำหรับคลาส "ไม่มีอาการ" โมเดลทำนายถูกต้องใน 19 กรณี และทำนายผิด 9 กรณี

สำหรับคลาส "รุนแรง" โมเดลทำนายถูกต้องใน 8 กรณี และทำนายผิด 4 กรณี

โดยในภาพ % สีเขียว คือโมเดลทำนายถูกต้องส่วน % สีแดง คือโมเดลทำนายผิดพลาดดัง

ภาพที่ 4.34

Decision Tree						
TARGET \ OUTPUT	Mild	Moderate	Moderately severe	None	Severe	SUM
Mild	33 62.26% 37.74%	12	2	5	1	53
Moderate	9	38 64.41% 35.59%	4	3	5	59
Moderately severe	7	10	9 32.14% 67.86%	2	0	28
None	4	2	3	19 67.86% 32.14%	0	28
Severe	2	2	0	0	8 66.67% 33.33%	12
SUM	55 60.00% 40.00%	64 59.38% 40.63%	18 50.00% 50.00%	29 65.52% 34.48%	14 57.14% 42.86%	107 / 180 59.44% 40.56%

ภาพที่ 4.34 Confusion Matrix ของ Test Result Decision Tree Model (5 Class)

ตารางที่ 4.39 Sensitivity และ Specificity (Decision Tree-5 Class)

Class	Sensitivity (ความไวต่อการตรวจพบ)	Specificity (ความไวต่อการตรวจปฏิเสธ)
สำหรับคลาส "ไม่มีอาการ"	ค่า Sensitivity คือ 70%	Specificity คือ 83%
สำหรับคลาส "น้อย"	ค่า Sensitivity คือ 62%	Specificity คือ 65%
สำหรับคลาส "ปานกลาง"	ค่า Sensitivity คือ 51%	Specificity คือ 58%
สำหรับคลาส "มาก"	ค่า Sensitivity คือ 31%	Specificity คือ 86%
สำหรับคลาส "รุนแรง"	ค่า Sensitivity คือ 53%	Specificity คือ 80%

สรุปได้ว่าโมเดลมีประสิทธิภาพในการจำแนกคลาส "ไม่มีอาการ" และ "รุนแรง" ในระดับที่ดีกว่าคลาสอื่น ๆ แต่ควรปรับปรุงเพื่อเพิ่มประสิทธิภาพในการจำแนกคลาส "มาก" และ "ปานกลาง" ซึ่งมีค่า Sensitivity ที่ต่ำกว่าค่า Specificity ในระดับที่น่าปรับปรุงเพิ่มขึ้น

AdaBoost

จากข้อมูล Confusion Matrix ของ Test Result มีการวิเคราะห์เพื่อให้เห็นภาพรวมของผลการทดสอบดังต่อไปนี้

สำหรับคลาส "น้อย" โมเดลทำนายถูกต้องใน 25 กรณี และทำนายผิด 28 กรณี

สำหรับคลาส "ปานกลาง" โมเดลทำนายถูกต้องใน 28 กรณี และทำนายผิด 31 กรณี

สำหรับคลาส "มาก" โมเดลทำนายถูกต้องใน 6 กรณี และทำนายผิด 22 กรณี

สำหรับคลาส "ไม่มีอาการ" โมเดลทำนายถูกต้องใน 6 กรณี และทำนายผิด 22 กรณี

สำหรับคลาส "รุนแรง" โมเดลทำนายถูกต้องใน 5 กรณี และทำนายผิด 7 กรณี

โดยในภาพ % สีเขียว คือโมเดลทำนายถูกต้องส่วน % สีแดง คือโมเดลทำนายผิดพลาดดัง

ภาพที่ 4.35

AdaBoost						
TARGET \ OUTPUT	Mild	Moderate	Moderately severe	None	Severe	SUM
Mild	25	14	3	8	3	53 47.17% 52.83%
Moderate	13	28	5	8	5	59 47.46% 52.54%
Moderately severe	7	9	6	2	4	28 21.43% 78.57%
None	10	4	8	6	0	28 21.43% 78.57%
Severe	2	5	0	0	5	12 41.67% 58.33%
SUM	57 43.86% 56.14%	60 46.67% 53.33%	22 27.27% 72.73%	24 25.00% 75.00%	17 29.41% 70.59%	70 / 180 38.89% 61.11%

ภาพที่ 4.35 Confusion Matrix ของ Test Result AdaBoost Model (5 Class)

ตารางที่ 4.40 Sensitivity และ Specificity (AdaBoost-5 Class)

Class	Sensitivity (ความไวต่อการตรวจพบ)	Specificity (ความไวต่อการตรวจปฏิเสธ)
สำหรับคลาส "ไม่มีอาการ"	ค่า Sensitivity คือ 24%	Specificity คือ 29%
สำหรับคลาส "น้อย"	ค่า Sensitivity คือ 45%	Specificity คือ 52%
สำหรับคลาส "ปานกลาง"	ค่า Sensitivity คือ 35%	Specificity คือ 58%
สำหรับคลาส "มาก"	ค่า Sensitivity คือ 16%	Specificity คือ 12%
สำหรับคลาส "รุนแรง"	ค่า Sensitivity คือ 31%	Specificity คือ 29%

สรุปได้ว่าโมเดลมีประสิทธิภาพในการจำแนกคลาส "ปานกลาง" และคลาส "ไม่มีอาการ" ในระดับที่ดีกว่าคลาสอื่น ๆ แต่ควรปรับปรุงเพื่อเพิ่มประสิทธิภาพในการจำแนกคลาส "น้อย" และ "มาก" ซึ่งมีค่า Sensitivity ที่ต่ำกว่าค่า Specificity ในระดับที่น่าปรับปรุงเพิ่มขึ้น

Stochastic Gradient Descent (SGD)

จากข้อมูล Confusion Matrix ของ Test Result มีการวิเคราะห์เพื่อให้เห็นภาพรวมของผลการทดสอบดังต่อไปนี้

สำหรับคลาส "น้อย" โมเดลทำนายถูกต้องใน 22 กรณี และทำนายผิด 31 กรณี

สำหรับคลาส "ปานกลาง" โมเดลทำนายถูกต้องใน 52 กรณี และทำนายผิด 7 กรณี

สำหรับคลาส "มาก" โมเดลทำนายถูกต้องใน 6 กรณี และทำนายผิด 22 กรณี

สำหรับคลาส "ไม่มีอาการ" โมเดลทำนายถูกต้องใน 5 กรณี และทำนายผิด 23 กรณี

สำหรับคลาส "รุนแรง" โมเดลทำนายถูกต้องใน 0 กรณี และทำนายผิด 12 กรณี

โดยในภาพ % สีเขียว คือโมเดลทำนายถูกต้องส่วน % สีแดง คือโมเดลทำนายผิดพลาดดัง

ภาพที่ 4.36

Stochastic Gradient Descent						
TARGET \ OUTPUT	Mild	Moderate	Moderately severe	None	Severe	SUM
Mild	22 41.51% 58.49%	31	0	0	0	53
Moderate	7	52 88.14% 11.86%	0	0	0	59
Moderately severe	3	19	6 21.43% 78.57%	0	0	28
None	13	9	1	5 17.86% 82.14%	0	28
Severe	0	12	0	0	0 0.00% 100.00%	12
SUM	45 48.89% 51.11%	123 42.28% 57.72%	7 85.71% 14.29%	5 100.00% 0.00%	0 NaN% NaN%	85 / 180 47.22% 52.78%

ภาพที่ 4.36 Confusion Matrix ของ Test Result Stochastic Gradient Descent (SGD) Model (5 Class)

ตารางที่ 4.41 Sensitivity และ Specificity (Stochastic Gradient Descent-5 Class)

Class	Sensitivity (ความไวต่อการตรวจพบ)	Specificity (ความไวต่อการตรวจปฏิเสธ)
สำหรับคลาส "ไม่มีอาการ"	ค่า Sensitivity คือ 14%	Specificity คือ 36%
สำหรับคลาส "น้อย"	ค่า Sensitivity คือ 36%	Specificity คือ 59%
สำหรับคลาส "ปานกลาง"	ค่า Sensitivity คือ 52%	Specificity คือ 59%
สำหรับคลาส "มาก"	ค่า Sensitivity คือ 15%	Specificity คือ 36%
สำหรับคลาส "รุนแรง"	ค่า Sensitivity คือ 0%	Specificity คือ 100%

สรุปได้ว่าโมเดลมีประสิทธิภาพในการจำแนกคลาส "ปานกลาง" และคลาส "น้อย" ในระดับที่ดีกว่าคลาสอื่น ๆ แต่มีประสิทธิภาพในการจำแนกคลาส "มาก" และ "ไม่มีอาการ" ที่ต่ำกว่าค่า Specificity ในระดับที่น่าปรับปรุงเพิ่มขึ้น ส่วนคลาส "รุนแรง" มีค่า Sensitivity ที่ต่ำมากและค่า Specificity เป็น 1.00 เนื่องจากมีจำนวนข้อมูลน้อยในคลาสนี้ แต่อาจไม่สามารถใช้โมเดลนี้ในการจำแนกคลาสนี้ได้ในทางปฏิบัติ เนื่องจากค่า Sensitivity มีค่าต่ำมาก

4.3 การทดลองอื่น ๆ

นอกจากนี้ผู้วิจัยยังมีการทดลองอีกหลายอย่างที่จะแสดงดังต่อไปนี้

4.3.1 เสียงที่ระบุตามเพศ

เสียงที่ระบุตามเพศจะมีความแตกต่างของ Accuracy หรือไม่ โดยเฉพาะเสียงของเพศหญิงจะได้ค่า Accuracy ที่มากกว่าเพศชาย จากการที่ผู้วิจัยได้ศึกษางานวิจัยอื่น ๆ ที่เกี่ยวข้องกับการจำแนกประเภทได้บอกว่าเสียงของเพศหญิงจะได้ค่า Accuracy ที่มากกว่าผู้วิจัยจึงทำการทดลองนี้ขึ้น

ตารางต่อไปนี้เป็นตารางที่แบ่งตามเพศของอาสาสมัคร ชาย และ หญิง ที่ผู้วิจัยใช้ในการทดลอง โดยได้คัดเลือกไฟล์ที่มีจำนวนเท่ากันและผล PHQ-9 มีคะแนนที่เท่ากันเพื่อทดลองว่าเสียงของเพศไหนที่ได้ค่า Accuracy มากที่สุด โดยผู้วิจัยได้เลือกทดลองด้วย Model 2D CNN ที่มีค่า Accuracy สูงสุดของงานวิจัยนี้

ตารางที่ 4.42 ตารางแสดงจำนวนของอาสาสมัครแบ่งตามเพศที่ใช้ในการทดลอง

ประเภท	อาสาสมัคร	จำนวนไฟล์
ชาย	4	34
หญิง	4	34

ตารางที่ 4.43 Sum Result แบ่งตามเพศ

Gender	Sum Result	Training Result	Validation Result	Test Result
Female	accuracy	0.95	0.91	0.77
Male	accuracy	0.49	0.55	0.54

จะเห็นได้ว่าเสียงของเพศหญิงมีค่า Accuracy สูงกว่าเพศชายคือ 77% ต่อ 54% แต่อย่างไรด้วยจำนวนของอาสาสมัครที่ใช้ในการทดลองยังน้อยอยู่จึงไม่อาจการันตีได้ว่าเสียงของเพศใดที่จะได้ผลการทดลองที่ดีที่สุด

4.3.2 การทดลองว่าแบบจำลองการเรียนรู้เชิงลึกจดจำที่เสียงของบุคคลหรือกลุ่มก่อนของบุคคล

ทั้ง 4 Model ของผู้วิจัยทำการทดลองมานั้นทั้งหมดจดจำที่เสียงหรือตัวบุคคลทางผู้วิจัยได้สงสัยในข้อนี้จึงทำการทดลองเพิ่มเติมคือผู้วิจัยได้ออกแบบการทดลองออกเป็น 2 Case เพื่อจะตรวจสอบว่า Model ของผู้วิจัยจับที่เสียงของบุคคลหรือกลุ่มก่อนของบุคคลตัวอย่างเช่น ผู้วิจัยแยก Dataset ออกเป็น 2 Case แล้ว Train Model เพื่อทดสอบว่าค่าของ Accuracy มีการเปลี่ยนแปลงมากน้อยเพียงใดในกรณีไฟล์ที่เท่ากันแต่ปริมาณของอาสาสมัครเพิ่มขึ้น

การทดลองที่ 1 อาสาสมัคร 26 คน 200 ไฟล์ แบ่งเป็น ไม่มีภาวะซึมเศร้า 13 คนและมีภาวะซึมเศร้า 13 คนโดยผู้วิจัยเลือก 1D-CNN, 2D-CNN, LSTM, GRU ในการทดลอง ผลการทดลองมีดังต่อไปนี้

ตารางที่ 4.44 Sum Result ของการทดลองที่ 1

Sum Result	Training Result	Validation Result	Test Result
1D CNN	1.00	0.59	0.65
2D CNN	1.00	0.94	0.90
LSTM	0.52	0.59	0.57
GRU	0.52	0.56	0.50

การทดลองที่ 2 อาสาสมัคร 30 คน 200 ไฟล์ แบ่งเป็น ไม่มีภาวะซึมเศร้า 15 คนและมีภาวะซึมเศร้า 15 คนโดยผู้วิจัยเลือก 1D-CNN, 2D-CNN, LSTM, GRU ในการทดลอง ผลการทดลองมีดังต่อไปนี้

ตารางที่ 4.45 Sum Result ของการทดลองที่ 2

Sum Result	Training Result	Validation Result	Test Result
1D CNN	1.00	0.59	0.60
2D CNN	1.00	0.94	0.95
LSTM	0.52	0.59	0.50
GRU	0.52	0.56	0.50

การทดลองที่ 3 อาสาสมัคร 26 คน 250 ไฟล์ แบ่งเป็น ไม่มีภาวะซึมเศร้า 13 คนและมีภาวะซึมเศร้า 13 คนโดยผู้วิจัยเพิ่มจำนวนไฟล์เพิ่มขึ้น 50 ไฟล์และเลือก 1D-CNN, 2D-CNN, LSTM, GRU ในการทดลอง ผลการทดลองมีดังต่อไปนี้

ตารางที่ 4.46 Sum Result ของการทดลองที่ 3

Sum Result	Training Result	Validation Result	Test Result
1D CNN	0.99	0.53	0.50
2D CNN	1.00	0.97	0.88
LSTM	0.50	0.50	0.50
GRU	0.53	0.53	0.54

การทดลองที่ 4 อาสาสมัคร 30 คน 250 ไฟล์ แบ่งเป็น ไม่มีภาวะซึมเศร้า 15 คนและมีภาวะซึมเศร้า 15 คนโดยผู้วิจัยเพิ่มจำนวนไฟล์เพิ่มขึ้น 50 ไฟล์และเลือก 1D-CNN, 2D-CNN, LSTM, GRU ในการทดลอง ผลการทดลองมีดังต่อไปนี้

ตารางที่ 4.47 Sum Result ของการทดลองที่ 4

Sum Result	Training Result	Validation Result	Test Result
1D CNN	1.00	0.57	0.58
2D CNN	1.00	0.97	0.88
LSTM	0.50	0.50	0.50
GRU	0.49	0.42	0.54

การทดลองที่ 5 อาสาสมัคร 26 คน 300 ไฟล์ แบ่งเป็น ไม่มีภาวะซึมเศร้า 13 คนและมีภาวะซึมเศร้า 13 คนโดยผู้วิจัยเพิ่มจำนวนไฟล์เพิ่มขึ้น 100 ไฟล์และเลือก 1D-CNN, 2D-CNN, LSTM, GRU ในการทดลอง ผลการทดลองมีดังต่อไปนี้

ตารางที่ 4.48 Sum Result ของการทดลองที่ 5

Sum Result	Training Result	Validation Result	Test Result
1D CNN	1.00	0.56	0.52
2D CNN	1.00	0.98	0.92
LSTM	0.60	0.54	0.65
GRU	0.77	0.70	0.70

การทดลองที่ 6 อาสาสมัคร 30 คน 300 ไฟล์ แบ่งเป็น ไม่มีภาวะซึมเศร้า 15 คนและมีภาวะซึมเศร้า 15 คนโดยผู้วิจัยเพิ่มจำนวนไฟล์เพิ่มขึ้น 100 ไฟล์และเลือก 1D-CNN, 2D-CNN, LSTM, GRU ในการทดลอง ผลการทดลองมีดังต่อไปนี้

ตารางที่ 4.49 Sum Result ของการทดลองที่ 6

Sum Result	Training Result	Validation Result	Test Result
1D CNN	1.00	0.65	0.62
2D CNN	1.00	0.98	0.90
LSTM	0.61	0.54	0.55
GRU	0.70	0.48	0.57

โดยมีผลสรุปของทั้ง 6 การทดลองดังต่อไปนี้

ตารางที่ 4.50 สรุปผลการทดลองทั้ง 6 การทดลอง

Type 1 (2 Class 200 FILE)			Type 2 (2 Class 250 FILE)			Type 3 (2 Class 300 FILE)		
Model	Case 1	Case 2	Model	Case 1	Case 2	Model	Case 1	Case 2
1D CNN	0.65	0.60	1D CNN	0.50	0.58	1D CNN	0.52	0.62
2D CNN	0.90	0.95	2D CNN	0.88	0.88	2D CNN	0.92	0.90
LSTM	0.57	0.50	LSTM	0.50	0.50	LSTM	0.65	0.55
GRU	0.50	0.50	GRU	0.54	0.54	GRU	0.70	0.57

จากการทดลองค่า Accuracy ก็ยังมีความใกล้เคียงกันไม่แตกต่างกันมากเมื่อเปรียบเทียบเป็น Case by Case ก็จะปรากฏว่าบาง model คนน้อยกว่าแต่มี Accuracy มากกว่าคนเยอะกว่าหรือกลับกัน คนเยอะกว่าก็มีค่า Accuracy มากกว่าคนน้อยกว่าเช่นกัน Accuracy ก็ไม่ค่าแตกต่างสลบ ๆ กัน ผู้วิจัย จึงใช้ Type 1 ใน Model 2D CNN ที่มีค่า accuracy สูงที่สุดของ 3 Type เป็นเกณฑ์ตัดสินเบื้องต้น ว่า Case 1 มีค่า Accuracy อยู่ที่ 90% แต่ Case 2 มีค่า Accuracy อยู่ที่ 95% โดย Case 1 มี อาสาสมัครน้อยกว่า Case 2 ดังนั้น model วิเคราะห์จากกลุ่มของอาสาสมัครมากกว่าไฟล์เสียงของ อาสาสมัคร

4.3.3 การผสม Dataset ระหว่างภาษาอังกฤษและภาษาไทย

โดยผู้วิจัยได้ทำการทดลองนำเสียงของอาสาสมัครที่เป็นภาษาไทยสามารถรวมเข้ากับข้อมูล อาสาสมัครที่เป็นภาษาอังกฤษได้หรือไม่ โดยผู้วิจัยได้รวม Dataset ที่เป็นภาษาไทยของผู้วิจัย (Dataset PHQ-9) และ Dataset ที่เป็นภาษาอังกฤษที่ได้รับการรับรองจาก University of Southern California ที่ เผยแพร่ไปทั่วโลก (Dataset DAIC-WOZ) โดยมีรายละเอียดดังนี้

Dataset PHQ-9 คือฐานข้อมูลภาพและเสียงของอาสาสมัครที่ทำแบบทดสอบ PHQ-9 ที่ทาง ผู้วิจัยได้ทำการเก็บข้อมูลเองในรูปแบบภาษาไทย

Dataset DAIC-WOZ คือฐานข้อมูลนี้ประกอบด้วยการสัมภาษณ์ทางคลินิกที่ออกแบบมาเพื่อ สนับสนุนการวินิจฉัยสภาวะความทุกข์ทางจิต เช่น ความวิตกกังวล อาการซึมเศร้า และโรคความเครียด ภายหลังเหตุการณ์สะเทือนใจ ข้อมูลที่รวบรวมได้แก่ การบันทึกเสียงและวิดีโอและการตอบแบบสอบถาม ที่ครอบคลุมในรูปแบบภาษาไทยรูปแบบภาษาอังกฤษ

อาสาสมัครของ PHQ-9 คือ 30 ท่านแบ่งเป็น 300 ไฟล์เสียงส่วนอาสาสมัครของ DAIC-WOZ คือ 70 ท่านแบ่งเป็น 210 ไฟล์เสียงดังตารางด้านล่าง

ตารางที่ 4.51 รวมฐานข้อมูล PHQ-9 และ Daic-Woz

ฐานข้อมูล	ประเภท	อาสาสมัคร	จำนวนไฟล์
PHQ-9	ไม่มีภาวะซึมเศร้า	15	150
	มีภาวะซึมเศร้า	15	150
DAIC-WOZ	ไม่มีภาวะซึมเศร้า	35	105
	มีภาวะซึมเศร้า	35	105

โดยผู้วิจัยทำการ Training Dataset ด้วยการรวม Dataset ภาษาไทยและภาษาอังกฤษเข้าด้วยกันแต่แบ่ง Unseen Test set ออกเป็น 3 รูปแบบคือ 1. รวม Dataset ภาษาไทยและภาษาอังกฤษ (Dataset PHQ-9 + DAIC-WOZ) 2. เฉพาะภาษาไทยอย่างเดียว (Dataset PHQ-9) 3. เฉพาะภาษาอังกฤษอย่างเดียว (Dataset DAIC-WOZ)

ผลการทดลองโดยผู้วิจัยเลือก 1D-CNN, 2D-CNN, LSTM, GRU เป็น model ในการทดลองครั้งนี้

ตารางที่ 4.52 ผลการทดลองการรวมฐานข้อมูลรูปแบบที่ 1 รวม Dataset ภาษาไทยและภาษาอังกฤษ (Dataset PHQ-9 + Dataset DAIC-WOZ)

Sum Result	Training Result	Validation Result	Test Result
1D CNN	0.96	0.56	0.49
2D CNN	1.00	0.59	0.71
LSTM	0.61	0.55	0.50
GRU	0.67	0.60	0.53

ตารางที่ 4.53 ผลการทดลองการรวมฐานข้อมูลรูปแบบที่ 2 เฉพาะภาษาไทยอย่างเดียว (Dataset PHQ-9)

Sum Result	Training Result	Validation Result	Test Result
1D CNN	1.00	0.61	0.52
2D CNN	1.00	0.72	0.65
LSTM	0.59	0.52	0.63
GRU	0.71	0.52	0.50

ตารางที่ 4.54 ผลการทดลองการรวมฐานข้อมูลรูปแบบที่ 3 เฉพาะภาษาอังกฤษอย่างเดียว (Dataset DAIC-WOZ)

Sum Result	Training Result	Validation Result	Test Result
1D CNN	1.00	0.52	0.64
2D CNN	1.00	0.63	0.79
LSTM	0.56	0.54	0.55
GRU	0.61	0.56	0.60

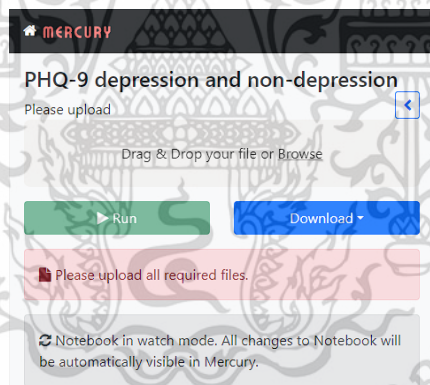
จากการทดลองสามารถจะแสดงให้เห็นว่าการรวมฐานข้อมูล PHQ-9 ที่เป็นภาษาไทย และ Daic-Woz ที่เป็นภาษาอังกฤษสามารถร่วมเข้ากันได้และ 2D CNN มีค่า accuracy ที่สูงที่สุดทั้ง 3 รูปแบบและรวมถึงมากที่สุด ใน 4 โมเดล

4.4 ส่วนต่อประสานกับผู้ใช้ (User Interface)

ผู้วิจัยพัฒนาเว็บไซต์ส่วนต่อประสานกับผู้ใช้ (User Interface) ที่นำ Model ที่มีค่า Accuracy ที่สุดของการทดลองในงานวิจัยนี้โดยการนำไปใช้งานจริง ได้แก่ 2D CNN ในประเภทที่ 1 การจำแนกประเภทระหว่างมีภาวะซึมเศร้าและไม่มีภาวะซึมเศร้าที่กล่าวไปข้างต้นโดยที่ค่า accuracy อยู่ที่ 95% โดยเว็บไซต์มีรายละเอียดดังนี้

4.4.1 Link: <http://161.246.127.201/app/1>

ผู้ใช้งานต้อง Upload ไฟล์เสียงโดยการกดปุ่ม Browse เพื่อ upload ไฟล์เสียงจากนั้นผู้ใช้งานต้องกดปุ่ม Run ให้ทางระบบวิเคราะห์เสียงของผู้ใช้งานว่าเป็นมีภาวะซึมเศร้าหรือไม่มีภาวะซึมเศร้า



ภาพที่ 4.37 ตัวอย่างเว็บไซต์แยกแยะเสียงผู้ที่มีภาวะซึมเศร้าหรือไม่มีภาวะซึมเศร้า

จากนั้นเว็บไซต์จะแสดงผลของผู้ใช้งานว่ามีภาวะซึมเศร้าหรือไม่มีภาวะซึมเศร้าดังภาพที่แสดงด้านล่างตรง Predicted Class

```
In [23]: # Make predictions
preds = loaded_model.predict(mfcc)
x_labels = ['depression', 'non-depression']
pred_class = x_labels[np.argmax(tf.nn.softmax(preds))]

print('Predicted class:', pred_class)

Predicted class: non-depression
```

```
In [24]: # Audio
ipd.Audio(path2)
```



ภาพที่ 4.38 ตัวอย่างผลที่แสดงในเว็บไซต์แยกแยะเสียงผู้ที่มีภาวะซึมเศร้าหรือไม่มีภาวะซึมเศร้า

ภาพที่ 4.39 ตัวอย่างภาพรวมเว็บไซต์แยกแยะเสียงผู้ที่มีภาวะซึมเศร้าหรือไม่มีภาวะซึมเศร้า

บทที่ 5

สรุปผลการศึกษา

5.1 สรุปผลของโมเดลแบบจำลองการเรียนรู้เชิงลึกและแบบจำลองการเรียนรู้ของเครื่อง

5.1.1 ประเภทที่ 1 การจำแนกประเภทระหว่างมีภาวะซึมเศร้าและไม่มีภาวะซึมเศร้า

แบบจำลองการเรียนรู้เชิงลึก

1. จากการทดลองจะเห็นได้ว่า 2D CNN มีค่าความแม่นยำที่สูงที่สุดคือ 95% จากทั้งหมด 4 โมเดลโดยลำดับที่ 2 คือ 1D CNN ลำดับที่ 3 คือ GRU และลำดับสุดท้าย LSTM

2. Dataset ที่ผู้วิจัยเก็บรวบรวมข้อมูลมีความเข้ากันกับ 2D CNN มากกว่า Model อื่น ๆ ที่ได้ทำการทดลอง

3. LSTM และ GRU มีคะแนนที่ค่อนข้างน้อยและเมื่อสังเกตลงไป Precision, Recall, F1-score จะเห็นได้ว่าไม่มีภาวะซึมเศร้ามีคะแนนดีกว่ามีภาวะซึมเศร้า

4. Model ที่ผู้วิจัยใช้งานนั้นสามารถจำแนกประเภทระหว่างผู้มีภาวะซึมเศร้าและไม่มีภาวะซึมเศร้าได้จากน้ำเสียงที่แปลงสัญญาณเป็น MFCC

แบบจำลองการเรียนรู้ของเครื่อง

1. จากการทดลองจะเห็นได้ว่า KNN, Random Forest และ MLP มีค่าความแม่นยำที่สูงที่สุดคือ 95% จากทั้งหมด 10 โมเดลโดยลำดับที่ 2 คือ SVM (Linear) ส่วนโมเดลที่ได้ค่าความแม่นยำต่ำที่สุดคือ Decision Tree อยู่ที่ 77%

4. Dataset ที่ผู้วิจัยเก็บรวบรวมข้อมูลมีความเข้ากันกับ KNN, Random Forest และ MLP มากกว่า Model อื่น ๆ ที่ได้ทำการทดลองจึงทำให้ค่าความแม่นยำออกมาสูง

5. Decision Tree มีค่าความแม่นยำที่น้อยที่สุดแต่ก็อยู่ในมาตรฐานที่รับได้

6. Model ที่ผู้วิจัยใช้งานนั้นสามารถจำแนกประเภทระหว่างผู้มีภาวะซึมเศร้าและไม่มีภาวะซึมเศร้าได้จากน้ำเสียงที่แปลงสัญญาณเป็น MFCC

5.1.2 ประเภทที่ 2 การจำแนกประเภทที่แบ่งออกเป็น 5 ประเภทตามเกณฑ์ของ PHQ-9 ได้แก่ 1. ไม่มีอาการ 2. น้อย 3. ปานกลาง 4. มาก 5. รุนแรง สรุปผลการทดลองได้ ดังนี้

แบบจำลองการเรียนรู้เชิงลึก

1. จากการทดลองจะเห็นได้ว่า 2D CNN มีค่าความแม่นยำที่สูงที่สุดคือ 65% จากทั้งหมด 4 โมเดลโดยลำดับที่ 2 คือ LSTM ลำดับที่ 3 คือ 1D CNN และลำดับสุดท้าย GRU

2. ถ้าดูจาก Confusion Matrix ในบทที่ 4 เห็นได้ว่าระดับของปานกลางและน้อยมีผลที่หายถูกมากที่สุดเพราะทั้งสองระดับมีจำนวนของข้อมูลเสียงที่เยอะกว่าระดับอื่น

3. Model ที่ผู้วิจัยใช้งานนั้นสามารถจำแนกประเภทระหว่าง 1. ไม่มีอาการ 2. น้อย 3. ปานกลาง 4. มาก 5. รุนแรง ได้จากน้ำเสียงที่แปลงสัญญาณเป็น MFCC

แบบจำลองการเรียนรู้ของเครื่อง

1. จากการทดลองจะเห็นได้ว่า KNN, Random Forest และ MLP มีค่าความแม่นยำที่สูงเป็น 3 อันดับแรกจากทั้งหมด

4. Dataset ที่ผู้วิจัยเก็บรวบรวมข้อมูลมีความเข้ากันกับ KNN, Random Forest และ MLP มากกว่า Model อื่น ๆ ที่ได้ทำการทดลองจึงทำให้ค่าความแม่นยำออกมาสูง

5. AdaBoost, SGD มีค่าความแม่นยำที่น้อยที่สุดแต่ก็อยู่ในมาตรฐานที่รับได้

6. Model ที่ผู้วิจัยใช้งานนั้นสามารถจำแนกประเภทระหว่าง 1. ไม่มีอาการ 2. น้อย 3. ปานกลาง 4. มาก 5. รุนแรง ได้จากน้ำเสียงที่แปลงสัญญาณเป็น MFCC

5.1.3 สรุป Model นั้นจดจำที่เสียงหรือตัวบุคคลจากการจำแนกประเภทระหว่างมีภาวะซึมเศร้าและไม่มีภาวะซึมเศร้า

ตารางที่ 5.1 ค่าความแม่นยำของ Case 1 ที่มีอาสาสมัคร 26 คนและ Case 2 ที่มีอาสาสมัคร 30 คน

Architecture	Test Result	Case 1 (จำนวนอาสาสมัคร 26 ท่าน)	Case 2 (จำนวนอาสาสมัคร 30 ท่าน)
1D CNN		0.52	0.62
2D CNN		0.92	0.90
LSTM		0.65	0.55
GRU		0.70	0.57

โดยผู้วิจัยได้แบ่งการทดลองออกเป็น 2 Case และในแต่ละ Case จะมีไฟล์เสียงของอาสาสมัครเท่ากันที่ 300 ไฟล์แต่จำนวนของอาสาสมัครไม่เท่ากัน โดยรายละเอียดมีดังต่อไปนี้

Case 1 คืออาสาสมัครทั้งหมด 26 ท่าน แบ่งออกเป็น มีภาวะซึมเศร้าทั้งหมด 13 ท่าน จำนวนไฟล์ทั้งหมด 150 ไฟล์ และไม่มีภาวะซึมเศร้าจำนวนไฟล์ทั้งหมด 150 ไฟล์

Case 2 คืออาสาสมัครทั้งหมด 30 ท่าน แบ่งออกเป็น มีภาวะซึมเศร้าทั้งหมด 13 ท่าน จำนวนไฟล์ทั้งหมด 150 ไฟล์ และไม่มีภาวะซึมเศร้าจำนวนไฟล์ทั้งหมด 150 ไฟล์

จากการทดลองค่าความแม่นยำก็ยังคงมีความใกล้เคียงกันไม่แตกต่างกันมากเมื่อเปรียบเทียบทั้งสอง Case โมเดล 2D CNN, LSTM, GRU ของ Case 1 ที่มีจำนวนอาสาสมัคร 26 คนมีค่าความ

แม่นยำมากกว่า Case 2 ที่มีจำนวนอาสาสมัคร 30 คนโดยทั้ง 2 Case มีไฟล์ของเสียงที่เท่ากันคือ 300 ไฟล์เสียง ดังนั้นจากผลของค่าความแม่นยำผู้วิจัยจึงสรุปได้ว่าโมเดลแยกสัญญาณของเสียงไม่ได้แยกที่ตัวบุคคลเพราะถ้าแยกที่ตัวบุคคล Case 2 ต้องได้ค่าความแม่นยำที่มากกว่า Case 1 แต่ยังมี 1D CNN ที่ Case 2 มากกว่า Case 1 จึงยังสรุปแบบ 100% ไม่ได้ว่าโมเดลของผู้วิจัยจำแนกประเภทระหว่างมีภาวะซึมเศร้าและไม่มีภาวะซึมเศร้าด้วยสัญญาณของเสียงหรือตัวของบุคคลคิดเป็น 75% ตามที่สรุปด้านบน

5.1.4 สรุปเพศมีผลต่อค่าความแม่นยำหรือไม่

ตารางที่ 5.2 ค่าความแม่นยำของ Case 1 ที่มีอาสาสมัคร 26 คนและ Case 2 ที่มีอาสาสมัคร 30 คน

Architecture	Test	เพศหญิง	เพศชาย
1D CNN	Result	0.54	0.77
2D CNN		0.77	0.54
LSTM		0.54	0.46
GRU		0.54	0.46

จากการทดลองด้วยการแบ่งข้อมูลเพศหญิงและเพศชายแล้วค่าความแม่นยำ ก็ยังคงมีความใกล้เคียงกันไม่แตกต่างกันมาก โมเดล 2D CNN, LSTM, GRU ให้เพศหญิงมีค่าความแม่นยำสูงกว่าเพศชายแต่ 1D CNN ให้เพศชายมีค่าความแม่นยำสูงกว่าเพศหญิง ดังนั้นสรุปได้ว่าโมเดลของผู้วิจัยนี้ เรื่องของเพศไม่สามารถวัดคุณภาพได้แบบ 100% คิดเป็น 75% และเนื่องจาก dataset ที่แบ่งตามเพศหญิงและชายที่ Balance Dataset ยังมีน้อยเกินไปนั่นเองจึงไม่สามารถวัดผลได้แบบ 100%

5.1.5 สรุปการรวม Dataset PHQ-9 และ Dataset DAIC-WOZ

ตารางที่ 5.3 ผลการทดลองการรวมฐานข้อมูล PHQ-9 และ Daic-Woz ของทั้ง 3 รูปแบบ

Sum Result	รูปแบบที่ 1	รูปแบบที่ 2	รูปแบบที่ 3
1D CNN	0.49	0.52	0.64
2D CNN	0.71	0.65	0.79
LSTM	0.50	0.63	0.55
GRU	0.53	0.50	0.60

โดยผู้วิจัยทำการ Training Dataset ด้วยการรวม Dataset ภาษาไทยและภาษาอังกฤษเข้าด้วยกันแต่แบ่ง Unseen Test set ออกเป็น 3 รูปแบบคือ 1. รวม Dataset ภาษาไทยและภาษาอังกฤษ (Dataset PHQ-9 + DAIC-WOZ) 2. เฉพาะภาษาไทยอย่างเดียว (Dataset PHQ-9) 3. เฉพาะภาษาอังกฤษอย่างเดียว (Dataset DAIC-WOZ) โดยสรุปได้ดังนี้

1. 2D CNN มีค่าความแม่นยำสูงที่สุดทั้ง 3 รูปแบบ
2. รูปแบบที่ 3 ที่เป็น เฉพาะภาษาอังกฤษอย่างเดียว (Dataset DAIC-WOZ) มีค่าความแม่นยำสูงที่สุดคือ 79% (นับเฉพาะ 2D CNN)
3. รูปแบบที่ 2 ที่เป็น เฉพาะภาษาไทยอย่างเดียว (Dataset PHQ-9) มีค่าความแม่นยำน้อยที่สุดคือ 65% (นับเฉพาะ 2D CNN)
4. รูปแบบที่ 1 รวม Dataset ภาษาไทยและภาษาอังกฤษ (Dataset PHQ-9 + DAIC-WOZ) มีค่าความแม่นยำอยู่ที่ 71%
5. จากการทดลองสามารถจะแสดงให้เห็นว่าการรวมฐานข้อมูล PHQ-9 ที่เป็นภาษาไทย และ Daic-Woz ที่เป็นภาษาอังกฤษสามารถรวมเข้ากันได้
6. จากการวิเคราะห์ค่าความแม่นยำจะแสดงให้เห็นว่าโมเดลมีความเข้ากันกับ Dataset ที่เป็นภาษาอังกฤษมากกว่า Dataset ที่เป็นภาษาไทย

5.2 สรุปผลการวิจัย

1. ผู้วิจัยสามารถพัฒนาระบบหรือเว็บไซต์ที่สามารถเก็บข้อมูลของอาสาสมัครตามแบบทดสอบของ PHQ-9 ได้
2. จำนวนอาสาสมัครที่เก็บได้คือ 100 คน แบ่งเป็น Online 50 คน และ Onsite (เก็บที่รามาริบัติ) 50 คน ได้จำนวนไฟล์เสียง 1,000 ไฟล์
3. เสียงของมนุษย์ที่แปลงเป็น MFCC สามารถจำแนกผู้มีภาวะซึมเศร้าและไม่มีภาวะซึมเศร้าได้และสามารถจำแนก 5 กลุ่มตามเกณฑ์ผลการประเมิน PHQ-9 ได้แก่ 1. ไม่มีอาการซึมเศร้า 2. น้อย 3. ปานกลาง 4. มาก 5. รุนแรง ได้
4. การทดลองที่แบ่งข้อมูลออกเป็น 2 Class (depression, non- depression) Model ที่ค่าความแม่นยำสูงที่สุดคือ 2D CNN, KNN, MLP และส่วนของ 5 Class (None, Mild, Moderate, Moderately Severe, Severe) Model ที่ค่าความแม่นยำสูงที่สุดคือ 2D CNN, MLP
5. เสียงที่เป็นภาษาไทยสามารถจำแนกผู้มีภาวะซึมเศร้าและไม่มีภาวะซึมเศร้าได้และสามารถจำแนก 5 กลุ่มตามเกณฑ์ผลการประเมิน PHQ-9 ได้แก่ 1. ไม่มีอาการซึมเศร้า 2. น้อย 3. ปานกลาง 4. มาก 5. รุนแรง ได้

6. สุดท้ายการตัดสินใจว่าบุคคลคนนี้จะ เป็นโรคซึมเศร้าหรือไม่เป็นโรคซึมเศร้าต้องใช้ดุลย พินิจของแพทย์ โดยเครื่องมือต่าง ๆ ที่ได้รับการพัฒนาขึ้นมากมายยังไม่สามารถฟันธงได้แบบ 100%

7. เสียงที่เป็นภาษาไทยและเสียงภาษาอังกฤษสามารถใช้ร่วมกันได้

8. นอกจากนี้ ผู้วิจัยได้สรุปผลประสิทธิภาพของแบบจำลองเพื่อเปรียบเทียบกับเอกสาร เผยแพร่อื่น ๆ ที่จำแนกผู้มีภาวะซึมเศร้าและไม่มีภาวะซึมเศร่าดังแสดงในตารางที่ 5.5 แสดงการ เปรียบเทียบสถาปัตยกรรมแบบจำลองโดยใช้งานที่เกี่ยวข้องและวิธีการของผู้วิจัยกับ CNN (1D และ 2D) และ RNN (LSTM และ GRU)

ตารางที่ 5.4 สรุปประสิทธิภาพของโมเดล CNN, LSTM, GRU สำหรับชุดข้อมูล SOUND WAVE

No.	Authors	Methodology	Model Architectures (Features Processing)	Accuracy
1	Chlasta, K., et al. 2019 [7]	Automated speech-based screening of depression	CNN Model (Spectrogram)	77%
2	Kim, A. Y., et al. (2023) [8]	Automatic Depression Detection Using Smartphone Based Text Dependent Speech Signals Deep Convolutional Neural Network Approach	(knn) MFCC	63%
			(svm) MFCC	59%
3	Seneviratne, N., et al. 2021 [29]	Speech-based depression of severity level classification using a multi-stage dilated CNN-LSTM model	CNN Model (MFCC)	42.76%
			LSTM (MFCC)	45.71%
4	Srimadhur, N., et al. 2020 [30]	An End-to-End model for the detection and assessment of depression levels using speech	CNN Model (Spectrogram)	59.20%
			End-to-end CNN Model (Spectrogram)	61.32%
5	Our Proposed Model	Depression screening system using data-driven audio and video by deep learning model	1D CNN Model (MFCC)	70%
			2D CNN Model (MFCC)	95%
			LSTM Model (MFCC)	48%
			GRU Model (MFCC)	50%

5.7.8 งานวิจัยนี้สามารถพัฒนาส่วนต่อประสาน (User Interface) กับผู้ใช้งานที่สามารถ แยกแยะได้ว่าเสียงที่อัดโหลด ถูกจำแนกอยู่ในกลุ่มไหนระหว่าง Depression หรือ Non-Depression โดยใช้ Model 2D-CNN เป็น Model หลัก

5.3 ข้อจำกัด

1. จำนวนของอาสาสมัคร
2. ระยะเวลาของการเก็บข้อมูลอาสาสมัคร
3. ความร่วมมือของอาสาสมัคร

5.4 ข้อเสนอแนะ

เนื่องจากค่าความแม่นยำของข้อมูลประเภทที่ 2 การจำแนกประเภทที่แบ่งออกเป็น 5 ประเภทตามเกณฑ์ของ PHQ-9 ได้แก่ 1. ไม่มีอาการ 2. น้อย 3. ปานกลาง 4. มาก 5. รุนแรง มีความแม่นยำค่อนข้างน้อยจึงต้องเพิ่มค่าความแม่นยำของโมเดลมากขึ้นและสามารถทำได้โดยดังนี้

1. ควรเพิ่มจำนวนอาสาสมัคร ของในแต่ละระดับให้มากขึ้น เพื่อให้ฐานข้อมูลในการสอนโมเดลมากขึ้นและทำให้โมเดลรู้จักข้อมูลที่มีความหลากหลาย
2. ลองใช้โมเดลอื่น ๆ นอกจาก 1D CNN, 2D CNN, LSTM, GRU
3. ลองใช้โมเดลอื่น ๆ นอกจาก SVM (RBF), SVM (Linear), Random Forest, Gaussian Naive Bayes, Logistic Regression, Decision Tree, AdaBoost, SGD (Stochastic Gradient Descent), MLP (Multi-Layer Perceptron), KNN (K Nearest Neighbour)
4. ทำการปรับค่าตัวแปรที่ใช้สำหรับการสอนโมเดลเพื่อได้โมเดลที่เหมาะสมที่สุด
5. ในส่วนของเว็บไซต์ก็มีข้อเสนอแนะจากผู้ใช้งานว่าควรที่จะใช้งานให้หลากหลาย Device มากกว่านี้เพราะตอนนี้สามารถใช้งานเว็บไซต์ได้เฉพาะ PC หรือ Notebook เท่านั้น ควรจะสามารถใช้งานในโทรศัพท์มือถือหรือแท็บเล็ตได้ด้วย และสุดท้ายควรทำให้เว็บไซต์ใช้งานได้ง่ายขึ้น

บรรณานุกรม

- [1] K. Kroenke, L. Robert, B. Janet, "The Patient Health Questionnaire-2: validity of a two-item depression screener," *Med. Care*, vol. 41, 2003, pp. 1284-1292.
- [2] K. Kroenke, W. Strine Tara, L. Robert Spitzer, B. W. Janet Williams, T. Joyce Berry, H. Ali Mokdad, "The PHQ-8 as a measure of current depression in the general population," *Journal of affective disorders*, val 114, 2009, pp. 163-173.
- [3] P. M. Lewinsohn, J. R. Seeley, R. E. Roberts, N. B. Allen, "Center for Epidemiologic Studies Depression Scale (CES-D) as a screening instrument for depression among community-residing older adults," *Psychol. Aging*, vol. 2, 1997, pp. 277-287.
- [4] Kroenke, K., et al. (2001). "The PHQ-9: validity of a brief depression severity measure." *J Gen Intern Med* 16(9): 606-613.
- [5] Mitra, V., et al. (2016). Noise and Reverberation Effects on Depression Detection from Speech.
- [6] Seneviratne, N. and C. Espy-Wilson (2021). Speech based Depression Severity Level Classification Using a Multi-Stage Dilated CNN-LSTM Model.
- [7] K. Chlasta, W. Krzysztof, K. Izabela, "Automated speech-based screening of depression using deep convolutional neural networks," *Procedia. Computer. Science*, vol. 164, 2019, pp. 618-628.
- [8] Kim, A. Y., et al. (2023). "Automatic Depression Detection Using Smartphone-Based Text-Dependent Speech Signals: Deep Convolutional Neural Network Approach." *J Med Internet Res* 25: e34474.
- [9] Hossan, M., et al. (2011). A novel approach for MFCC feature extraction.
- [10] woraphu, n. (2022). "ปัญญาประดิษฐ์ (AI: Artificial Intelligence) unpublished.
- [11] Institute, D. I. a. G. (2022). "Machine Learning คืออะไร?" unpublished.
- [12] Nielsen, M. (2015). "Neural Networks and Deep Learning Determination Press" unpublished.

- [13] Chan, W., et al. (2016). Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).
- [14] Chung, C. T., et al. (2015). Enhancing automatically discovered multi-level acoustic patterns considering context consistency with applications in spoken term detection. 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).
- [15] Hershey, J. R., et al. (2016). Deep clustering: Discriminative embeddings for segmentation and separation. 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).
- [16] Hinton, G., et al. (2012). "Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups." Signal Processing Magazine, IEEE 29: 82-97.
- [17] Schmitt, M. and B. Schuller (2018). Deep Recurrent Neural Networks for Emotion Recognition in Speech.
- [18] Amazon Web Services, I. ห. ส. (2023). "ตีปเลิร์นนิงคืออะไร." unpublished.
- [19] Shahriar, N. (2023). "What is Convolutional Neural Network — CNN (Deep Learning)." unpublished.
- [20] Kiranyaz, S., et al. (2021). "1D convolutional neural networks and applications: A survey." Mechanical Systems and Signal Processing 151: 107398.
- [21] algoaddict (2019). "ว่าด้วย Recurrent Neural Networks: Part 1." unpublished.
- [22] Smitsomboon, S. (2020). "สรุปความเข้าใจ RNN, LSTM, GRU." unpublished.
- [23] Dobilas, S. (2022). "GRU Recurrent Neural Networks — A Smart Way to Predict Sequences in Python." unpublished.
- [24] Srivastava, T. (2018). "A Complete Guide to K-Nearest Neighbors." unpublished.
- [25] Saini, A. (2021). "Guide on Support Vector Machine (SVM) Algorithm." unpublished.
- [26] TIBCO (2021). "What is a Random Forest?". unpublished.
- [27] javatpoint.com (2020). "Decision Tree Classification Algorithm." unpublished.

- [28] Camp, D. B. (2022). "What is the Naive Bayes Algorithm?". unpublished.
- [29] Kumar, A. (2023). "Logistic Regression Concepts, Python Example." unpublished.
- [30] Jaichuen, S. (2019). "AdaBoost Algorithm." unpublished.
- [31] MLTut.com (2021). "Stochastic Gradient Descent- A Super Easy Complete Guide!". unpublished.
- [32] PEIXOTO, F. (2020). "A Simple overview of Multilayer Perceptron (MLP)." unpublished.
- [33] Promrit, N. (2021). "Evaluation Metrics for Classification Model." unpublished.
- [34] Narkhede, S. (2018). "Understanding Confusion Matrix." unpublished.
- [35] B, H. N. (2019). "Confusion Matrix, Accuracy, Precision, Recall, F1 Score." unpublished.
- [36] Gustana (2022). "Machine Learning: Classification (Positive & Negative Class)." unpublished.
- [37] หวังศุภชาติ, ร. พ. ส. (2012). "คุณสมบัติเฉพาะของเครื่องมือ." unpublished.
- [38] Veeravattanayothin, T. (2022). "FRONT END , BACK END , FULL STACK คืออะไร มีความแตกต่างกันอย่างไรบ้าง ?." unpublished.
- [39] O.T.Chom, M. W. (2023). "PHP คืออะไร? มาทำความรู้จักภาษาคอมพิวเตอร์ที่ได้รับความนิยม." unpublished.
- [40] Veeravattanayothin, T. (2022). "NodeJS คือ อะไร ? มาทำความรู้จักตัวช่วยพัฒนาเว็บไซต์ยอดนิยม!". unpublished.
- [41] TH, Z. U. (2023). "API คืออะไร? ทำไมถึงฮิตขึ้นทุกวัน? อธิบายแบบเข้าใจง่าย." unpublished.
- [42] Cristianini, N. and E. Ricci (2008). Support Vector Machines. Encyclopedia of Algorithms. M.-Y. Kao. Boston, MA, Springer US: 928-932.
- [43] Breiman, L. (2001). "Random Forests." Machine Learning 45(1): 5-32.
- [44] Poolpol, P. (2021). "Naïve Bayes Classification." unpublished.
- [45] (1995). Logistic regression. Reading and understanding multivariate statistics. Washington, DC, US, American Psychological Association: 217-244.

- [46] Fürnkranz, J. (2010). Decision Tree. Encyclopedia of Machine Learning. C. Sammut and G. I. Webb. Boston, MA, Springer US: 263-267.
- [47] investment, c. (2018). "Adaptive Boosting Algorithm." unpublished.
- [48] Bottou, L. (2012). Stochastic Gradient Descent Tricks. Neural Networks: Tricks of the Trade: Second Edition. G. Montavon, G. B. Orr and K.-R. Müller. Berlin, Heidelberg, Springer Berlin Heidelberg: 421-436.
- [49] Banoula, M. (2023). "An Overview on Multilayer Perceptron (MLP)." unpublished.
- [50] Mucherino, A., et al. (2009). k-Nearest Neighbor Classification. Data Mining in Agriculture. A. Mucherino, P. J. Papajorgji and P. M. Pardalos. New York, NY, Springer New York: 83-106.
- [51] S. P. Dubagunta, V. Bogdan, M. Mathew, "Learning voice source related information for depression detection," ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), IEEE, 2019.
- [52] M. Vikramjit, T. Andreas, S. Elizabeth, "Noise and reverberation effects on depression detection from speech," 2016.
- [53] D. DeVault, K. SimSensei, "A virtual human interviewer for healthcare decision support," Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems, 2014.
- [54] N. Seneviratne, C. Espy-Wilson, "Speech based depression severity level classification using a multi-stage dilated cnn-lstm model," arXiv. Preprint. arXiv, 2021, pp. 2104.04195.
- [55] N. Srimadhur, S. Lalitha, "An end-to-end model for detection and assessment of depression levels using speech," Procedia. Computer. Science, vol. 171, 2020, pp. 12-21.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. Login

```
<html>
<head>
<title>PHQ-9</title>
</head>
<body>
<form name="form1" method="post" action="check_login.php">
  Login<br>
  <table border="1" style="width: 300px">
    <tbody>
      <tr>
        <td> &nbsp;&nbsp;&nbsp;Username</td>
        <td>
          <input name="txtUsername" type="text" id="txtUsername">
        </td>
      </tr>
      <tr>
        <td> &nbsp;&nbsp;&nbsp;Password</td>
        <td><input name="txtPassword" type="password" id="txtPassword">
        </td>
      </tr>
    </tbody>
  </table>
  <br>
  <input type="submit" name="Submit" value="Login">
</form>
</body>
</html>
```

2. Intro

```
<?php
ini_set('display_errors', 0);
error_reporting(~0);
session_start();
$serverName = "WIN-NTQEOAJ5N2E\SQLEXPRESS";
$username = "";
$password = '';
$dbName = "PHQSystem";
$connInfo = array("Database"=>$dbName, "UID"=>$username,
"PWD"=>$password, "MultipleActiveResultSets"=>true);
$objCon = sqlsrv_connect( $serverName, $connInfo);
if( $conn === false ) {
  die( print_r( sqlsrv_errors(), true));
}
```

```

    $strSQL = "SELECT * FROM Patient WHERE Username =
    '".$_SESSION['Username']."' ";
    $objQuery = sqlsrv_query($objCon, $strSQL);
    $objResult = sqlsrv_fetch_array($objQuery,SQLSRV_FETCH_ASSOC);
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1,
    shrink-to-fit=no" />
    <meta name="description" content="" />
    <meta name="author" content="" />
    <title>ระบบคัดกรองผู้ป่วยภาวะโรคซึมเศร้าแบบอัตโนมัติ เพื่อสร้างแบบจำลองการเรียนรู้เชิงลึกในการช่วยคัดกรองผู้ป่วย
</title>
    <!-- Favicon-->
    <link rel="icon" type="image/x-icon" href="assets/favicon.ico" />
    <!-- Bootstrap Icons-->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap-
    icons@1.5.0/font/bootstrap-icons.css" rel="stylesheet" />
    <!-- Google fonts-->
    <link
    href="https://fonts.googleapis.com/css?family=Merriweather+Sans:400,700
    " rel="stylesheet" />
    <link
    href="https://fonts.googleapis.com/css?family=Merriweather:400,300,300i
    talic,400italic,700,700italic" rel="stylesheet" type="text/css" />
    <!-- SimpleLightbox plugin CSS-->
    <link
    href="https://cdnjs.cloudflare.com/ajax/libs/SimpleLightbox/2.1.0/simpl
    eLightbox.min.css" rel="stylesheet" />
    <!-- Core theme CSS (includes Bootstrap)-->
    <link href="css/styles1.css" rel="stylesheet" />
    <link href="styles.css" rel="stylesheet" />
    <!-- Add icon library -->
    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/font-
    awesome/4.7.0/css/font-awesome.min.css">
    <style>
        a {
            text-decoration: none;
            display: inline-block;
            padding: 8px 16px;
        }

```



```

<script src="https://cdn.startbootstrap.com/sb-forms-
latest.js"></script>
</body>
</html>

<script>
function PostBlob(audioBlob, videoBlob, fileName) {
    var formData = new FormData();
    formData.append('filename', fileName);
    formData.append('audio-blob', audioBlob);
    formData.append('video-blob', videoBlob);
    xhr('RecordRTC-to-PHP/save-video-audio.php', formData,
function(ffmpeg_output){});
}

var record = document.getElementById('record');
var stop = document.getElementById('stop');
var audio = document.querySelector('audio');
var recordVideo = document.getElementById('record-video');
var preview = document.getElementById('preview');
var container = document.getElementById('container');

var recordAudio, recordVideo;
record.onclick = function() {
    record.disabled = true;
    !window.stream && navigator.getUserMedia({
        audio: true,
        video: true
    }, function(stream) {
        window.stream = stream;
        onstream();
    }, function(error) {
        alert(JSON.stringify(error, null, '\t'));
    });

    window.stream && onstream();

    function onstream() {
        setSrcObject(stream, preview);
        var binaryData = [];
        binaryData.push(stream);
    }
}

```

```

        window.URL.createObjectURL(new Blob(binaryData, {type:
"application/zip"}))
        preview.play();
        preview.muted = true;

        recordAudio = RecordRTC(stream, {
            type: 'audio',
            recorderType: StereoAudioRecorder,
            sampleRate: 44100,
            // bufferSize: 16384,
            onAudioProcessStarted: function() {
                recordVideo.startRecording();
            }
        });

        var videoOnlyStream = new MediaStream();
        videoOnlyStream.addTrack(stream.getVideoTracks()[0]);
        recordVideo = RecordRTC(videoOnlyStream, {
            type: 'video',
            // recorderType: MediaStreamRecorder || WhammyRecorder
        });
        recordAudio.startRecording();
        stop.disabled = false;
    }
};

var fileName;
stop.onclick = function() {
    record.disabled = false;
    stop.disabled = true;
    preview.src = '';
    preview.poster = 'ajax-loader.gif';
    fileName = "Intro"
    recordAudio.stopRecording(function() {
        recordVideo.stopRecording(function() {
            PostBlob(recordAudio.getBlob(), recordVideo.getBlob(),
fileName);
        });
    });
    document.getElementById("preview").style.display = "none";
    document.getElementById("record").style.display = "none";
    document.getElementById("stop").style.display = "none";
};

```

```

document.getElementById("demo").innerHTML = "วิไลโอบันทึกเสียงร้อย";
};

function xhr(url, data, callback) {
    var request = new XMLHttpRequest();
    request.onreadystatechange = function() {
        if (request.readyState == 4 && request.status == 200) {
            callback(request.responseText);
        }
    };
    request.open('POST', url);
    request.send(data);
}
</script>

```

3. PHQ-9 (คำถาม)

```

<?php
error_reporting(0);
session_start();
$_SESSION['nonce'] = substr(str_shuffle(MD5(microtime())) , 0, 10);
ini_set('display_errors', 0);
$serverName = "WIN-NTQEOAJ5N2E\SQLEXPRESS";
$username = "";
$password = '';
$dbName = "PHQSystem";

$connInfo = array("Database"=>$dbName, "UID"=>$username,
"PWD"=>$password, "MultipleActiveResultSets"=>true);

$objCon = sqlsrv_connect( $serverName, $connInfo);

if( $conn === false ) {
    die( print_r( sqlsrv_errors(), true));
}

$strSQL = "SELECT * FROM Patient WHERE Username =
'".$_SESSION['Username']."' ";
$objQuery = sqlsrv_query($objCon, $strSQL);
$objResult = sqlsrv_fetch_array($objQuery,SQLSRV_FETCH_ASSOC);
$folder = $objResult["Username"];
$audio_file_path = 'RecordRTC-to-PHP/' . 'uploads/' . $folder .
'/' . 'ชื่อ1(16000).wav';
$data = array(
    'wavfile' => new CURLFile($audio_file_path,
mime_content_type($audio_file_path), basename($audio_file_path)),

```

```

        'outputlevel' => '--uttlevel',
        'outputformat' => '--txt',
    );

    $curl = curl_init();

    curl_setopt_array($curl, array(
        CURLOPT_URL => "https://api.aiforthai.in.th/partii-webapi",
        CURLOPT_RETURNTRANSFER => true,
        CURLOPT_ENCODING => "",
        CURLOPT_MAXREDIRS => 10,
        CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
        CURLOPT_CUSTOMREQUEST => "POST",
        CURLOPT_POSTFIELDS => $data,
        CURLOPT_HTTPHEADER => array(
            "Content-Type: multipart/form-data",
            "Apikey: BvxWcV010x09SEKfP2W1U2Av5yYuAUAB"
        )
    ));

    $response = curl_exec($curl);
    $err = curl_error($curl);
    $obj = json_decode($response);

    curl_close($curl);

    if($errno = curl_errno($ch)) {
        $error_message = curl_strerror($errno);
        echo "cURL error ({$errno}):\n {$error_message}";
    }

?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no" />
    <meta name="description" content="" />
    <meta name="author" content="" />
    <title>ระบบคัดกรองผู้ป่วยภาวะโรคซึมเศร้าแบบอัตโนมัติ เพื่อสร้างแบบจำลองการเรียนรู้เชิงลึกในการช่วยคัดกรองผู้ป่วย
</title>
    <!-- Favicon-->
    <link rel="icon" type="image/x-icon" href="assets/favicon.ico" />

```

```

<!-- Bootstrap Icons-->
<link href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.5.0/font/bootstrap-icons.css" rel="stylesheet" />
<!-- Google fonts-->
<link
href="https://fonts.googleapis.com/css?family=Merriweather+Sans:400,700
" rel="stylesheet" />
<link
href="https://fonts.googleapis.com/css?family=Merriweather:400,300,300i
talic,400italic,700,700italic" rel="stylesheet" type="text/css" />
<!-- SimpleLightbox plugin CSS-->
<link
href="https://cdnjs.cloudflare.com/ajax/libs/SimpleLightbox/2.1.0/simpl
eLightbox.min.css" rel="stylesheet" />
<!-- Core theme CSS (includes Bootstrap)-->
<link href="css/styles1.css" rel="stylesheet" />
<link href="styles.css" rel="stylesheet" />
<!-- Add icon library -->
<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">
<style>
a {
text-decoration: none;
display: inline-block;
padding: 8px 16px;
}
a:hover {
background-color: #ddd;
color: black;
}
.previous {
background-color: #f1f1f1;
color: black;
}
.next {
background-color: #c34e2e;
color: white;
}
.round {

```



```

target="#navbarResponsive" aria-controls="navbarResponsive" aria-
expanded="false" aria-label="Toggle navigation"><span class="navbar-
toggler-icon"></span></button>
    <table border="0" style="width: 40px">
        <tbody>
            <tr>
                <td width="10">Welcome:&nbsp;&nbsp;&nbsp;</td>
                <td width="10"><?php echo
$objarResult["Name_Surname"];?></td>
            </tr>
        </tbody>
    </table>
</div>
</nav>
<!-- Masthead-->
<header class="masthead">
    <div class="container px-4 px-lg-5 h-100">
        <div class="row gx-4 gx-lg-5 h-100 align-items-center
justify-content-center text-center">
            <div class="col-lg-8 align-self-end">
                <h1 class="text-white font-weight-bold">1. เมื่อทำอะไร ก็
ไม่เพิกเฉย</h1>
                <hr class="divider" />
            </div>
            <div class="col-lg-8 align-self-baseline">
                <p class="text-white-75 mb-6">โปรดตอบคำถามของท่าน <br>(ด้วย
คำตอบเหล่านี้: ไม่เลย, มีบางวันหรือไม่บ่อย, มีค่อนข้างบ่อย, มีเกือบทุกวัน)</p>
                <a class="btn btn-primary btn-xl" href="#Q1"
onclick="myFunction()" id="start">ตอบคำถาม</a>
            </div>
        </div>
    </div>
</header>
<!-- Q1-->
<section class="page-section bg-primary" id="Q1">
    <div class="container px-4 px-lg-5">
        <div class="row gx-4 gx-lg-5 justify-content-center">
            <div class="col-lg-12 text-center">
                <h2 class="text-white mt-0">ตอบคำถามในรูปแบบบันทึกวีดีโอ</h2>
                <p class="text-white-75 mb-4">โปรดอ่านทวนคำถามแล้วจึงตอบคำถาม
(ตอบ ไม่เลย, มีบางวันหรือไม่บ่อย, มีค่อนข้างบ่อย, มีเกือบทุกวัน)</p>
                <div id="container">
                    <head>
                        <section class="experiment">

```



```

        <input type="text" name="txt1" id="txt1" style="display: none;"
value="<?php echo str_replace(' ', '', $obj -> message) ?>"><br/>
        <input id="name" name="name" type="text" style="display: none;"
value='<?php echo $objResult["Name_Surname"]; ?>'>
        <hr class="divider" />
        <center>
            <div class="radio-container">
                <input type="radio" name="result" value="0" id="1" />
                <label for="1">ไม่เลย</label>

                <input type="radio" name="result" value="1" id="2" />
                <label for="2">มีบางวันหรือไม่บ่อย</label>

                <input type="radio" name="result" value="2" id="3" />
                <label for="3">มีค่อนข้างบ่อย</label>

                <input type="radio" name="result" value="3" id="4" />
                <label for="4">มีเกือบทุกวัน</label>
            </div>
        </center>
        <center>
            <br>
            <input class="btn btn-primary" type="submit">
        </center>
    </div>
</form>
</section>
<audio controls autoplay style="display:none;">
    <source src="audio_auto/1.mp3" type="audio/mpeg">
</audio>

<!-- Footer-->
<footer class="bg-light py-5">
    <div class="container px-4 px-lg-5">
        <div id="container">
            <br>
            <div class="small text-center text-muted">Created by
Phanomkorn Homsiang</div>
        </div>
    </footer>
<!-- Bootstrap core JS-->

```

```

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"></script>
  <!-- SimpleLightbox plugin JS-->
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/SimpleLightbox/2.1.0/simpleLightbox.min.js"></script>
  <!-- Core theme JS-->
  <script src="js/scripts1.js"></script>
  <script src="https://cdn.startbootstrap.com/sb-forms-latest.js"></script>
  <script src="js/recorder.js"></script>
  <script src="js/app.js"></script>
</body>
</html>
<script>
</script>
<script>
function myFunction() {
  alert("ผู้วิจัยขอให้อาสาสมัครอ่านทวนคำถามแล้วจึงตอบคำถามโดยให้อ่านออกเสียงตัวอย่างเช่น คำถามข้อที่ 1 เมื่อทำอะไร ๆ ก็ไม่
เพิลิดเพลิน คำตอบคือ ไม่เลย");
}
</script>
<script>
function check() {
  if(document.form1.txt1.value.match(/เลย*/))
  {
    document.getElementById("1").checked = true;
  }
  else if(document.form1.txt1.value.match(/ไม่เลย*/)) {
    document.getElementById("1").checked = true;
  }
  else if(document.form1.txt1.value.match(/มีบางวัน*/)) {
    document.getElementById("2").checked = true;
  }
  else if(document.form1.txt1.value.match(/ไม่บ่อย*/)) {
    document.getElementById("2").checked = true;
  }
  else if(document.form1.txt1.value.match(/มีบางวันหรือไม่บ่อย*/)) {
    document.getElementById("2").checked = true;
  }
  else if(document.form1.txt1.value.match(/ค่อนข้างบ่อย*/)) {

```

```

        document.getElementById("3").checked = true;
    }
    else if(document.form1.txt1.value.match(/มีค่อนข้างบ่อย*/)) {
        document.getElementById("3").checked = true;
    }
    else if(document.form1.txt1.value.match(/ข้างบ่อย*/)) {
        document.getElementById("3").checked = true;
    }
    else if(document.form1.txt1.value.match(/ค่อนข้าง*/)) {
        document.getElementById("3").checked = true;
    }
    else if(document.form1.txt1.value.match(/เกือบทุกวัน*/)) {
        document.getElementById("4").checked = true;
    }
    else if(document.form1.txt1.value.match(/มีเกือบทุกวัน*/)) {
        document.getElementById("4").checked = true;
    }
    else if(document.form1.txt1.value.match(/มีบางวัน*/)) {
        document.getElementById("2").checked = true;
    }
    else if(document.form1.txt1.value.match(/มีบางวันหรือไม่บ่อย*/)) {
        document.getElementById("2").checked = true;
    }
    else if(document.form1.txt1.value.match(/ค่อนข้างบ่อย*/)) {
        document.getElementById("3").checked = true;
    }
    else if(document.form1.txt1.value.match(/มีค่อนข้างบ่อย*/)) {
        document.getElementById("3").checked = true;
    }
    else if(document.form1.txt1.value.match(/เกือบทุกวัน*/)) {
        document.getElementById("4").checked = true;
    }
    else if(document.form1.txt1.value.match(/มีเกือบทุกวัน*/)) {
        document.getElementById("4").checked = true;
    }
    else if(document.form1.txt1.value.match(/มีบางวัน*/)) {
        document.getElementById("2").checked = true;
    }
    else if(document.form1.txt1.value.match(/มีบางวันหรือไม่บ่อย*/)) {
        document.getElementById("2").checked = true;
    }
    else if(document.form1.txt1.value.match(/ค่อนข้างบ่อย*/)) {
        document.getElementById("3").checked = true;
    }
}

```

```

else if(document.form1.txt1.value.match(/มีค่อนข้างบ่อย*/)) {
    document.getElementById("3").checked = true;
}
else if(document.form1.txt1.value.match(/เกือบทุกวัน*/)) {
    document.getElementById("4").checked = true;
}
else if(document.form1.txt1.value.match(/มีเกือบทุกวัน*/)) {
    document.getElementById("4").checked = true;
}
}
</script>
<script>
function PostBlob(audioBlob, videoBlob, fileName) {
    var formData = new FormData();
    formData.append('filename', fileName);
    formData.append('audio-blob', audioBlob);
    formData.append('video-blob', videoBlob);
    xhr('RecordRTC-to-PHP/save-video-audio.php', formData,
function(ffmpeg_output){});
}

var record = document.getElementById('record');
var stop = document.getElementById('stop');
var audio = document.querySelector('audio');
var recordVideo = document.getElementById('record-video');
var preview = document.getElementById('preview');
var container = document.getElementById('container');

var recordAudio, recordVideo;
record.onclick = function() {
    record.disabled = true;
    !window.stream && navigator.getUserMedia({
        audio: true,
        video: true
    }, function(stream) {
        window.stream = stream;
        onstream();
    }, function(error) {
        alert(JSON.stringify(error, null, '\t'));
    });

    window.stream && onstream();

    function onstream() {

```

```

        setSrcObject(stream, preview);
        var binaryData = [];
        binaryData.push(stream);
        window.URL.createObjectURL(new Blob(binaryData, {type:
"application/zip"}))
        preview.play();
        preview.muted = true;

        recordAudio = RecordRTC(stream, {
            type: 'audio',
            recorderType: StereoAudioRecorder,
            sampleRate: 44100,
            // bufferSize: 16384,
            onAudioProcessStarted: function() {
                recordVideo.startRecording();
            }
        });

        var videoOnlyStream = new MediaStream();
        videoOnlyStream.addTrack(stream.getVideoTracks()[0]);
        recordVideo = RecordRTC(videoOnlyStream, {
            type: 'video',
            // recorderType: MediaStreamRecorder || WhammyRecorder
        });
        recordAudio.startRecording();
        stop.disabled = false;
    }
};

var fileName;
stop.onclick = function() {
    record.disabled = false;
    stop.disabled = true;
    preview.src = '';
    preview.poster = 'ajax-loader.gif';
    fileName = "ชื่อ1"

    recordAudio.stopRecording(function() {
        recordVideo.stopRecording(function() {
            PostBlob(recordAudio.getBlob(), recordVideo.getBlob(),
fileName);
        });
    });
});

```

```

document.getElementById("preview").style.display = "none";
document.getElementById("record").style.display = "none";
document.getElementById("stop").style.display = "none";
document.getElementById("demo").innerHTML = "กรุณาคลิกปุ่มเพื่อโหลดจากนั้นกดคืนยัน
คำตอบ";
};

function xhr(url, data, callback) {
    var request = new XMLHttpRequest();
    request.onreadystatechange = function() {
        if (request.readyState == 4 && request.status == 200) {
            callback(request.responseText);
        }
    };
    request.open('POST', url);
    request.send(data);
}
</script>

```

4. ผลคะแนนของ PHQ-9

```

<?php
error_reporting(0);
session_start();
$_SESSION['nonce'] = substr(str_shuffle(MD5(microtime())), 0, 10);
ini_set('display_errors', 0);
$serverName = "WIN-NTQEOAJ5N2E\SQLEXPRESS";
$username = "";
$password = '';
$dbName = "PHQSystem";

$connInfo = array("Database"=>$dbName, "UID"=>$username,
"PWD"=>$password, "MultipleActiveResultSets"=>true);

$conn = sqlsrv_connect( $serverName, $connInfo);

if( $conn === false ) {
    die( print_r( sqlsrv_errors(), true));
}

$stmt = "SELECT * FROM result WHERE name_surname =
'".$_SESSION['Username']."' ";
$query = sqlsrv_query($conn, $stmt);
$objResult = sqlsrv_fetch_array($query,SQLSRV_FETCH_ASSOC);

```

```

    $q1 = $objResult["one"];
    $q2 = $objResult["two"];
    $q3 = $objResult["three"];
    $q4 = $objResult["four"];
    $q5 = $objResult["five"];
    $q6 = $objResult["six"];
    $q7 = $objResult["seven"];
    $q8 = $objResult["eight"];
    $q9 = $objResult["nine"];
    $sum = $q1 + $q2 + $q3 + $q4 + $q5 + $q6 + $q7 + $q8 + $q9;
    $score = $sum;
    sqlsrv_close($conn);
?>
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no" />
  <meta name="description" content="" />
  <meta name="author" content="" />
  <title>ระบบคัดกรองผู้ป่วยภาวะ โรคซึมเศร้าแบบอัตโนมัติ เพื่อสร้างแบบจำลองการเรียนรู้เชิงลึก ในการช่วยคัดกรองผู้ป่วย
</title>
  <!-- Favicon-->
  <link rel="icon" type="image/x-icon" href="assets/favicon.ico" />
  <!-- Bootstrap Icons-->
  <link href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.5.0/font/bootstrap-icons.css" rel="stylesheet" />
  <!-- Google fonts-->
  <link
href="https://fonts.googleapis.com/css?family=Merriweather+Sans:400,700
" rel="stylesheet" />
  <link
href="https://fonts.googleapis.com/css?family=Merriweather:400,300,300i
talic,400italic,700,700italic" rel="stylesheet" type="text/css" />
  <!-- SimpleLightbox plugin CSS-->
  <link
href="https://cdnjs.cloudflare.com/ajax/libs/SimpleLightbox/2.1.0/simpl
eLightbox.min.css" rel="stylesheet" />
  <!-- Core theme CSS (includes Bootstrap)-->
  <link href="css/styles1.css" rel="stylesheet" />
  <link href="styles.css" rel="stylesheet" />
  <!-- Next and Previous Buttons -->

```

```

<style>
  a {
    text-decoration: none;
    display: inline-block;
    padding: 8px 16px;
  }

  a:hover {
    background-color: #ddd;
    color: black;
  }

  .previous {
    background-color: #f1f1f1;
    color: black;
  }

  .next {
    background-color: #c34e2e;
    color: white;
  }

  .round {
    border-radius: 50%;
  }
</style>
</head>

<body id="page-top">
  <!-- Navigation-->
  <nav class="navbar navbar-expand-lg navbar-light fixed-top py-3"
  id="mainNav">
    <div class="container px-4 px-lg-5">
      <a class="navbar-brand" href="#page-top">ผลการทดสอบ</a>
      <button class="navbar-toggler navbar-toggler-right"
      type="button" data-bs-toggle="collapse" data-bs-
      target="#navbarResponsive" aria-controls="navbarResponsive" aria-
      expanded="false" aria-label="Toggle navigation"><span class="navbar-
      toggler-icon"></span></button>
      <div class="collapse navbar-collapse"
      id="navbarResponsive">
        <ul class="navbar-nav ms-auto my-2 my-lg-0">

```



```

        }else if( $score >= 9 && $score <= 14 ){
            echo "ปานกลาง";
        }else if( $score >= 5 && $score <= 8 ){
            echo "น้อย";
        }else{
            echo "ไม่มีอาการ";};?<h4>
    </div>
</div>
</div>
</div>
</section>

<!-- Bootstrap core JS-->
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"></script>
<!-- Simplelightbox plugin JS-->
<script
src="https://cdnjs.cloudflare.com/ajax/libs/SimpleLightbox/2.1.0/simpleLightbox.min.js"></script>
<!-- Core theme JS-->
<script src="js-Q/Q9.js"></script>
<script src="js/scripts1.js"></script>
<script src="https://cdn.startbootstrap.com/sb-forms-latest.js"></script>
</body>
</html>

```

5. Upload File Audio and Video

```

<?php
session_start();
$serverName = "WIN-NTQE0AJ5N2E\SQLEXPRESS";
$username = "";
$password = '';
$dbName = "PHQSystem";

$connectionInfo = array("Database"=>$dbName, "UID"=>$username,
"PWD"=>$password, "MultipleActiveResultSets"=>true);

$objCon = sqlsrv_connect( $serverName, $connectionInfo);

if( $conn === false ) {
    die( print_r( sqlsrv_errors(), true));
}

```

```

}

$strSQL = "SELECT * FROM Patient WHERE Username =
'".$_SESSION['Username']."' ";
$objQuery = sqlsrv_query($objCon, $strSQL);
$objResult = sqlsrv_fetch_array($objQuery,SQLSRV_FETCH_ASSOC);
$folder = $objResult["Username"];
$filePath = 'uploads/'. $folder .'/';

$OSList = array
(
'Windows 3.11' => 'Win16',
'Windows 95' => '(Windows 95)|(Win95)|(Windows_95)',
'Windows 98' => '(Windows 98)|(Win98)',
'Windows 2000' => '(Windows NT 5.0)|(Windows 2000)',
'Windows XP' => '(Windows NT 5.1)|(Windows XP)',
'Windows Server 2003' => '(Windows NT 5.2)',
'Windows Vista' => '(Windows NT 6.0)',
'Windows 7' => '(Windows NT 7.0)',
'Windows NT 4.0' => '(Windows NT 4.0)|(WinNT4.0)|(WinNT)|(Windows
NT)',
'Windows ME' => 'Windows ME',
'Open BSD' => 'OpenBSD',
'Sun OS' => 'SunOS',
'Linux' => '(Linux)|(X11)',
'Mac OS' => '(Mac_PowerPC)|(Macintosh)',
'QNX' => 'QNX',
'BeOS' => 'BeOS',
'OS/2' => 'OS/2',
'Search
Bot'=>'(nuhk)|(Googlebot)|(Yammybot)|(Openbot)|(Slurp)|(MSNBot)|(Ask
Jeeves/Teoma)|(ia_archiver)'
);
// Loop through the array of user agents and matching operating
systems
foreach($OSList as $CurrOS=>$Match)
{
// Find a match
if (preg_match("/".$Match."/i", $_SERVER['HTTP_USER_AGENT']))
{
// We found the correct match
break;
}
}
}

```

```

// if it is audio-blob
if (isset($_FILES["audio-blob"])) {
    $uploadDirectory = $filePath . $_POST["filename"].'.wav';
    if (!move_uploaded_file($_FILES["audio-blob"]["tmp_name"],
$uploadDirectory)) {
        echo("Problem writing audio file to disk!");
    }
    else {
        // if it is video-blob
        if (isset($_FILES["video-blob"])) {
            $uploadDirectory = $filePath
.$_POST["filename"].'.webm';
            if (!move_uploaded_file($_FILES["video-
blob"]["tmp_name"], $uploadDirectory)) {
                echo("Problem writing video file to disk!");
            }
            else {
                $audioFile = $filePath . $_POST["filename"].'.wav';
                $videoFile = $filePath . $_POST["filename"].'.webm';
                $mergedFile = $filePath . $_POST["filename"].'-
merged.webm';

                // ffmpeg depends on yasm
                // libvpx depends on libvorbis
                // libvorbis depends on libogg
                // make sure that you're using newest ffmpeg
version!

                if(!strrpos($CurrOS, "Windows")) {
                    $cmd = '-i '.$audioFile.' -i '.$videoFile.' -
map 0:0 -map 1:0 '.$mergedFile;
                }
                else {
                    $cmd = ' -i '.$audioFile.' -i '.$videoFile.' -
c:v mpeg4 -c:a vorbis -b:v 64k -b:a 12k -strict experimental
'.$mergedFile;
                }

                exec('ffmpeg '.$cmd.' 2>&1', $out, $ret);
                if ($ret){
                    echo "There was a problem!\n";
                    print_r($cmd.'\n');

```

```

        print_r($out);
    } else {
        echo "Ffmpeg successfully merged audi/video
files into single WebM container!\n";

        unlink($audioFile);
        unlink($videoFile);
    }
}
}
}
}
?>

```





เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. 2 Class

1.1 1D CNN

<https://www.kaggle.com/code/gooytuatagofficial/1d-cnn-2-class-thesis-9-file>

1.2 2D CNN

<https://www.kaggle.com/code/gooytuatagofficial/2d-cnn-2-class-thesis-9-file>

1.3 LSTM

<https://www.kaggle.com/code/gooytuatagofficial/lstm-2-class-thesis-9-file>

1.4 GRU

<https://www.kaggle.com/code/gooytuatagofficial/gru-2-class-thesis-9-file>

2. 5 Class

2.1 1D CNN

<https://www.kaggle.com/code/gooytuatagofficial/1d-cnn-5-class-thesis-9-file>

2.2 2D CNN

<https://www.kaggle.com/code/gooytuatagofficial/2d-cnn-5-class-thesis-9-file>

2.3 LSTM

<https://www.kaggle.com/code/gooytuatagofficial/lstm-5-class-thesis-9-file>

2.4 GRU

<https://www.kaggle.com/code/gooytuatagofficial/gru-5-class-thesis-9-file>



ภาคผนวก ค.

Source Code ของเว็บไซต์ Machine Learning

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. 2 Class

https://drive.google.com/file/d/1mgy_x-B-MsJHakChsFBCIs9li94SSPHy/view?usp=drive_link

2. 5 Class

https://drive.google.com/file/d/1vfprGwXTtiRqtHfr4Al8SeiUFaqrSgd/view?usp=drive_link



ประวัติผู้เขียน

ชื่อ-นามสกุล	นายพนมกร หอมเสียง
วัน เดือน ปีเกิด	20 สิงหาคม 2539 ที่ชลบุรี
ที่อยู่	18/1 หมู่ 6 ต.หนองตำลิ่ง อ.พานทอง จ.ชลบุรี 20160
ประวัติการศึกษา	2558 สาขาวิชาเทคโนโลยีสารสนเทศและการสื่อสาร คณะเทคโนโลยีสารสนเทศ มหาวิทยาลัยศรีปทุม วิทยาเขตชลบุรี (ปริญญาตรี) 2563 สาขาวิศวกรรมชีวการแพทย์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง งบประมาณการทำงานและผลงานวิจัย (ปริญญาโท)
พ.ศ.2562-2563	ตำแหน่ง System Engineer ของ Hitachi Automotive System Co.,Ltd.
พ.ศ.2563-2564	ตำแหน่ง IT Engineer ของ Hitachi metals Thailand Co.,Ltd.
ปัจจุบัน	ตำแหน่ง System Engineer ของ Toyota Tsusho Forklift (Thailand) Co.,Ltd.