แบบจำลองทำนายโรคหลอดเลือดสมองด้วยเทคนิคการเรียนรู้ของเครื่อง

STROKE PREDICTION MODEL USING MACHINE LEARNING TECHNIQUES

พชร ดอกชะเอม

บัณฑิตวิทยาลัย มหาวิทยาลัยศรีนครินทรวิโรฒ

2566

แบบจำลองทำนายโรคหลอดเลือดสมองด้วยเทคนิคการเรียนรู้ของเครื่อง

พชร ดอกชะเอม

สารนิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
วิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการข้อมูล
คณะวิทยาศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ
ปีการศึกษา 2566

STROKE PREDICTION MODEL USING MACHINE LEARNING TECHNIQUES

POTCHARA DOCKCHAAM

A Master's Project Submitted in Partial Fulfillment of the Requirements

for the Degree of MASTER OF SCIENCE

(Data Science)

Faculty of Science, Srinakharinwirot University

2023

สารนิพนธ์

เรื่อง

แบบจำลองทำนายโรคหลอดเลือดสมองด้วยเทคนิคการเรียนรู้ของเครื่อง

ของ

พชร ดอกชะเอม

ได้รับอนุมัติจากบัณฑิตวิทยาลัยให้นับเป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

ปริญญาวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการข้อมูล

ของมหาวิทยาลัยศรีนครินทรวิโรฒ

(รองศาสตราจารย์ นายแพทย์ฉัตรชัย เอกปัญญาสกุล)

คณบดีบัณฑิตวิทยาลัย

คณะกรรมการสอบปากเปล่าสารนิพนธ์

..... ที่ปรึกษาหลัก

(อาจารย์ ดร.เรืองศักดิ์ ตระกูลพุทธิรักษ์)

..... ประธาน

(อาจารย์ ดร.สุทธิพงศ์ ธัชยพงษ์)

..... กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.ศิริสรรพ เหล่าหะเกียรติ)

ชื่อเรื่อง	แบบจำลองทำนายโรคหลอดเลือดสมองด้วยเทคนิคการเรียนรู้ของเครื่อง
ผู้วิจัย	พชร ดอกชะเอม
ปริญญา	วิทยาศาสตรมหาบัณฑิต
ปีการศึกษา	2566
อาจารย์ที่ปรึกษา	อาจารย์ ดร. เรืองศักดิ์ ตระกูลพุทธิรักษ์

โรคหลอดเลือดสมอง (Stroke) เป็นหนึ่งในสาเหตุการเสียชีวิตและทุพพลภาพที่สำคัญของประชากรทั่วโลก การวินิจฉัยโรคหลอดเลือดสมองในระยะเริ่มแรกมีความสำคัญอย่างมากในการลดอัตราการเสียชีวิตและความพิการที่ตามมา อย่างไรก็ตามการวินิจฉัยโรคหลอดเลือดสมองต้องอาศัยความเชี่ยวชาญของแพทย์ ซึ่งมีอยู่อย่างจำกัด ผู้วิจัยจึงเล็งเห็นการนำเทคนิคการเรียนรู้ของเครื่อง (Machine Learning) ในการสร้างแบบจำลองเพื่อช่วยในการจำแนกผู้ป่วยโรคหลอดเลือดสมอง โดยอาศัยข้อมูลคุณลักษณะของผู้ป่วยในการสร้างแบบจำลองเพื่อลดภาระของแพทย์และทำให้สามารถช่วยลดระยะเวลาคัดกรองผู้ป่วยได้ งานวิจัยนี้เป็นการศึกษาการสร้างแบบจำลองด้วยเทคนิคการเรียนรู้ของเครื่อง โดยชุดข้อมูลที่นำมาใช้ในการสร้างแบบจำลองมาจากเว็บไซต์ Kaggle ซึ่งเป็นข้อมูลทางคลินิกของผู้ป่วยมี 2 ประเภทผู้ คือ ผู้ป่วยปกติและผู้ป่วยโรคหลอดเลือดสมอง จำนวนทั้งหมด 5,110 คน ในการศึกษาข้อมูลชุดนี้มีลักษณะชุดข้อมูลไม่สมดุล (Imbalanced Data) ซึ่งอาจส่งผลต่อประสิทธิภาพของแบบจำลอง ทำให้ต้องนำเทคนิคการจัดการข้อมูลไม่สมดุลของข้อมูลด้วยวิธีต่างๆมาช่วยในการจัดการข้อมูลร่วมด้วย ในการหาแบบจำลองที่มีประสิทธิภาพดีที่สุดจะมาจากทำการเปรียบเทียบการสร้างแบบจำลองด้วยอัลกอริทึมที่หลากหลายได้แก่ Logistic Regression, Decision Tree, Random Forest, XGBoost, LightGBM, AdaBoost และCatBoost การเปรียบเทียบจะใช้ตัววัดประสิทธิภาพที่มาจากผลลัพธ์การทำนายของแบบจำลองด้วย Confusion Matrix ประกอบด้วย ความแม่นยำ (Accuracy), ความอ่อนไหว, F1-score, Specificity, ROC Curve และความแม่นยำสมดุล (Balanced Accuracy) แต่ในงานวิจัยนี้จะให้ความสำคัญกับความแม่นยำสมดุลเป็นตัววัดประสิทธิภาพหลัก เป็นเพราะชุดข้อมูลไม่สมดุลที่มีความต่างของจำนวนประเภทข้อมูลทั้งสอง ทำให้ต้องเลือกใช้ตัววัดประสิทธิภาพที่ให้ความสำคัญกับน้ำหนักของประเภทจำนวนข้อมูล จากผลการสร้างแบบจำลองพบว่าแบบจำลองที่สร้างด้วยอัลกอริทึม AdaBoost ให้ประสิทธิภาพสูงที่สุดด้วยค่าความแม่นยำสมดุลที่ 0.72 และหากผู้ศึกษาต้องการเพิ่มประสิทธิภาพของแบบจำลองสามารถทำได้โดยการเพิ่มตัวอย่างข้อมูล และการปรับจูนพารามิเตอร์ (Parameter-Tuning) ด้วยอัลกอริทึม GridSearchCV

คำสำคัญ : การเรียนรู้ของเครื่อง, ชุดข้อมูลไม่สมดุล, ความแม่นยำสมดุล

Title	STROKE PREDICTION MODEL USING MACHINE LEARNING TECHNIQUES
Author	POTCHARA DOCKCHAAM
Degree	MASTER OF SCIENCE
Academic Year	2023
Thesis Advisor	Lecturer Ruangsak Trakunphutthirak , Ph.D.

Stroke is one of the leading causes of death and disability worldwide; therefore, the early diagnosis of stroke is crucial in reducing mortality rates and subsequent disabilities. However, diagnosing a stroke required the limited expertise of medical professionals. The researchers realized the potential of using Machine Learning techniques to create models that can help classify stroke patients based on patient characteristic data, thereby reducing the burden on doctors and enabling faster patient screening. This research involved the study of model creation using Machine Learning techniques, with the dataset used for model creation coming from the Kaggle website. This dataset includes clinical data of 5110 samples, comprised of both normal individuals and stroke patients, and features imbalanced data, which can affect the performance of the model. Various techniques were employed to manage the imbalanced data. The study compared different models created using various algorithms including Logistic Regression, Decision Tree, Random Forest, XGBoost, LightGBM, AdaBoost, and CatBoost. The comparison used performance metrics derived from the Confusion Matrix, including Accuracy, Sensitivity, F1-score, Specificity, ROC Curve, and Balanced Accuracy. However, this research prioritized Balanced Accuracy as the main performance metric due to the imbalanced data set, which required a performance metric that considered the weight of the data categories. The results showed that the model created with the AdaBoost algorithm had the highest performance with a Balanced Accuracy score of 0.72. If researchers want to improve performance, they can do so by increasing the sample size and performing parameter tuning using the GridSearchCV algorithm.

Keyword : Machine learning, Imbalanced data, Balanced accuracy

กิตติกรรมประกาศ

การจัดทำวิจัยฉบับนี้สำเร็จลุล่วงไปได้ด้วยดีจากการสนับสนุน ความรู้ ความช่วยเหลือ คำแนะนำ ตลอดจนแนวทางในการทำวิจัยและจัดทำสารนิพนธ์ของ อ.ดร.เรืองศักดิ์ ตระกูลพุทธิรักษ์ อาจารย์ที่ปรึกษา และคณาจารย์ทุกท่านในหลักสูตรวิทยาการข้อมูล ภาควิชาวิทยาการคอมพิวเตอร์ คณะ วิทยาศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรต การสนับสนุนจากบัณฑิตวิทยาลัย มหาวิทยาลัยศรีนคริ นทรวิโรฒ ในการนำเสนอผลงานวิจัย ผู้วิจัยจึงขอขอบคุณมา ณ ที่นี้

พชร ดอกชะเอม

สารบัญ

หน้า

สารบัญตาราง

หน้า

สารบัญรูปภาพ

หน้า

บทที่ 1
บทนำ

1.1 ความเป็นมาของงานวิจัย

โรคหลอดเลือดสมอง (Stroke) เป็นหนึ่งในโรคที่เกิดขึ้นเมื่อการไหลเวียนของเลือดไปยังสมองถูกขัดขวางทำให้เซลล์สมองถูกทำลายและเกิดความเสียหายได้ ซึ่งความเสียหายนี้ส่งผลให้เกิดผลกระทบต่อร่างกายอย่างมาก เช่น ปากเบี้ยว ไม่มีแรง หรือถึงขั้นเสียชีวิต และยังมีโอกาสทำให้เกิดโรคแทรกซ้อนต่าง ๆระหว่างการรักษา ไม่ว่าจะเป็น อัมพฤกษ์ อัมพาต แต่สามารถลดความเสี่ยงและความเสียหายได้หากผู้ป่วยรักษาได้รับการรักษาได้อย่างถูกต้อง และได้รับคำแนะนำก่อนที่จะเป็นโรคหลอดเลือดสมอง

ประเทศจีนมีอัตราส่วนของผู้ป่วยโรคหลอดเลือดสมองอยู่ที่ 253-620 คน ต่อแสนประชากร ซึ่ง 50-70% ของผู้ป่วยจะกลายเป็นผู้ทุพพลภาพ โดยเฉพาะในปี 2019 ประเทศจีนเผชิญกับวิกฤตผู้ป่วยโรคหลอดเลือดสมองครั้งใหญ่ มีผู้ป่วยรวมทั้งหมด 28.76 ล้านคน ซึ่งมีผู้ป่วยรายใหม่เพิ่มขึ้น 3.94 ล้านคน และเสียชีวิต 2.19 ล้านคน (Shan et al., 2023)

ในประเทศไทยมีผู้ป่วยที่เป็นโรคหลอดเลือดสมองเพิ่มขึ้นมาโดยตลอดตั้งแต่ปี 2560-2565 ผู้ป่วยโรคหลอดเลือดสมองมีอัตราการเสียชีวิตที่อยู่ในเกณฑ์ที่สูง คือจากการเก็บข้อมูลตัวอย่างผู้ที่มีอายุ 15 ปีขึ้นไป มีผู้ป่วยเป็นโรคหลอดเลือดสมองอยู่ที่ 279-331 คน ต่อแสนประชากร ซึ่งผู้ป่วยโรคหลอดเลือดสมองมีอัตราการเสียชีวิตที่อยู่ในเกณฑ์ที่สูง คือประมาณร้อยละ 10-11 มาโดยตลอด และยังไม่มีแนวโน้มจะลดลง ในตลอดช่วง 6 ปีที่ผ่านมา โดยเฉพาะในปี 2565 มีอัตราการเสียชีวิตที่สูงถึงร้อยละ 16 และถึงแม้ว่าผู้ป่วยโรคหลอดเลือดสมองจะได้รับการรักษาและรอดชีวิตได้ ร่างกายผู้ป่วยจะมีความพิการหลงเหลืออยู่คิดเป็นร้อยละ 90 และร้อยละ 50 มีความพิการรุนแรงถึงขั้นไม่สามารถช่วยเหลือตัวเองได้ (เทียมเก่า, 2023)

ตาราง 1 จำนวนผู้ป่วยโรคหลอดเลือดสมองต่อแสนประชากรไทยอายุ 15 ปีขึ้น

	2560	2561	2562	2563	2564	2565
เขต 1 เชียงใหม่	253.16	280.02	297.59	299.36	313.45	315.01
เขต 2 พิษณุโลก	284.84	319.76	329.61	326.76	331.61	342.26
เขต 3 นครสวรรค์	347.34	369.01	387.2	392.67	392.27	398.44
เขต 4 สระบุรี	333.5	344.04	356.32	348.74	359.75	357.84
เขต 5 ราชบุรี	300.04	329.86	346.01	349.45	342.32	356.35
เขต 6 ระยอง	313.11	331.41	342.26	336.39	332.95	340.05
เขต 7 ขอนแก่น	247.69	272.67	284.97	292.42	299.49	318.20
เขต 8 อุดรธานี	243.34	264.57	275.69	289.79	293.71	299.64
เขต 9 นครราชสีมา	304.81	340.77	358.53	363.41	372.34	379.03
เขต 10 อุบลราชธานี	249.37	269.83	279.95	303.45	298.76	303.82
เขต 11 สุราษฎร์ธานี	271.35	290.76	318.66	325.01	315.5	323.07
เขต 12 สงขลา	255.96	295.54	318.67	307	304.58	298.54
เขต 13 กรุงเทพมหานคร	250.1	264.32	280.22	346.4	219.86	285.44
ประเทศไทย	278.49	303.2	318.89	328.01	330.22	330.72

ที่มา: (เทียมเก่า, 2023)

จากตาราง 1 พบว่าจำนวนผู้ป่วยโรคหลอดเลือดสมองมีจำนวนที่เพิ่มขึ้นทุกปี แต่ในปี 2564-2565 บางจังหวัดมีผู้ป่วยลดลง เนื่องจากสถานการณ์ของโรคโควิดที่ทำให้มีมาตรการ lock down ทำให้เข้าถึงการรักษาในระบบที่ยากลำบากส่งผลให้การเก็บข้อมูลลดลงด้วย (เทียมเก่า, 2023)

ปัจจัยที่ทำให้เกิดโรคหลอดเลือดสมองนั้น สามารถแบ่งได้เป็น 2 ปัจจัยหลัก ดังนี้

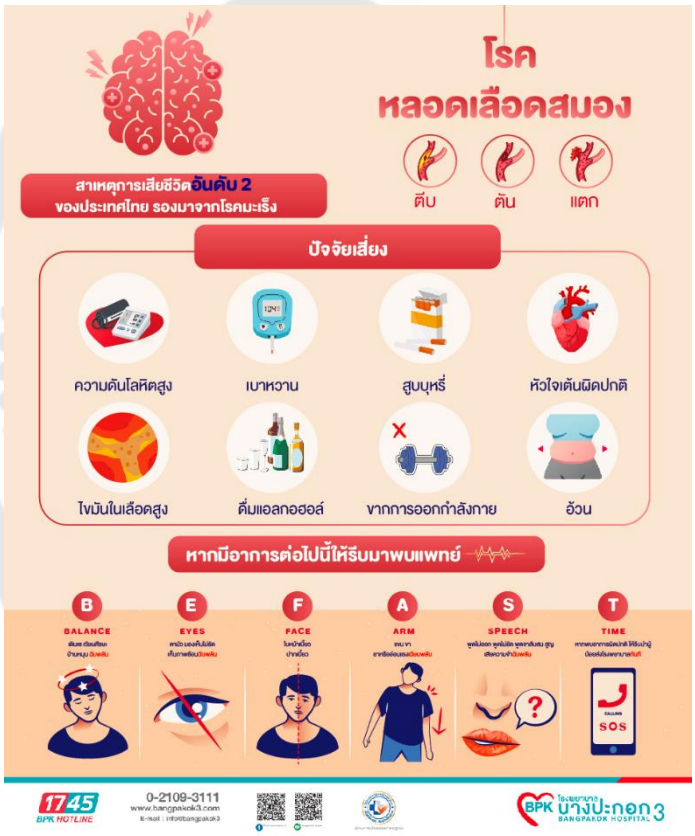
1. ปัจจัยที่ปรับเปลี่ยนไม่ได้ เช่น อายุ เพศ พันธุกรรม
2. ปัจจัยที่สามารถปรับเปลี่ยนได้ เช่น พฤติกรรมการบริโภคอาหาร การออกกำลังกาย, การจัดการความเครียด, การสูบบุหรี่ และการดื่มเครื่องดื่มแอลกอฮอล์

จากการศึกษาพบว่ากลุ่มเสี่ยงโรคหลอดเลือดสมอง มีพฤติกรรมการดูแลสุขภาพไม่เหมาะสมในด้านการรับประทานอาหาร การออกกำลังการ การใช้ยา และมีโรคประจำตัว รวมถึงมีความรู้ความเข้าใจเกี่ยวกับโรคหลอดเลือดสมองไม่เพียงพอ ขาดทักษะและความตระหนักในการปรับเปลี่ยนพฤติกรรมเพื่อป้องกันโรคหลอดเลือดสมอง (Shan et al., 2023)

การที่บุคคลจะเกิดความตระหนักและปฏิบัติพฤติกรรมสุขภาพที่เหมาะสม จะต้องมีความรู้ความเข้าใจเรื่องโรคและการดูแลตนเอง โดย ความรู้ทั่วไปที่ประชาชนควรรู้ควรครอบคลุม สาเหตุ ปัจจัย เสี่ยง อาการของโรค อันตราย อาการเตือน การควบคุม ป้องกัน การรักษา การดูแลตนเองเมื่อมีอาการเตือนโรคหลอดเลือดสมองและการช่วยเหลือผู้ที่มีอาการโรคหลอดเลือดสมอง

และการปรับเปลี่ยนพฤติกรรมเป็นวิธีการหนึ่งที่มีประสิทธิผลในการป้องกันการเกิดโรค (จิตตนูนท์, 2021)

ทางผู้วิจัยเห็นว่าการทำนายผู้ป่วยที่มีโอกาสจะเป็นโรคหลอดเลือดสมอง มีความสำคัญในการตัดสินใจที่จะช่วยวางแผนในการรักษาแก่ผู้ป่วยได้อย่างทันท่วงที จึงมีความคิดที่จะนำเอาเทคโนโลยีการเรียนรู้ของเครื่อง (Machine Learning) มาใช้สร้างแบบจำลองจำแนกเพื่อทำนายผู้ป่วยโรคหลอดเลือดสมองและผู้ป่วยปกติ โดยใช้ข้อมูลทางการแพทย์และคลินิกเพื่อนำมาใช้คำนวณสร้างแบบจำลอง



ภาพประกอบ 1 ปัจจัยเสี่ยงและอาการที่มีโอกาสเป็นโรคหลอดเลือดสมอง

ที่มา: https://www.gardenia.net/plant/pistacia-vera

จากภาพประกอบ 1 ปัจจัยเสี่ยงที่ทำให้เกิดโรคหลอดเลือดสมองจะเกิดจากได้ทั้งปัจจัยภายในร่างกาย เช่น ความดันโลหิตสูง, เบาหวาน หัวใจเต้นผิดปกติ ฯลฯ ซึ่งเป็นผลมาจากการที่มีพฤติกรรมที่ไม่ดีต่าง ๆ เช่น การสูบบุหรี่, การดื่มแอลกอฮอล์ รวมถึงการขาดการออกกำลังกาย

อาการของเบื้องต้นของผู้ป่วยโรคหลอดเลือดสมองประกอบด้วย วิงเวียนศีรษะ, สายตาพล่ามัว, ร่างกายสูญเสียสมดุล (ใบหน้าและริมฝีปากเบี้ยว และทรงตัวไม่อยู่) หากมีการเหล่านี้ควรพบแพทย์ผู้เชี่ยวชาญ เพื่อรับการวินิจฉัยในการหาแนวทางในการรักษาให้ทันท่วงที

1.2 ความมุ่งหมายของงานวิจัย

ในการวิจัยนี้ผู้วิจัยได้ตั้งความมุ่งหมายไว้ดังนี้

1. เพื่อศึกษาและสร้างแบบจำลองจำแนกเพื่อทำนายผู้ป่วยโรคหลอดเลือดสมอง โดยใช้เทคนิคการเรียนรู้ของเครื่อง
2. เพื่อทราบถึงลำดับขั้นตอนในการสร้างแบบจำลองที่ถูกต้อง
3. เพื่อศึกษาและเปรียบเทียบประสิทธิภาพของแบบจำลองของอัลกอริทึม
4. เพื่อหาวิธีที่เหมาะสมในการแก้ปัญหาชุดข้อมูลที่ไม่สมดุล (Imbalanced Data)
5. เพื่อทราบถึงการวัดประสิทธิภาพของแบบจำลองทำนายที่เหมาะสม

1.3 ขอบเขตงานวิจัย

1. ข้อมูลที่ใช้ในงานวิจัยนี้จากเว็บไซต์ Kaggle ที่เป็นข้อมูลสาธารณะชื่อว่า Stroke Prediction เป็นข้อมูลจำนวนผู้ป่วยมี 2 คำตอบคือ 0: ผู้ป่วยปกติ และ 1: ผู้ป่วยโรคหลอดเลือดสมอง มีจำนวนข้อมูลทั้งหมด 5,110 รายการ
2. ใช้ภาษา Python ในการนำอัลกอริทึมไปใช้สร้างแบบจำลอง ดังนี้
3. แบบจำลองพื้นฐาน เช่น Logistic Regression และDecision Tree
4. แบบจำลองซับซ้อน เช่น Random forest, XGBoost, AdaBoost, LightGBM และ CatBoost
5. ใช้เทคนิคการจัดการประเภทของข้อมูลให้อยู่ในรูปแบบที่เหมาะสม
6. LabelEncoder ในการจัดการข้อมูล ข้อมูลจัดเป็นกลุ่ม (Categorical Data) ให้อยู่ในรูปของ ข้อมูลเชิงตัวเลข (Numerical Data)
7. StandardScaler เพื่อให้ข้อมูลอยู่ในมาตรฐานเดียวกันและกระจายตัวของข้อมูลให้น้อยที่สุด
8. ใช้เทคนิค RandomGridsearch เพื่อหาค่า Hyperparameter เพื่อสร้างแบบจำลองที่มีประสิทธิภาพสูงที่สุดของแต่ละอัลกอริทึม
9. การเลือกตัวประเมินประสิทธิภาพที่เหมาะสมกับการนำไปใช้ในการวิเคราะห์กับชุดข้อมูลและแบบจำลอง

10. การปรับค่า Threshold เพื่อให้แบบจำลองให้น้ำหนักข้อมูลที่นำมาใช้ทำนายได้อย่างเหมาะสม

11. ใช้เทคนิค การลดจำนวนข้อมูลแบบสุ่ม (Random Undersampling), การเพิ่มข้อมูลแบบสุ่ม (Random Oversampling) และ SMOTE (Synthetic Minority Oversampling Technique) ในการแก้ปัญหาชุดข้อมูลที่มีลักษณะชุดข้อมูลไม่สมดุล

1.4 ขั้นตอนการดำเนินงานวิจัย

1. ทำความเข้าใจคุณลักษณะที่ทำให้เกิดโรคหลอดเลือดสมอง
2. ศึกษางานวิจัยที่เกี่ยวกับการจัดการชุดข้อมูลไม่สมดุล
3. ศึกษางานวิจัยที่เกี่ยวข้องกับการสร้างแบบจำลองจำแนกแบบไบนารี (Binary Classification Model) เพื่อใช้ในการทำนายผู้ป่วยโรคหลอดเลือดสมอง
4. ศึกษาโครงสร้างอัลกอริทึมของการเรียนรู้ของเครื่อง
5. นำเข้าชุดข้อมูลที่เตรียมไว้ โดยมี 2 คำตอบที่มีจำนวนข้อมูลไม่เท่ากัน
6. จัดการข้อมูลให้อยู่ในรูปแบบที่เหมาะสม เพื่อนำไปใช้ในการวิเคราะห์และคำนวณ รวมถึงการจัดการชุดข้อมูลไม่สมดุล ให้เหมาะสมกับการนำไปใช้ในการสร้างแบบจำลอง
7. เปรียบเทียบประสิทธิภาพของแบบจำลองของแต่ละอัลกอริทึม จากตัวประเมินประสิทธิภาพของแบบจำลอง โดยดูจากผลลัพธ์และวิเคราะห์ข้อดีข้อเสียที่ได้จากการจำแนกของแบบจำลอง
8. ปรับค่า Hyperparameter ของแบบจำลอง เพื่อให้แบบจำลองมีประสิทธิภาพสูงที่สุด
9. สามารถวิเคราะห์ข้อดีข้อเสียของแบบจำลองที่สร้างได้ รวมถึงข้อจำกัดที่จะนำไปใช้งานในแต่ละสถานการณ์

1.5 สมมติฐานในงานวิจัย

1. ตัวประเมินประสิทธิภาพของแบบจำลอง มีปัญหาความไม่เหมาะสมในการคำนวณกับชุดข้อมูลไม่สมดุล สามารถแก้ปัญหาด้วยหน่วยวัดประสิทธิภาพ ความแม่นยำสมดุล (Balanced Accuracy)
2. แบบจำลองที่ซับซ้อนมีโครงสร้างการคำนวณที่ซับซ้อนกว่าแบบจำลองพื้นฐาน ทำให้การจำแนกมีประสิทธิภาพที่สูงกว่า

3. การปรับ Threshold สามารถช่วยแก้ปัญหาของแบบจำลองในการตรวจจับการจำแนกกลุ่มข้อมูลจำนวนมาก (Majority Class) และกลุ่มข้อมูลจำนวนน้อย (Minority Class) อย่างไร

1.6 **สรุปบทนำ**

จากการศึกษาบทความที่เกี่ยวข้องกับโรคหลอดเลือดสมองพบว่าโรคหลอดเลือดสมองมีอัตราผู้ป่วยใหม่เพิ่มขึ้นทุกปีทั้งในประเทศไทยและนอกประเทศ ทำให้ผู้วิจัยเห็นว่าการหากมีการนำเทคโนโลยีการเรียนรู้ของเครื่องมาใช้ในการสร้างแบบจำลองเพื่อจำแนกผู้ป่วยโรคหลอดเลือดสมอง จะสามารถช่วยลดระยะเวลาและการสูญเสียให้ลดน้อยลงได้ โดยการสร้างแบบจำลองจะจะต้องอยู่ในขอบเขตของงานวิจัย และสามารถตอบโจทย์สมมติฐานได้งานวิจัยได้

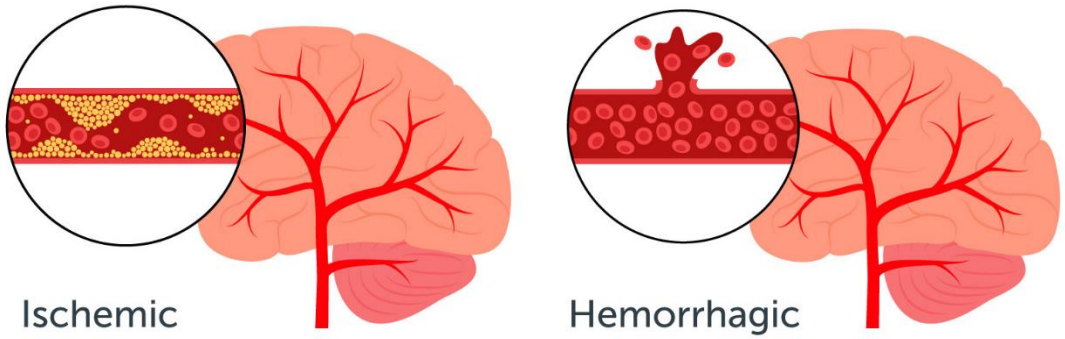
บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในงานวิจัยครั้งนี้ผู้วิจัยได้ศึกษาเอกสารและงานวิจัยที่เกี่ยวข้อง และได้จำแนกออกตามหัวข้อดังต่อไปนี้

1. ความรู้เกี่ยวกับโรคหลอดเลือดสมอง
2. ทฤษฎีที่เกี่ยวกับ Machine Learning
3. ทฤษฎีที่เกี่ยวกับอัลกอริทึม
4. ทฤษฎีที่เกี่ยวกับการเตรียมข้อมูล
5. ทฤษฎีการประเมินประสิทธิภาพแบบจำลอง
6. งานวิจัยที่เกี่ยวข้อง
7. สรุปทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวกับโรคหลอดเลือดสมอง

เป็นโรคที่เกิดจากปริมาณเลือดที่ไปเลี้ยงสมองลดลง ทำให้เกิดอาการและอาการแสดงทางระบบประสาทอย่างทันทีทันใด โดยมี 2 สาเหตุ คือ มีการอุดตันภายในหลอดเลือดสมองจนทำให้เกิดภาวะสมองขาดเลือด (Ischemic Stroke) หรือมีการแตกของหลอดเลือดสมอง (Hemorrhagic Stroke)



ภาพประกอบ 2 ตัวอย่างสมองที่มีภาวะขาดเลือด (Ischemic Stroke) และสมองที่มีการแตกของหลอดเลือด (Hemorrhagic Stroke)

ที่มา: https://www.memorialcare.org/services/stroke-care/stroke

ภาพประกอบ 2 อาการของโรคหลอดเลือดสมองจะมี 2 ลักษณะ คือ 1. โรคหลอดเลือดสมองที่มีภาวะขาดเลือด (โรคหลอดเลือดสมองตีบ) เกิดจากการที่มีลิ่มเลือดหรือหลอดเลือดสมองมีการอุดตัน ทำให้เลือดไม่สามารถไหลเวียนไปเลี้ยงที่สมองได้ 2. โรคหลอดเลือดสมองที่มีภาวะแตก (โรคหลอดเลือดสมองแตก) เกิดจากการที่หลอดเลือดมีการแตกทำให้ขาดเลือดไปเลี้ยงสมอง ซึ่งทั้ง 2 ลักษณะ หากไม่ได้รับการรักษาในทันทีภายในระยะเวลาไม่นานเนื้อสมองจะจะตายลง

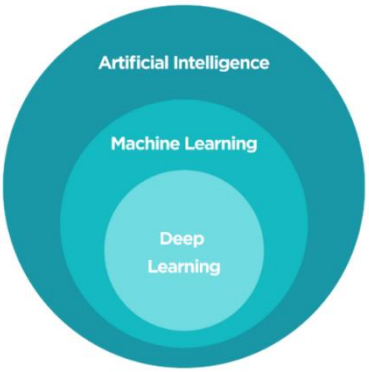
โรคหลอดเลือดสมองเป็นเหตุฉุกเฉินทางการแพทย์ ซึ่งต้องได้รับการรักษาอย่างเร่งด่วน หากได้รับการรักษาอย่างทันท่วงทีจะช่วยลดผลกระทบที่เกิดกับร่างกายและลดอัตราการเสียชีวิตได้ อาการที่เกิดจากการเป็นโรคหลอดเลือดสมองขึ้นอยู่กับตำแหน่งที่สมองขาดเลือดไปเลี้ยง อาการที่สามารถพบได้บ่อย เช่น การอ่อนแรงที่ใบหน้า แขนขา หรืออาจมีอาการอื่นร่วมด้วย

ปัจจัยเสี่ยงที่ทำให้เกิดโรคหลอดเลือดสมองแบ่งเป็น 2 ปัจจัยหลัก คือ ปัจจัยที่ไม่สามารถปรับเปลี่ยนได้ เช่น อายุที่เพิ่มขึ้น เพศ พันธุกรรม และปัจจัยที่สามารถปรับเปลี่ยนได้ ได้แก่ การบริโภคอาหาร การออกกำลังกาย การจัดการความเครียด การสูบบุหรี่ การป้องกันโรคจึงต้องมีการควบคุมและลดปัจจัยเสี่ยง โดยการปรับเปลี่ยนพฤติกรรมให้เหมาะสม (จิตตนูนท์, 2021)

2.2 ทฤษฎีที่เกี่ยวกับการเรียนรู้ของเครื่อง

การเรียนรู้ของเครื่อง คือ กลไกการเขียนอัลกอริทึม โดยนำอัลกอริทึม ไปเรียนรู้ชุดข้อมูลต้นแบบ (Training Set) ให้ได้ออกมาเป็นสมการ จากนั้นนำไปใช้งานกับชุดข้อมูลทดสอบ (Testing Set) ได้ออกมาเป็นแบบจำลองในการนำไปใช้งานตามวัตถุประสงค์

การเรียนรู้ของเครื่องเป็นหน่วยย่อยของหัวข้อ ปัญญาประดิษฐ์ (Artificial Intelligence) ซึ่งมีการทำงานโดยมีเป้าหมายเพื่อให้คอมพิวเตอร์เรียนรู้ได้ด้วยตนเองโดยการใช้ “ข้อมูล” และการเรียนรู้ของเครื่องมีหน่วยย่อยอีกเรียกว่า การเรียนรู้เชิงลึก (Deep Learning) ซึ่งมีลักษณะการทำงานเสมือนการเรียนรู้ของเครื่อง แต่มีการคิดคำนวณที่ซับซ้อนกว่า ทำให้ประสิทธิภาพเหมาะกับงานที่มีการใช้การคำนวณที่ยากกว่า



ภาพประกอบ 3 ความสัมพันธ์ของปัญญาประดิษฐ์ การเรียนรู้ของเครื่อง และการเรียนรู้เชิงลึก

ที่มา: https://www.byteant.com/blog/computer-vision-vs-machine-learning-vs-deep-learning-guide-to-ai-applications/

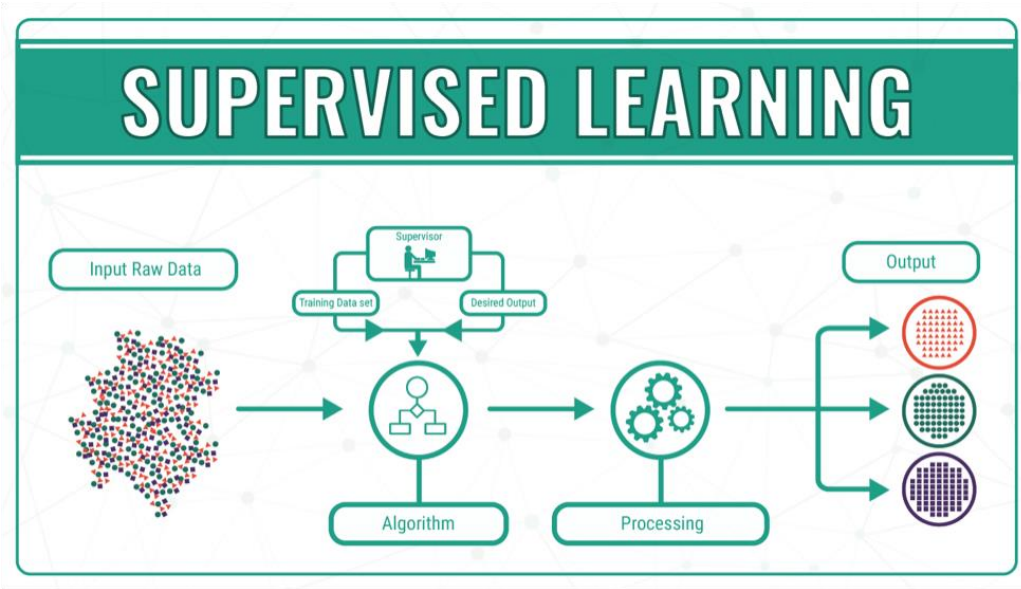
ภาพประกอบ 3 การเรียนรู้ของเครื่องเป็นหน่วยย่อยของหัวข้อ ปัญญาประดิษฐ์ (Artificial Intelligence) ซึ่งมีการทำงานโดยมีเป้าหมายเพื่อให้คอมพิวเตอร์เรียนรู้ได้ด้วยตนเองโดยการใช้ “ข้อมูล” และการเรียนรู้ของเครื่องมีหน่วยย่อยอีกเรียกว่า การเรียนรู้เชิงลึก (Deep Learning) ซึ่งมีลักษณะการทำงานเสมือนการเรียนรู้ของเครื่อง แต่มีการคิดคำนวณที่ซับซ้อนกว่า ทำให้ประสิทธิภาพเหมาะกับงานที่มีการใช้การคำนวณที่ยากกว่า

ประเภทของการเรียนรู้ของเครื่อง การสร้างขึ้นอยู่กับเป้าหมายในสร้างแบบจำลอง สามารถแบ่งออกได้เป็น 3 ประเภท ดังนี้

2.2.1 Supervised Learning

ประเภทของอัลกอริทึมที่สามารถแก้ปัญหา โดยอาศัยชุดข้อมูลที่มีผลลัพธ์อยู่แล้วในการเรียนรู้ หลังจากเรียนรู้ระยะหนึ่งอัลกอริทึมจะมีความสามารถเพียงพอที่จะสามารถแก้ปัญหาเองได้

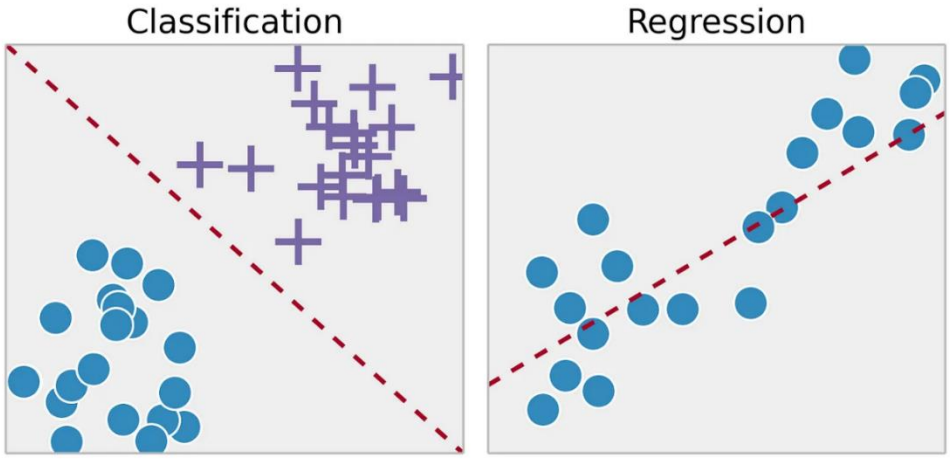
ตัวอย่างการแก้ปัญหาในรูปแบบ Supervised Learning ตัวอย่างเช่น การจำแนกภาพหมาและแมว ซึ่งการสร้างอัลกอริทึมนี้ต้องมีการรวบรวมข้อมูล แล้วให้ผู้สร้างแบบจำลองใส่คำตอบให้กับข้อมูลภาพเหล่านั้นแล้วจึงนำไปสร้างแบบจำลอง เพื่อใช้ในการจำแนกหมาและแมว



ภาพประกอบ 4 ขั้นตอนการทำงานของอัลกอริทึม Supervised Learning

ที่มา: https://www.wmaterials.net/data-science.html

ภาพประกอบ 4 ชุดข้อมูลที่นำมาใช้กับอัลกอริทึม Supervised Learning ต้องเป็นชุดข้อมูลที่มีผลลัพธ์อยู่ก่อนแล้ว งานที่เหมาะกับการสร้างด้วยอัลกอริทึม Supervised Learning มี 2 ประเภท คือ การจำแนก (Classification) และการพยากรณ์ (Regression)



ภาพประกอบ 5 ประเภทของอัลกอริทึม Supervised Learning

ที่มา: https://towardsdatascience.com/supervised-vs-unsupervised-learning-14f68e32ea8d

จากภาพประกอบ 5 แบบจำลองที่สร้างด้วยอัลกอริทึม Supervised Learning มี 2 ประเภท คือ การจำแนก ที่ใช้ในการหาคำตอบของกลุ่มข้อมูลที่ได้มีการกำหนดเป็นตัวเลือกไว้ และการพยากรณ์ ที่ใช้ในการทำนายอนาคต และหาค่าความสัมพันธ์ระหว่างตัวแปร

2.2.1.1 **แบบจำลองการถดถอย** (Regression Model)

การนำไปใช้กับงานหาคำตอบเป็นเชิงตัวเลข เหมาะกับงานนำไปใช้ทำนาย หรือพยากรณ์ เช่น การทำนายยอดขาย การทำนายระดับความดันเลือดฯลฯ

2.2.1.2 **แบบจำลองการจำแนก** (Classification Model)

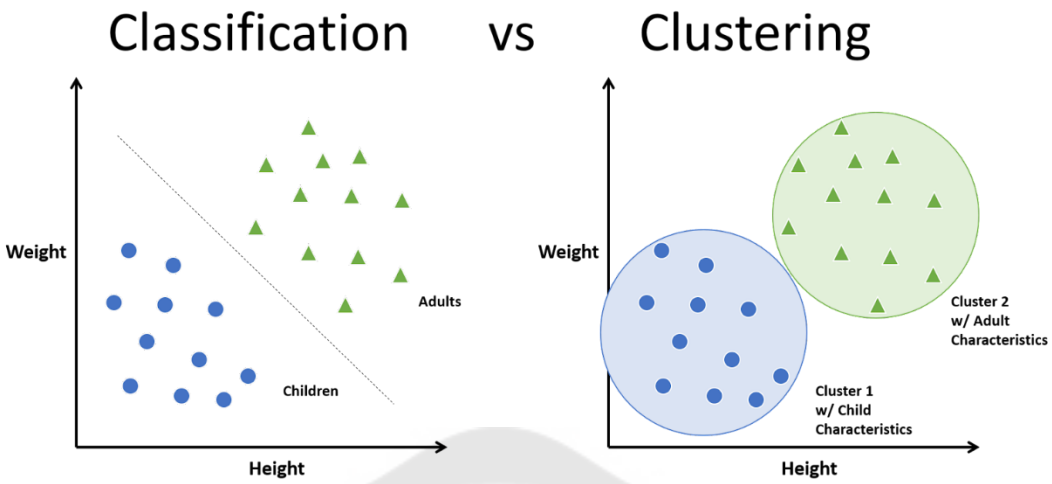
ใช้ในงานหาคำตอบเชิงจำแนก ซึ่งประเภทของงานจำแนกมีหลายประเภท เช่น การจำแนกเพศหญิงหรือชาย (คำตอบที่ได้มีเพียงชาย/หญิง การจำแนกประเภทนี้เรียกว่า การจำแนกแบบสองกลุ่มข้อมูล หรือการจำแนกกลุ่มข้อมูลหลายประเภทของสัตว์สี่ขา (คำตอบที่ได้มีมากกว่า 2 กลุ่ม เช่น หมู, สุนัข, ลา หรือ แพะ เป็นต้น การจำแนกประเภทนี้เรียกว่า การจำแนกหลายกลุ่มข้อมูล (Multiclass Classification))

2.2.2 Unsupervised Learning

อัลกอริทึมที่เรียนรู้โดยอาศัยชุดข้อมูลที่ไม่มีผลลัพธ์ของข้อมูล แต่ผู้สร้างแบบจำลองจะต้องกำหนดสิ่งที่ต้องการจากข้อมูลเหล่านั้น อัลกอริทึมจะทำการวิเคราะห์และจำแนกจากข้อมูลที่ได้รับมา ประเภทอัลกอริทึม Unsupervised Learning สามารถแบ่งได้ออกเป็น 3 ประเภทหลัก

2.2.2.1 **การจับกลุ่ม** (Clustering)

กระบวนการจัดกลุ่ม เป็นการนำข้อมูลมาจัดกลุ่ม โดยข้อมูลที่อยู่ในกลุ่มเดียวกันจะมีความสัมพันธ์หรือมีคุณลักษณะบางอย่างที่คล้ายกัน งานที่เหมาะกับการใช้อัลกอริทึมนี้ ตัวอย่างเช่น การจัดกลุ่มลูกค้า จากพฤติกรรมการซื้อเครื่องสำอาง ทำให้สามารถแบ่งกลุ่มลูกค้าออกมาได้ทั้งหมด 5 กลุ่ม ได้แก่ กลุ่ม Skincare (ผม ผิวหน้า ผิวกาย), กลุ่มตา (มาสคาร่า ที่เขียนขอบตา อายแชโดว์), กลุ่มเครื่องสำอางบริเวณหน้า (แป้ง รองพื้น คอนซิลเลอร์), กลุ่ม ลิปสติก, และน้ำหอม เมื่อได้กลุ่มของลูกค้ามาแล้ว จึงนำไปทำการตลาดด้วยวิธีการต่าง ๆ เพื่อให้ลูกค้าสามารถเข้าถึงสินค้าได้มากขึ้น

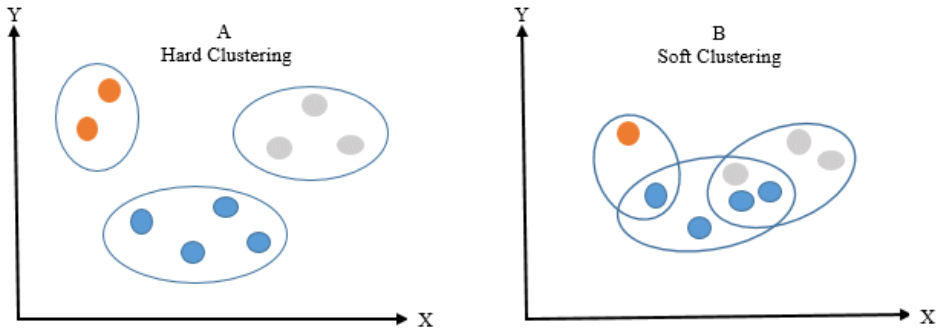


ภาพประกอบ 6 เปรียบเทียบการทำแบบจำลอง Classification กับ Clustering

ที่มา: https://www.analyticsvidhya.com/blog/2021/05/what-why-and-how-of-spectral-clustering/

ภาพประกอบ 6 อัลกอริทึมการสร้างแบบจำลองการจับกลุ่ม มีหลายรูปแบบ ยกตัวอย่าง เช่น Centroid-based Clustering, Density-based Clustering, Distribution-based Clustering, Hierarchical Clustering เป็นต้น การเลือกใช้เทคนิคขึ้นอยู่กับผลลัพธ์ของแบบจำลองที่ต้องการ

แต่ประเภทของผลลัพธ์ที่ได้จากการใช้มี 2 ประเภท คือ Hard Clustering คือกำหนดให้ผลลัพธ์ของข้อมูลที่จัดกลุ่มแยกออกจากกันโดยสิ้นเชิง (มีเพียงผลลัพธ์เดียว) และ Soft Clustering คือการที่ข้อมูลมีโอกาสที่จะอยู่ในหลายกลุ่มได้ (ผลลัพธ์ที่ได้เป็นความน่าจะเป็น) (Srimarong, 2020) ดังภาพประกอบ 7

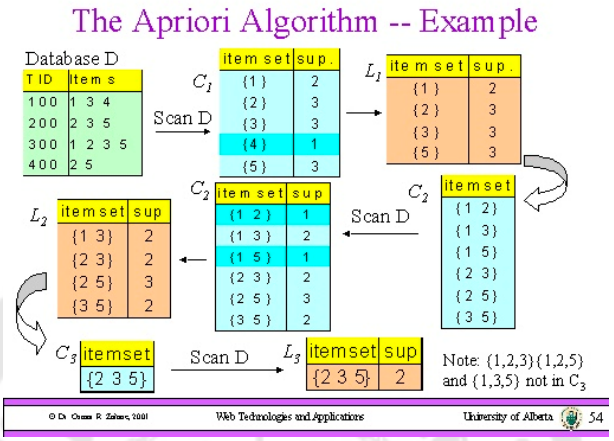


ภาพประกอบ 7 ผลลัพธ์จากการแบ่งกลุ่มของ Hard clustering และ Soft clustering

ที่มา: (Obiedat et al., 2020)

2.2.2.2 Association Rules

วิธีการแบบกฎที่ใช้ในการค้นหาความสัมพันธ์ระหว่างตัวแปรในชุดข้อมูล วิธีการเหล่านี้มักใช้ในการวิเคราะห์ตะกร้าตลาด (Market Basket Analysis) ทำให้ผู้ใช้งานเข้าใจความสัมพันธ์ของผลิตภัณฑ์ และพฤติกรรมการบริโภคของลูกค้าได้ดีขึ้น (Team, 2019)



ภาพประกอบ 8 ตัวอย่างขั้นตอนการทำงานของ Apriori Algorithm

ที่มา: https://sivaanalytics.wordpress.com/tag/market-basket-analysis/

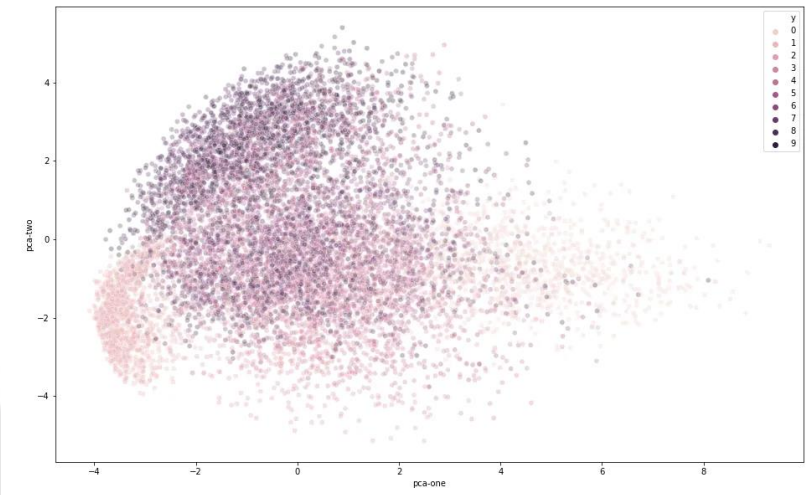
ภาพประกอบ 8 Association Rules มีโครงสร้างอัลกอริทึมที่หลากหลายเช่น Apriori, Eclat และ FP-Growth แต่อัลกอริทึมนี่นิยมแพร่หลายมากที่สุดคือ Apriori ซึ่งเป็นวิธีที่ใช้สร้างกฎความสัมพันธ์ของจำนวนสินค้า โดยดูจากความถี่ของจำนวนรายการที่มีการซื้อร่วมกัน งานที่เหมาะกับการใช้อัลกอริทึมนี้จะเป็นงานประเภท Recommendation เช่น การแนะนำเพลงแนวเดียวกัน โดยดูจากเพลงที่เคยฟังในอดีต หรือการแนะนำสินค้าที่มักมีรายการซื้อร่วมกัน เป็นต้น

2.2.2.3 Dimensionality reduction

การลดขนาดหรือมิติของข้อมูลที่มีจำนวนมากและไม่มีความสำคัญต่อการสร้างแบบจำลอง เนื่องจากข้อมูลที่มีขนาดใหญ่อาจส่งผลต่อระยะเวลาและความแม่นยำในการในการเรียนรู้ของอัลกอริทึมและเครื่องมือที่ใช้ในการวิเคราะห์ข้อมูล ในบางครั้งอาจทำให้แบบจำลองที่สร้างมาเกิดการ Overfitting กับชุดข้อมูลที่นำมาใช้ในการทำนาย

หลักการของการลดขนาดข้อมูลคือการเลือกคุณลักษณะ ที่มีความน่าจะเป็นให้ผลลัพธ์ได้ดีที่สุด ตัวอย่างเทคนิคการลดขนาดข้อมูลที่มีการใช้กันอย่างแพร่หลาย ดังนี้

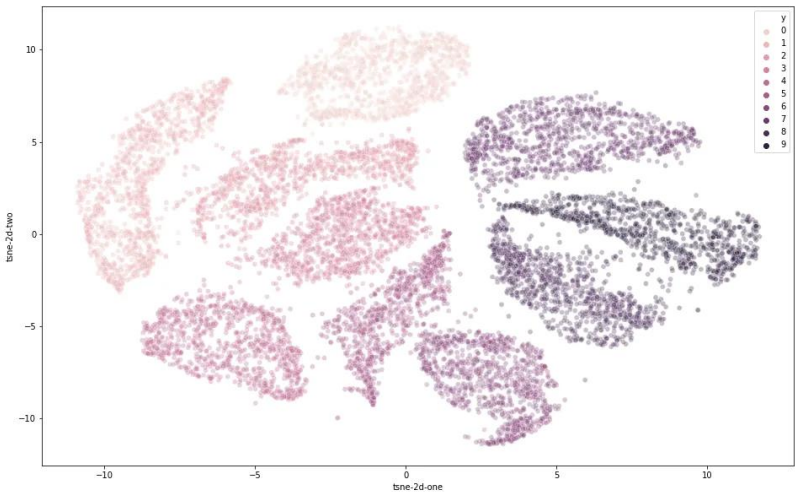
1. PCA (Principal Component Analysis) ใช้ในกระบวนการจัดการข้อมูลเพื่อลดความซ้ำซ้อน ตัดตัวแปรที่มีความสำคัญน้อยออก และให้คุณลักษณะของข้อมูลมีมิติที่เล็กลง โดยไม่ทำให้ข้อมูลมีประสิทธิภาพลดลง ดังภาพประกอบ 9



ภาพประกอบ 9 การแสดงตำแหน่งของข้อมูลหลังการทำ PCA

ที่มา: (Kanraweekultana, 2019)

2. t-SNE (t-Distributed Stochastic Neighbor Embedding) ใช้ในการลดขนาดของข้อมูล โดยอาศัยความคล้ายคลึงหรือใกล้เคียงกันของมิติข้อมูล เทคนิคนี้เหมาะกับการนำคุณลักษณะของข้อมูลมาจับคู่ แล้วแบ่งเป็นกลุ่มที่ชัดเจน และทำให้อยู่ในรูปของภาพสองมิติ ทำให้ผู้ใช้งานสามารถใช้งานได้ง่ายขึ้น ดังภาพประกอบ 10



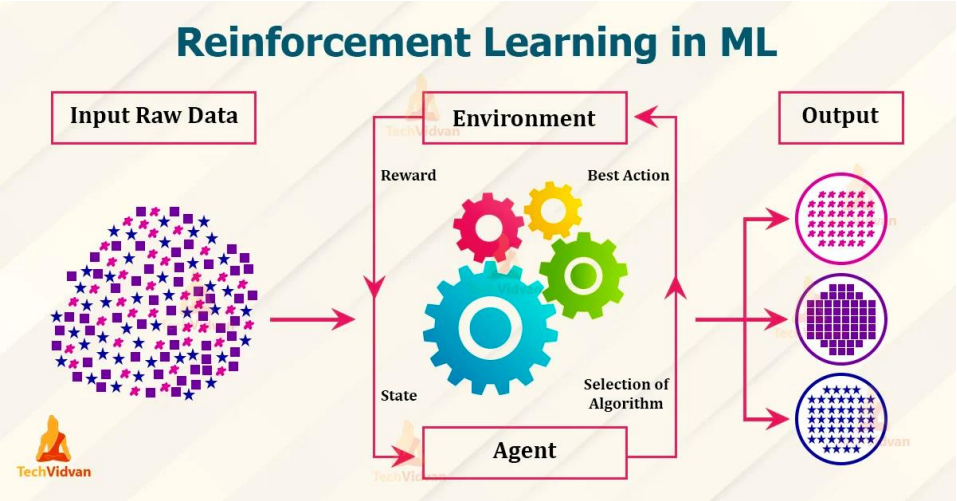
ภาพประกอบ 10 การแสดงจุดของข้อมูลหลังการทำ t-SNE

ที่มา: (Kanraweekultana, 2019)

เทคนิค Dimensional Reduction ยังมีอีกหลายรูปแบบ เช่น Singular Value Decomposition, Autoencoders และ Univariate Analysis เป็นต้น ขึ้นอยู่กับผลลัพธ์ของแบบจำลองที่ต้องการ โดยทั้งหมดจะใช้หลักการเดียวกันคือ การลดขนาดมิติของข้อมูลให้ลดลง เพื่อลดระยะเวลาในการเรียนรู้ของอัลกอริทึม และลดการเกิด Overfitting ของแบบจำลอง โดยไม่ทำให้ประสิทธิภาพของแบบจำลองลดลง (Kanraweekultana, 2019)

2.2.3 Reinforcement Learning

การเรียนรู้โดยให้อัลกอริทึมเรียนรู้ด้วยตัวเอง โดยกำหนดให้มีลักษณะการทำงาน การตัดสินใจเหมือนกับการเรียนรู้ของมนุษย์ ในการเรียนรู้อัลกอริทึมจะเรียนรู้จากการลองผิดลองถูกจากสถานการณ์ในอดีต และพยายามพัฒนาความสามารถให้ดีขึ้น ตัวอย่างงานที่ใช้อัลกอริทึมนี้ การเล่นเกมโกะให้ชนะมนุษย์ หรือการพิจารณาเลือกซื้อสินทรัพย์ และการลงทุนรูปแบบต่าง ๆ (Kanraweekultana, 2019)



ภาพประกอบ 11 ระบบขั้นตอนการทำงานของอัลกอริทึม Reinforcement Learning

ที่มา: (Abdullahi, 2023)

ภาพประกอบที่ 11 เป็นกระบวนการที่ให้แบบจำลองได้เรียนรู้เสมือนกับมนุษย์ โดยการให้ข้อมูลจำนวนหนึ่ง และให้แบบจำลองจำแนกข้อมูลดังกล่าวจาก Environment ที่กำหนดเพื่อให้ได้ผลลัพธ์ที่ดีที่สุดจากการจำแนกครั้งที่ผ่านมา

2.3 ทฤษฎีเกี่ยวกับอัลกอริทึม

จากการทบทวนวรรณกรรมพบว่ามีอัลกอริทึมของการเรียนรู้ที่ใช้ในการสร้างแบบจำลองที่หลากหลาย แต่ผู้วิจัยพบว่ามีอัลกอริทึมเพียงไม่กี่ประเภทที่ให้ประสิทธิภาพที่ดีในการทำนายผู้ป่วยโรคหลอดเลือดสมอง ดังนั้นจากการทบทวนวรรณกรรม ผู้วิจัยตัดสินใจเลือกใช้ 7 อัลกอริทึม ประกอบด้วย Logistic Regression, Decision Tree, Random Forest, XGBoost, Adaboost, LightGBM และCatboost ในการสร้างแบบจำลองจำแนกเพื่อทำนายผู้ป่วยโรคหลอดเลือดสมอง และเปรียบเทียบผลลัพธ์ที่ดีที่สุดของอัลกอริทึมในการทำนายผู้ป่วยโรคหลอดเลือดสมอง

2.3.1 Logistic Regression

$$y = \frac{e^{(b_0+b_1X)}}{1+e^{(b_0+b_1X)}} \quad (1)$$

y = คำตอบของผลลัพธ์จากการทำนายของแบบจำลอง ถ้ามีผลลัพธ์เท่ากับ 0 คือผู้ป่วยปกติ และเท่ากับ 1 ผู้ป่วยโรคหลอดเลือดสมอง

โดยที่

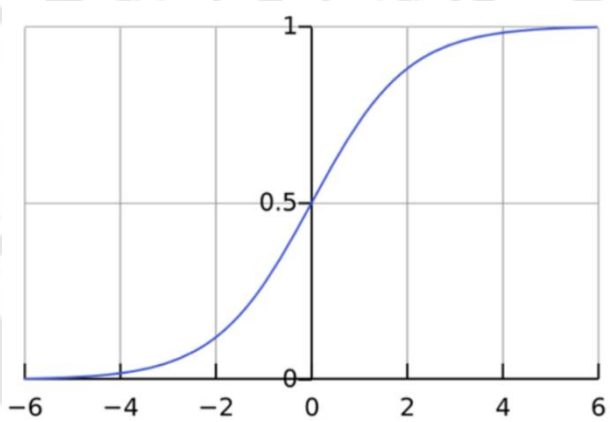
X = ข้อมูล

b_0= ค่าความอคติ (Bias)

b_1= ค่าความสัมพันธ์ระหว่างข้อมูล (Coefficient)

Logistic Regression เป็นการวิเคราะห์ที่มีเป้าหมายเพื่อทำนายโอกาสที่จะเกิดเหตุการณ์ที่สนใจ และสามารถแบ่งการทำนายออกได้เป็น 2 ประเภท คือ Binary Logistic Regression ใช้กับข้อมูลที่มีตัวแปรเกณฑ์ที่แบ่งออกเป็น 2 กลุ่มย่อย และ Multinomial Logistic Regression ใช้กับตัวแปรเกณฑ์ที่มีหลายกลุ่มย่อยหรือมีมากกว่า 2 กลุ่มย่อย (ไกยวรรณ์, 2012)

สำหรับงานวิจัยนี้ใช้ Binary Logistic Regression เนื่องจากผลลัพธ์ของคำตอบมีเพียง 2 คำตอบ คือ 0 : ผู้ป่วยปกติ และ 1 : ผู้ป่วยโรคหลอดเลือดสมอง



ภาพประกอบ 12 ผลลัพธ์ที่ได้จากแบบจำลองที่ใช้อัลกอริทึม Logistic Regression

ที่มา: https://en.wikipedia.org/wiki/Logistic_function

จากภาพประกอบที่ 12 อัลกอริทึม Binary Logistic Regression ให้ผลลัพธ์เป็นแบบ Sigmoid หากผลลัพธ์ที่คำนวณได้น้อยกว่าหรือเท่ากับ 0 จะให้ผลลัพธ์เป็นค่า 0 แต่หากค่าที่ได้มากกว่า 0 จะให้ผลลัพธ์เป็นค่า 1 เท่านั้น

2.3.2 Decision Tree

Decision Tree เป็นอัลกอริทึมที่ใช้การโยงเหตุการณ์ต่าง ๆที่อาจเกิดขึ้นในลักษณะของกิ่งไม้หรือต้นไม้ เพื่อหาทางเลือกที่ดีที่สุด ซึ่งสามารถนำไปใช้กับการเรียนรู้ของเครื่อง ได้ทั้งประเภท Supervised learning และ Unsupervised learning ได้ และ Decision Tree สามารถแบ่งได้เป็น 2 ประเภท ดังนี้

2.3.2.1 Regression Tree

ใช้สำหรับการสร้างแบบจำลองการถดถอย โดยมีค่า RSS (Residual sum of squares) เป็นเป้าหมายในการหาจุดที่ดีที่สุดในการแบ่งข้อมูล โดยการให้ค่า RSS มีค่าน้อยที่สุด

e_i (Residual) คือ ค่าความคลาดเคลื่อน ระหว่างตัวแปร y_i ในทุก ๆ จุดในข้อมูล กับ $\hat{y}$ ที่ได้มาจากการประมาณค่าขึ้นมาการคำนวณ Residual ของข้อมูลตัวที่ i

$$e_i = y_i - \hat{y}_i \tag{2}$$

จากสมการ 2 การคำนวณด้วย Residual จะมีข้อบกพร่องในกรณีที่ค่าของ $\hat{y}$ มีค่ามากและ y มีผลลัพธ์ที่มีค่าบวกและลบสลับกันในการคำนวณของแต่ละข้อมูล จะทำให้ e_i มีค่าน้อยกว่าความเป็นจริง ยกตัวอย่างเช่น

ข้อมูลที่ 1 y_1 : 5 และ $\hat{y}$: 7 ค่า e_1 เท่ากับ -2

ข้อมูลที่ 2 y_2 : 7 และ $\hat{y}$: 5 ค่า e_2 เท่ากับ 2

เมื่อนำ e_i ของข้อมูลทั้ง 2 มารวมกับจะทำให้มีค่าของแบบจำลองมีค่า e_i เท่ากับ 0 ซึ่งค่า e เท่ากับ -2 และ 2 ถือเป็นค่าความคลาดเคลื่อนจากการทำนายของแบบจำลองทั้งสองค่า ดังนั้นจึงต้องมีการปรับสมการให้มีความเหมาะสมมากขึ้น โดยใช้ค่า RSS ในการคำนวณค่าความคลาดเคลื่อน ดังสมการที่ 3

$$RSS = {e_1}^2 + {e_2}^2 + {e_3}^2 + \ldots + {e_n}^2 \tag{3}$$

RSS คือ การวัดค่าความคลาดเคลื่อนของทุก ๆข้อมูลในชุดข้อมูล แล้วนำมายกกำลังสองเพื่อให้ค่า e_i มีค่าเป็นบวก และเป็นการทำ Normalize ด้วย เพื่อแก้ปัญหาความคลาดเคลื่อนน้อยกว่าความเป็นจริง

2.3.2.2 Classification Tree

ใช้สำหรับการสร้างแบบจำลองจำแนก โดยมีค่า Gini Impurity หรือ Entropy เป็นเป้าหมายในการหาจุดที่ดีที่สุดในการแบ่งข้อมูล โดยการแบ่งข้อมูลจุดที่มีค่า Gini Impurity ต่ำที่สุด

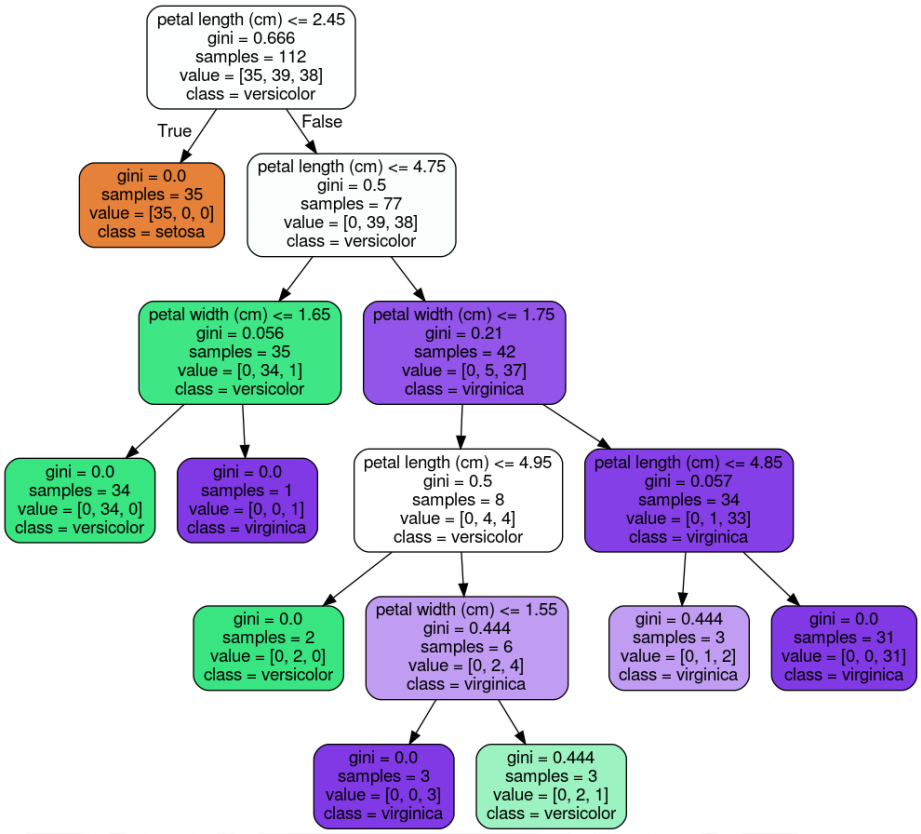
Gini Impurity คือ การวัดค่าความไม่บริสุทธิ์ (Impurity) ในการอธิบายกลุ่มของข้อมูลที่ถูกแบ่งออกมาจากคุณลักษณะ นั่นหมายความว่าถ้าค่าความไม่บริสุทธิ์ยิ่งต่ำก็ยิ่งแบ่งข้อมูลออกมาได้ดีนั่นเอง โดยค่าของ Gini Impurity จะมีค่าระหว่าง 0 ถึง 0.05

$$G = \sum_{K=1}^{K} \hat{p}\, mk(1 - \hat{p}mk) \quad (4)$$

สมการที่ 4 แสดงการคำนวณ Gini Impurity เป็นการหาค่าความน่าจะเป็นของเหตุการณ์ที่สนใจ แล้วนำมารวมกันใช้ในการหาค่า Weighted Gini Impurity สำหรับการแยกข้อมูล โดยการเลือกเหตุการณ์ที่คุณลักษณะของข้อมูลมีค่า Weighted Gini Impurity มีค่าต่ำสุด

$$Gini_{split} = \sum_{i=1}^{K} \frac{n_i}{n} GINI(i) \quad (5)$$

สมการที่ 5 แสดง Weighted Gini Impurity คือ การนำผลรวม Gini Impurity ของชุดข้อมูล คูณด้วยจำนวนข้อมูลที่สนใจ แล้วหารกับจำนวนข้อมูลในชุดข้อมูลทั้งหมด ยิ่งมีค่าน้อยความน่าจะเป็นจะถูกนำมาใช้ในการแบ่งชุดข้อมูลก่อน



ภาพประกอบ 13 ขั้นตอนกระบวนการแบ่งข้อมูลของ Classification Tree

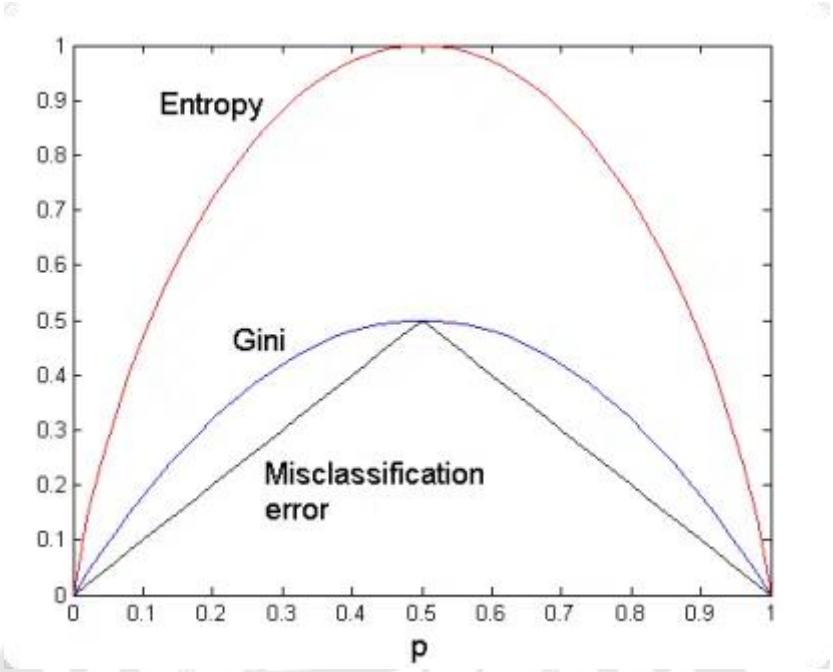
ที่มา: https://guopai.github.io/ml-blog09.html

จากภาพประกอบ 13 Classification Tree จะมี Hyper parameter ที่สำคัญในการแบ่งข้อมูลที่สามารถปรับค่าได้ โดยมีค่าที่สำคัญบางส่วนในการป้องกันการเกิดปัญหา overfitting เช่น max_depth คือ จำนวนความลึกของต้นไม้ และ min_samples_leaf จำนวนขั้นต่ำในการแบ่งข้อมูล และ criterion เป็น Hyper parameter ที่ใช้ในการเลือกสมการในการหาเป้าหมายการแบ่งข้อมูลที่ดีที่สุด โดยจะมีค่ามาตรฐานเป็น Gini index และสามารถเลือกเป็นสมการ Entropy ได้ ทั้ง 2 สมการจะให้ผลลัพธ์ที่แตกต่างกัน

Entropy คือ การวัด Randomness (ความไม่แน่นอน) เหมาะกับการนำไปใช้ในการอธิบายการแบ่งคุณลักษณะ ยิ่งมีค่าที่ต่ำยิ่งบอกถึงการแบ่งข้อมูลที่ดี

$$D = -\sum_{K=1}^{K} \hat{p}\, mk\ log\ \hat{p}mk \qquad (6)$$

มีการคำนวณคล้าย Gini Impurity แต่ใช้ค่า Log ของเหตุการณ์ที่สนใจ และค่า Entropy มีค่าอยู่ระหว่าง 0 ถึง 1 (Jamjumrat, 2022) ดังภาพประกอบที่ 14



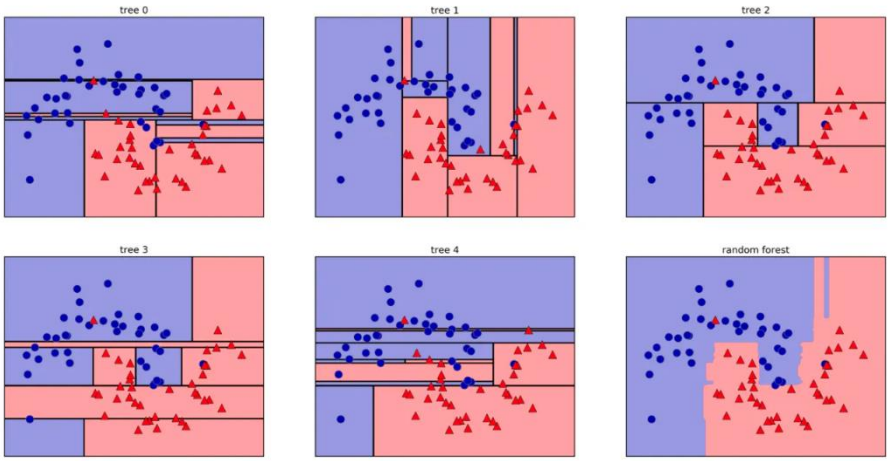
ภาพประกอบ 14 ความสัมพันธ์ค่าคำนวณในการอธิบายการแบ่งข้อมูลโดยใช้ Entropy

ที่มา: https://www.researchgate.net/figure/Relation-among-Entropy-Gini-Index-and-Misclassification-error_fig1_339471092

2.3.3 Random Forest

Random Forest เป็นอัลกอริทึมที่ใช้เทคนิค Bagging method ในการสร้างแบบจำลอง สามารถใช้สร้างแบบจำลองการถดถอย และแบบจำลองจำแนก โดย Random Forest เป็นขั้นตอนที่พัฒนาต่อยอดมาจาก Decision Tree ต่างกันที่ Random Forest จะนำ Decision Tree หลาย ๆแบบจำลองมาทำงานร่วมกัน ทำให้มีประสิทธิภาพการทำงานและพยากรณ์สูงขึ้น อีกทั้งยังช่วยลดโอกาสการเกิด Overfitting ของแบบจำลอง

Random Forest มีหลักการทำงาน คือ จะแบ่งข้อมูลออกเป็น Decision Tree หลาย ๆ ต้น โดยแต่ละต้นจะได้รับคุณลักษณะและข้อมูลที่ไม่เหมือนกันทั้งหมด เพื่อทำให้ได้ต้นไม้ที่มีความหลากหลายและมีความอิสระต่อกันมากขึ้น



ภาพประกอบ 15 แสดงการทำงานของอัลกอริทึม Random Forest

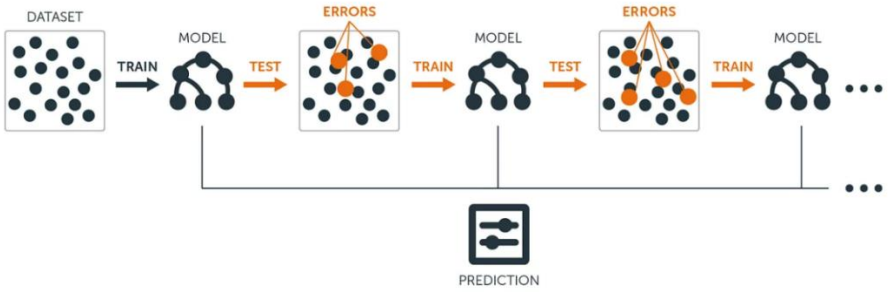
ที่มา: (PradyaSin, 2019)

ภาพประกอบที่ 15 มีการแบ่งข้อมูลให้ Decision Tree ทั้งหมด 5 ต้น (tree0-tree4) โดยที่ต้นไม้แต่ละต้นจะมีคุณลักษณะและข้อมูลที่ไม่เหมือนกัน แล้วนำผลลัพธ์ของต้นไม้ทั้งหมดมารวมกัน ในกรณีของแบบจำลองจำแนกจะใช้การโหวตของผลการจำแนก (Vote) เป็นตัวตัดสิน หากกลุ่มข้อมูลไหนมีจำนวนโหวตมากที่สุด (Majority Vote) คำตอบของแบบจำลอง Random Forest จะเป็นกลุ่มข้อมูลนั้น

การแบ่งจำนวนต้นไม้จะใช้ค่า Random Forest Error Rate หาค่าความสัมพันธ์ระหว่าง Decision Tree แต่ละแบบจำลอง ถ้าหากมีค่าสูง หมายความว่าต้นไม้แต่ละต้นมี Feature และข้อมูลที่ใกล้เคียงกัน ดังนั้นการสร้างแบบจำลองควรมีค่าความสัมพันธ์ที่ต่ำ ทำให้ต้นไม้ที่แบ่งออกมามี Feature ที่แตกต่างกัน และจะทำให้แบบจำลองที่สร้างมีประสิทธิภาพ

2.3.4 Extreme Gradient Boosting (XGBoost)

XGBoost เป็นอัลกอริทึมที่ใช้เทคนิค Boosting method ในการสร้างแบบจำลอง พัฒนามาจาก Gradient Boosting มีหลักการทำงานสร้าง Decision Tree หลาย ๆแบบจำลอง แล้วนำข้อผิดพลาดจากการทำนาย มาปรับปรุงแบบจำลอง Decision Tree ถัดไป เพื่อลดค่าความผิดพลาด ทำให้ประสิทธิภาพของแบบจำลองที่สร้างด้วยอัลกอริทึมดีขึ้น แต่มีข้อเสียคือเกิด Overfitting ของแบบจำลองได้ง่าย



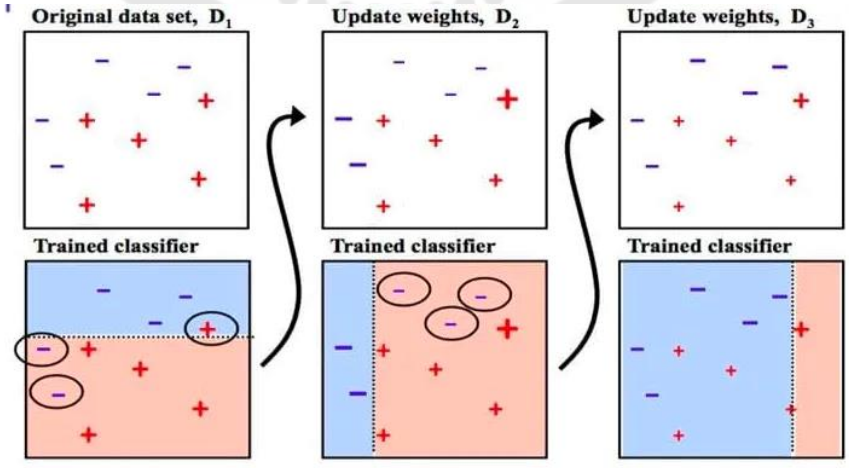
ภาพประกอบ 16 แสดงขั้นตอนการสร้างแบบจำลองด้วย XGBoost

ที่มา: https://blog.bigml.com/2017/03/14/introduction-to-boosted-trees/

ภาพประกอบ 16 การสร้างแบบจำลองด้วย XGBoost เริ่มจากการนำข้อมูลมาให้แบบจำลองทำการเรียนรู้ ผลลัพธ์จากการทำนายที่ผิดพลาดจะถูกนำมาเรียนรู้ให้กับแบบจำลองไปจนกว่าจะได้แบบจำลองที่ได้ประสิทธิภาพที่ดีที่สุด

2.3.5 Adaptive Boosting Algorithm (AdaBoost)

AdaBoost เป็นอัลกอริทึมที่ใช้เทคนิค Boosting method หลักการ คือการใช้ Decision Tree ชั้นเดียว เรียกว่า Weak model มาทำการเรียนรู้ต่อกันเป็นลูกโซ่ ในแต่ละครั้งที่เรียนรู้จะมีการเก็บค่าความผิดพลาดที่เกิดขึ้นจากการเรียนรู้ แล้วทำการปรับน้ำหนักในกลุ่มข้อมูล แล้วทำการเรียนรู้ใหม่จนกว่าจะได้ Strong model มีค่าประสิทธิภาพที่สูงมาใช้งาน (โค้ววิลัยแสง, 2021)



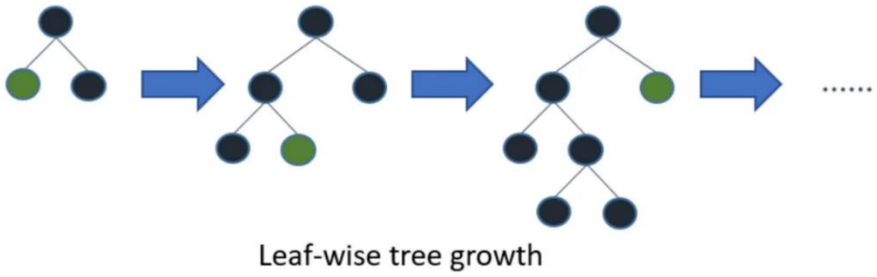
ภาพประกอบ 17 แสดงขั้นตอนการสร้างแบบจำลองด้วย Adaboost

ที่มา: https://pub.towardsai.net/all-about-adaboost-ba232b5521e9

ภาพประกอบ 17 แนวคิดในการสร้างแบบจำลองด้วย Adaboost จะเป็นการนำผลลัพธ์ในการทำนายที่ทำนายถูก และทำนายผิดกลุ่มข้อมูลของแบบจำลองมาทำการปรับน้ำหนัก (reweighting) โดยข้อมูลที่ทำนายผิดจะถูกทำการเพิ่มน้ำหนักแล้วจึงนำข้อมูลการทำนายใหม่จนสุดท้ายได้แบบจำลองที่ให้ผลลัพธ์สูงสุดออกมา

2.3.6 Light Gradient Boosting Machine (LightGBM)

LightGBM ถูกพัฒนามาจาก Gradient Boosting ถูกปรับปรุงโดยการให้สามารถประมวลผลข้อมูลที่มีคุณลักษณะจำนวนมาก หลักการทำงานคือใช้รูปแบบของ Leaf-wise Algorithm ซึ่งจะทำการแตกโหนด (Node) ออกมาหากตรงตามเงื่อนไขที่กำหนดไว้ เช่น ค่า ค่าความสูญเสีย (Loss) ทำให้ค่าความผิดพลาดจากการทำนายลดลงด้วย และยังทำงานได้เร็วกว่า Level-wise Algorithm เพราะไม่ต้องแตกโหนดที่ไม่จำเป็นออกมา



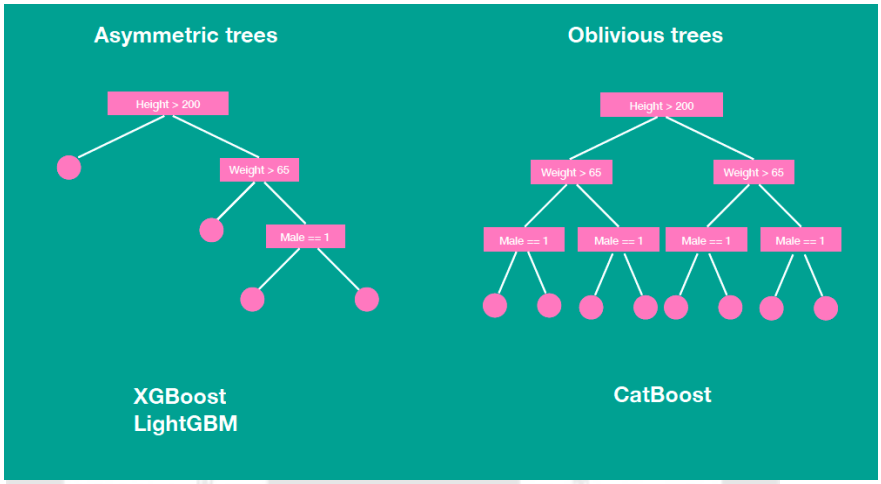
ภาพประกอบ 18 การแตกโหนดด้วยเทคนิค Leaf-wise tree growth ของ LightGBM

ที่มา: https://www.finnomena.com/finnomena-ic/light-gradient-boosting-machine-model/

จากภาพประกอบ 18 แบบจำลองจะทำการค้นหาตัวแปรที่มีความสำคัญ โดยจะทำการตรวจสอบผลลัพธ์ของการแบ่งกลุ่มระหว่างการแบ่งกลุ่มกับไม่แบ่งกลุ่มสถานการณ์ไหนให้ผลลัพธ์ที่ดีกว่า หากไม่ได้ผลลัพธ์ที่ดีกว่าแบบจำลองจะไม่ทำการแบ่งกลุ่ม หากมีการแบ่งกลุ่มจะทำการแบ่งข้อมูลต่อไปเรื่อย ๆจนถึงเงื่อนไขที่ได้มีการกำหนดไว้

2.3.6 CatBoost

CatBoost มีพื้นฐานการทำงานมาจาก Gradient Boost algorithm จึงมีหลักการทำงานคือสร้างแบบจำลอง Decision Tree มาทำงานต่อกัน การแตกโหนดมีลักษณะ Oblivious Trees หรือ Symmetric Tree ทำให้ Catboost สามารถประมวลผลได้อย่างรวดเร็วและทำนายผลได้อย่างแม่นยำ



ภาพประกอบ 19 เปรียบเทียบการทำงานของ CatBoost และ Boosting method อื่น ๆ

ที่มา: https://towardsdatascience.com/introduction-to-gradient-boosting-on-decision-trees-with-catboost-d511a9ccbd14

จากภาพประกอบ 19 CatBoost จะมีการให้ความสำคัญกับลำดับข้อมูล เพื่อป้องกันการเกิด Overfitting และระหว่างนั้นจะปรับปรุงข้อมูลไปด้วยตามลำดับ และข้อดีของ CatBoost อีกอย่างคือมีการทำ Pre-processing กับข้อมูลโดยอัตโนมัติ เช่น การทำ Encoding ให้กับข้อมูลจัดเป็นกลุ่ม, มีการทำ Auto-Tuning ในการสร้างแบบจำลองอัตโนมัติ ทำให้ตัวแบบจำลองมีประสิทธิภาพที่ดีที่สูงสุด รวมถึงจัดการกับ ข้อมูลที่ขาดหาย (Missing value) โดยอัตโนมัติ (Pattayapon, 2023)

2.4 ทฤษฎีที่เกี่ยวกับการเตรียมข้อมูล

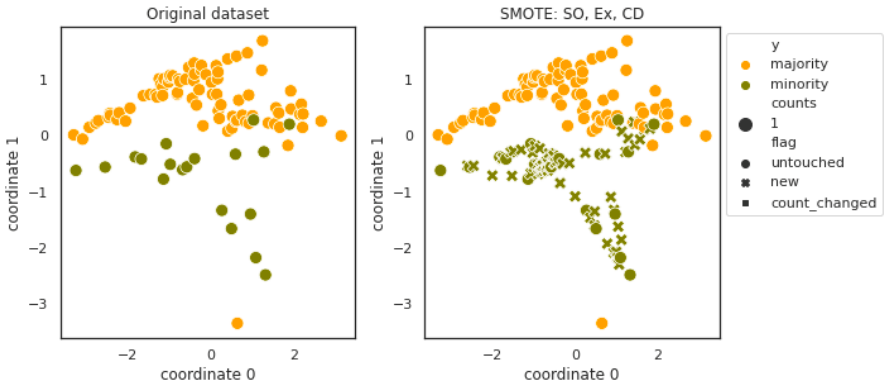
2.4.1 ปัญหาข้อมูลไม่สมดุล

ข้อมูลไม่สมดุล คือ ข้อมูลที่มีจำนวนกลุ่มข้อมูลหนึ่งมากกว่าอีกกลุ่มข้อมูลหนึ่งเป็นจำนวนมาก และเป็นปัญหาเฉพาะการสร้างแบบจำลองจำแนก ซึ่งปัญหาอาจเกิดจากข้อจำกัดในการเก็บข้อมูลมีต้นทุนที่สูง เช่น การเก็บจำนวนผู้ป่วยที่เป็นโรคหลอดเลือดสมอง ซึ่งมีจำนวนน้อยกว่ามากเมื่อเทียบกับผู้ป่วยปกติ

การที่ข้อมูลมีปัญหาข้อมูลไม่สมดุล จะส่งผลต่อการจำแนกข้อมูลจำนวนกลุ่มน้อย เพราะการสร้างแบบจำลองทั่วไปจะมีประสิทธิภาพก็ต่อเมื่อจำนวนของข้อมูลแต่ละกลุ่มมีจำนวนใกล้เคียงกัน การที่ข้อมูลมีลักษณะไม่สมดุลจะทำให้อัลกอริทึมมีความอ่อนไหว (Sensitive) ต่อจำนวนข้อมูลกลุ่มใหญ่มากกว่า เนื่องจากแบบจำลองจะตรวจจับข้อมูลที่มีจำนวนน้อยกว่าเป็นไปได้ยาก เป็นเพราะข้อมูลที่ใช้ในการเรียนรู้มีน้อย

2.4.1.1 **การเพิ่มจำนวนข้อมูล** (Oversampling)

การเพิ่มจำนวนข้อมูลให้กลุ่มข้อมูลที่มีจำนวนน้อย ให้มีจำนวนเท่ากับกลุ่มจำนวนข้อมูลใหญ่ แต่เป็นการเพิ่มจำนวนข้อมูลแบบสุ่ม ซึ่งจะทำให้เกิดปัญหาตามมาทีหลังคือ เกิดข้อมูลซ้ำเป็นจำนวนมาก ทำให้แบบจำลองที่สร้างเกิดปัญหา Overfitting ได้ ต่อมาจึงมีการแก้ปัญหา โดยการใช้วิธี SMOTE ซึ่งเป็นการสร้างข้อมูลขึ้นมาใหม่ โดยการอ้างอิงข้อมูลที่อยู่ใกล้กัน เพื่อป้องกันการเกิด Overfitting จากการสุ่มข้อมูล ดังภาพประกอบ 20

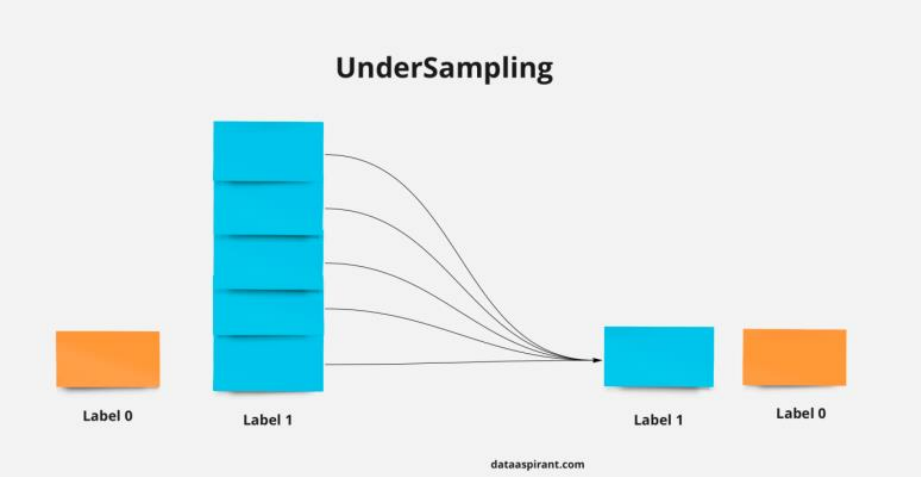


ภาพประกอบ 20 การแก้ปัญหาข้อมูลไม่สมดุลแบบ Oversampling เทียบกับ SMOTE

ที่มา: https://smote-variants.readthedocs.io/en/latest/oversamplers.html

2.4.1.2 **การลดจำนวนข้อมูล** (Undersampling)

การลดจำนวนข้อมูลที่มีกลุ่มจำนวนข้อมูลจำนวนมากด้วยวิธีการสุ่ม เพื่อให้กลุ่มข้อมูลจำนวนมากมีกลุ่มจำนวนข้อมูลน้อย แต่วิธีนี้อาจทำให้เกิดการตัดข้อมูลที่สำคัญออกไป ทำให้ประสิทธิภาพของแบบจำลองลดลง ดังภาพประกอบ 21



ภาพประกอบ 21 การแก้ปัญหาข้อมูลไม่สมดุลด้วยวิธีการลดจำนวนข้อมูล

ที่มา: https://neptune.ai/blog/how-to-deal-with-imbalanced-classification-and-regression-data

2.4.1.3 Class Weights

เทคนิคที่ใช้ในการจัดการข้อมูลไม่สมดุลด้วยการปรับค่าความผิดพลาดจากการทำนาย (error) ของการทำนายกลุ่มข้อมูลจำนวนน้อย ซึ่งเป็นกลุ่มข้อมูลที่เราสนใจของแบบจำลองด้วยการใช้ Log loss function เพื่อให้แบบจำลองมีการทำนายกลุ่มข้อมูลที่เราสนใจมากขึ้น ดังสมการที่ 7

$$log\ loss = \frac{1}{N}\sum_{i=1}^{N}[-(y_i * log(\widehat{y_i}) + (1-y_i) * log(1-\widehat{y_i})) \quad (7)$$

โดยที่

N คือ จำนวนข้อมูลทั้งหมด

y_i คือ กลุ่มข้อมูลที่เราสนใจ

$\hat{y}$ คือ ความน่าจะเป็นของกลุ่มข้อมูลที่เราสนใจ

จากสมการที่ 7 จะต้องมีการสังเกตค่าความน่าจะเป็นของกลุ่มข้อมูลที่เราสนใจ ที่จะนำมาใช้ในการคำนวณกับ Log loss function ผลที่ได้คือค่า Penalty Cost สำหรับการนำไปใช้เพื่อปรับค่าน้ำหนัก

ตาราง 2 ค่าของ Penalty Cost ในแต่ละจุดค่าความน่าจะเป็น

Actual Values	Predicted Prob. (Class 1)	Penalty Cost
0	0.32	0.385
0	0.18	0.198
0	0.28	0.328
0	0.12	0.127
0	0.08	0.083
1	0.44	0.820
0	0.24	0.274
0	0.01	0.010
0	0.22	0.248
0	0.16	0.174

ที่มา: (K. Singh, 2023)

จากตาราง 2 นำค่า Penalty Cost ที่ได้ไปใช้ในการปรับค่าน้ำหนักด้วยการใช้ Weight log loss function ซึ่งจะมีการนำค่าน้ำหนักของกลุ่มจำนวนข้อมูลมาใช้ในการคำนวณด้วย (K. Singh, 2023) ดังสมการที่ 8

$$log\ loss = \frac{1}{N}\sum_{i=1}^{N}[-(w_0(y_i * log(\widehat{y_i}))+w_1((1-y_i)*log(1-\widehat{y_i})))] \quad (8)$$

โดยที่

w_0 คือ น้ำหนักของกลุ่มข้อมูล 0

w_1 คือ น้ำหนักของกลุ่มข้อมูล 1

ผลที่ได้จากการนำค่า Penalty Cost เข้าสมการ Weight log loss function คือ จะทำให้ค่าความผิดพลาดของกลุ่มข้อมูล 0 ที่เป็นกลุ่มที่เราไม่สนใจมีค่าที่ต่ำลง และค่าความผิดพลาดของกลุ่มข้อมูลที่ 1 ที่เป็นกลุ่มข้อมูลที่เราสนใจมีค่าสูงขึ้น ดังตาราง 3

ตาราง 3 ค่า Penalty Cost หลังจากคำนวณ Weight log loss function ของกลุ่มข้อมูลของแต่ละจุดความน่าจะเป็น

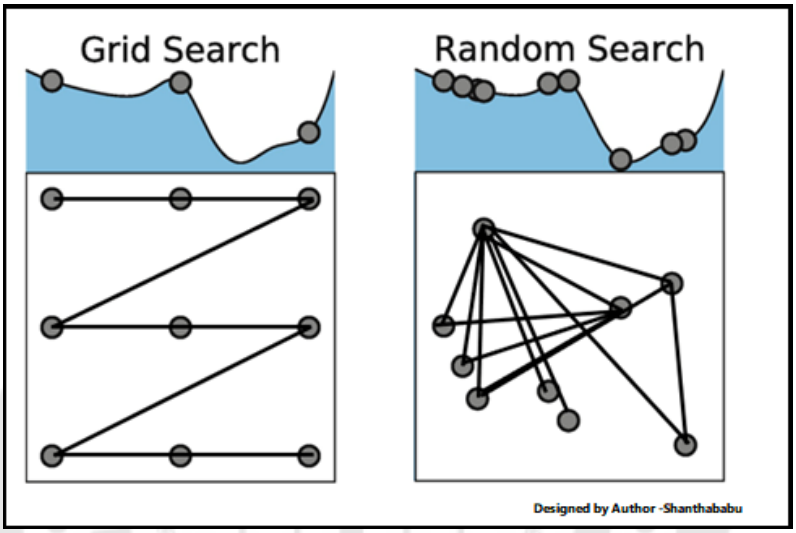
Actual Values	Predicted Prob. (Class 1)	Penalty Cost
0	0.32	0.211
0	0.18	0.109
0	0.28	0.180
0	0.12	0.069
0	0.08	0.045
1	0.44	4.104
0	0.24	0.150
0	0.01	0.005
0	0.22	0.136
0	0.16	0.095

ที่มา: (K. Singh, 2023)

จากตาราง 3 ในแต่ละความน่าจะเป็น ค่า Penalty Cost หลังจากปรับน้ำหนักกับจำนวนข้อมูลของกลุ่มข้อมูล ค่า Penalty Cost ของกลุ่มข้อมูลจำนวนมากจะมีค่าที่ต่ำลง ทำให้เมื่อนำไปให้แบบจำลองเรียนรู้ใหม่จะถูกลดความสำคัญลง ส่งผลให้กลุ่มข้อมูลจำนวนน้อยมีโอกาสที่จะถูกเรียนรู้มากขึ้นกว่าเดิม จากการที่ค่า Penalty Cost ที่เพิ่มขึ้นอย่างมีนัยสำคัญ ทำให้แบบจำลองมีความสามารถที่จะทำนายกลุ่มข้อมูลจำนวนน้อยเพิ่มขึ้นด้วย (K. Singh, 2023)

2.4.2 RandomizedSearch

Random Search CV เป็นหนึ่งในเครื่องมือของ Scikit-Learn ใช้ในการปรับแต่งแบบจำลอง โดยการสุ่มค่า Hyperparameter เพื่อให้แบบจำลองมีประสิทธิภาพที่ดีขึ้น ผลลัพธ์ที่ได้จากการใช้ Random Search ในบางครั้งอาจจะไม่ใช้ค่าที่ดีที่สุด แต่เป็นค่าที่สามารถทำให้แบบจำลองมีประสิทธิภาพที่ดีขึ้น



ภาพประกอบ 22 เปรียบเทียบการใช้อัลกอริทึม Grid Search และ Random Search

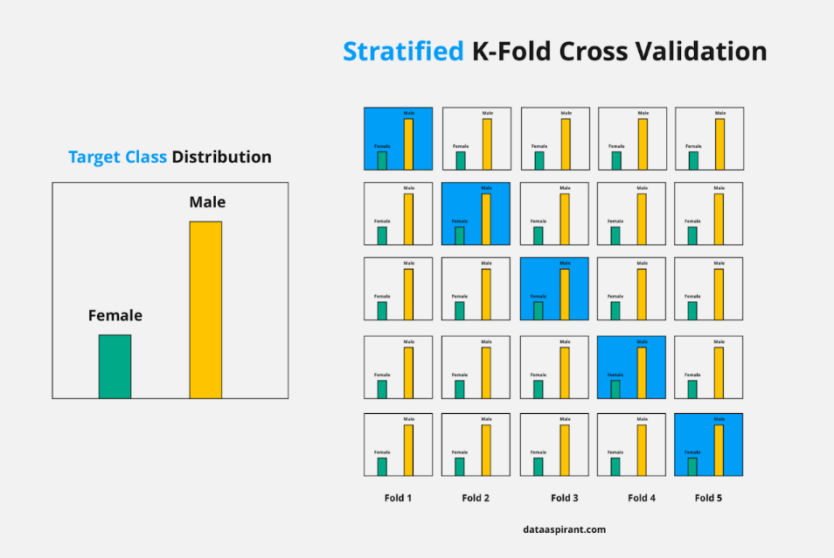
ที่มา: (Pandian, 2022)

จากภาพประกอบ 22 การใช้ Random Search และ Grid Search จะเหมือนกันคือ ต้องกำหนดขอบเขตของตัวแปรของ Hyperparameter ในแต่ละอัลกอริทึม Grid Search จะทำงานโดยเรียงลำดับตัวแปรของ Hyperparameter ค่อย ๆทีละตัวแปรจนครบทุกตัวแปร ซึ่งต่างจาก Random Search ที่จะทำการสุ่ม Hyperparameter แล้วนำไปปรับให้กับอัลกอริทึม

2.4.3 Stratified K-Fold Cross validation

ขั้นตอนในการประเมินความแม่นยำของอัลกอริทึม เพื่อป้องกันการเกิด Overfitting โดยการพยายามใช้ข้อมูลในการเรียนรู้ให้มากที่สุดในชุดข้อมูล ในการทำ Cross validation จะทำการแบ่งข้อมูลเป็น ชุดข้อมูลฝึกฝน และชุดข้อมูลตรวจสอบ และจะสลับข้อมูลที่ใช้ในการเรียนรู้ไปจนครบทั้งชุดข้อมูลแล้วเรียนรู้ซ้ำ ตามจำนวน K-Fold ที่กำหนดไว้ ทำให้อัลกอริทึมเจอข้อมูลที่

หลากหลาย โดยข้อมูลที่อัลกอริทึมเจอสามารถเจอได้ทั้งข้อมูลฝึกฝน และข้อมูลตรวจสอบ ซึ่งการทำ Cross validation แบบปกติ จะมีปัญหาเมื่อข้อมูลมีลักษณะข้อมูลไม่สมดุล เพราะกลุ่มข้อมูลจำนวนน้อย มีไม่เพียงพอ ทำให้ในบาง Fold แบ่งกลุ่มข้อมูลได้ไม่ครบทุกกลุ่ม หรือมีกลุ่มข้อมูลเดียวกันทั้งหมด



ภาพประกอบ 23 การแบ่งข้อมูลด้วย Stratified K-Fold

ที่มา: (Sachinsoni, 2023)

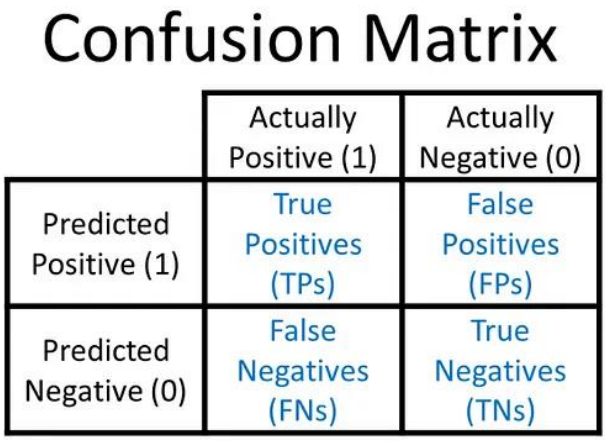
จากภาพประกอบ 23 สามารถใช้ Stratified K-Fold ซึ่งเป็นวิธีการแบ่งข้อมูลเป็น K-Fold เช่นเดียวกับ Cross validation ปกติ แต่จะแบ่งข้อมูลให้ครบทุกกลุ่มข้อมูล ในทุก ๆ Fold เพื่อให้อัลกอริทึมสามารถเรียนรู้ได้ทุกกลุ่มข้อมูล

2.5 ทฤษฎีการประเมินประสิทธิภาพแบบจำลอง

การประเมินประสิทธิภาพของแบบจำลอง มีความจำเป็นในการสร้างแบบจำลองเป็นอย่างมาก เพราะเป็นการวัดความสามารถแบบจำลองว่ามีการทำงานเป็นอย่างไร มีการประเมินที่พึงพอใจหรือไม่ และในการเลือกตัววัดประสิทธิภาพของแบบจำลอง ควรเลือกให้เหมาะสมกับประเภทของแบบจำลอง ละชุดข้อมูล

การสร้างแบบจำลองจำแนกสองกลุ่มข้อมูล หากข้อมูลมีปัญหาข้อมูลไม่สมดุล อาจไม่สามารถใช้เพียงตัววัดประสิทธิภาพ ความแม่นยำ (Accuracy) อย่างเดียวอาจไม่เพียงพอ

เนื่องจากแบบจำลองที่สร้างอาจทำนายข้อมูลกลุ่มข้อมูลจำนวนมาก ให้ถูกต้องเพียงกลุ่มเดียว ก็ทำให้ตัววัดประสิทธิภาพความแม่นยำมีค่าที่สูงมาก และถึงแม้ว่าจะทำนายข้อมูลส่วนน้อยผิดพลาด ค่าความแม่นยำก็ลดลงไม่มาก เพราะสัดส่วนข้อมูลที่มีจำนวนที่น้อยกว่ามาก ดังนั้นในการวัดประสิทธิภาพของแบบจำลองในปัญหาแบบการจำแนกประเภทจะประกอบด้วยดังต่อไปนี้



ภาพประกอบ 24 รูปแบบของ Confusion Matrix

ที่มา: https://medium.com/@pagongatchalee/confusion-matrix-เครื่องมือสำคัญในการประเมินผลลัพธ์ของการทำนาย-ในmachine-learning-fba6e3f9508c

ภาพประกอบ 24 Confusion Matrix เป็นเครื่องมือสำคัญในการประเมินประสิทธิภาพของแบบจำลองที่สร้างขึ้น มีหลักการจากการวัดที่สิ่งแบบจำลองทำนายกับสิ่งที่เกิดขึ้นจริงมีสัดส่วนเป็นอย่างไร โดยตัวเลขที่อยู่ในตาราง คือค่าความถี่ของจำนวนข้อมูลจากการทำนายของแบบจำลองกับที่เกิดขึ้นจริง

รายละเอียดความหมายของตาราง Confusion Matrix มีดังนี้

True Positives (TP) คือ จำนวนข้อมูลที่ถูกจำแนกได้อย่างถูกต้องในกลุ่มของประเภทที่เราสนใจ (Positive class) ใน กรณีที่แบบจำลองทำนายทำนายว่าเป็นกลุ่มข้อมูลที่เราสนใจ คือการทำนายถูกต้อง

True Negatives (TN) คือ จำนวนข้อมูลที่ถูกจำแนกถูกต้องในกลุ่มของประเภทที่เราไม่สนใจ (Negative class) กรณีนี้คือข้อมูลที่ถูกทำนายว่าเป็นกลุ่มข้อมูลที่เราไม่สนใจ คือการทำนายถูกต้อง

False Positives (FP) คือ จำนวนข้อมูลที่ถูกจำแนกผิดในกลุ่มข้อมูลที่เราสนใจ ในกรณีนี้คือข้อมูลที่ถูกทำนายว่าเป็นกลุ่มที่เราสนใจ แต่ข้อมูลจริงเป็นกลุ่มที่เราไม่สนใจ นั่นคือการทำนายผิดว่าเป็นสิ่งที่เราทำการประเมินไว้

False Negatives (FN) : คือจำนวนข้อมูลที่ถูกจำแนกผิดในกลุ่มของประเภทที่เราไม่สนใจ ในกรณีนี้คือข้อมูลที่ถูกทำนายว่าเป็นกลุ่มที่เราไม่สนใจ แต่ข้อมูลจริงกลับเป็นกลุ่มที่เราสนใจ นั่นคือการทำนายผิดว่าไม่เป็นสิ่งที่เราประเมินไว้ (Gatchalee, 2019)

เราสามารถนำข้อมูลค่าความถี่จาก Confusion Matrix มาใช้ในการคำนวณวัดประสิทธิภาพของแบบจำลอง ในรูปแบบต่าง ๆได้ดังนี้

ความแม่นยำ (Accuracy) คือ ค่าความถูกต้องของการทำนายผลทั้งหมดของแบบจำลอง ดังสมการที่ 9

$$Accuracy = \frac{(TP+TN)}{(TP+TN+FP+FN)} \quad (9)$$

ค่าความอ่อนไหว (Recall (Sensitivity)) คือ สัดส่วนที่แบบจำลองทำนายประเด็นที่เราสนใจทั้งหมด ดังสมการที่ 10

$$Recall = \frac{TP}{(TP+FN)} \quad (10)$$

ค่าความถูกต้อง (Precision) คือ ความถูกต้องของประเด็นที่แบบจำลองสนใจที่จะทำนาย ดังสมการที่ 11

$$Precision = \frac{TP}{(TP+FP)} \quad (11)$$

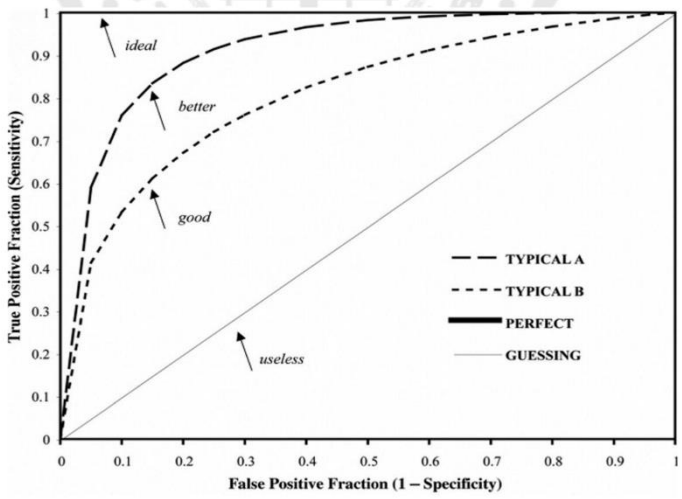
F1-Score คือ ค่าเฉลี่ยของค่าความถูกต้อง และค่าความอ่อนไหว ดังสมการที่ 12

$$F1\ Score = 2\frac{(Precision\ x\ Recall)}{(Precision + Recall)} \quad (12)$$

ค่าความจำเพาะ (Specificity) คือ สัดส่วนความถูกต้องของผลการทำนายของกลุ่มข้อมูลที่เราไม่ได้สนใจ ดังสมการ 13

$$Specificity = \frac{TN}{(TN + FP)} \quad (13)$$

ROC Curve (Receiver Operating Characteristic) คือ ตัวบ่งชี้ประสิทธิภาพของแบบจำลองที่ทำนายกลุ่มข้อมูลที่เราสนใจ เทียบกับสัดส่วนกลุ่มข้อมูลที่เราไม่ได้สนใจ



ภาพประกอบ 25 แนวคิดการวัด Receiver Operating Characteristic (ROC Curve)

ที่มา: https://pub.aimind.so/unveiling-the-power-of-roc-curves-and-auc-your-guide-to-evaluating-model-performance-in-plain-d961c32c18c0

จากภาพประกอบ 25 ROC Curve (Receiver Operating Characteristic) คือ ตัวบ่งชี้ประสิทธิภาพของแบบจำลองที่ทำนายกลุ่มข้อมูลที่เราสนใจ เทียบกับสัดส่วนกลุ่มข้อมูลที่เราไม่ได้สนใจในทุก ๆเกณฑ์ (Threshold) ที่แบบจำลองทำนายเหมาะกับการนำไปใช้ในการหาค่าที่สูงที่สุดที่แบบจำลองทำนายได้

2.6 งานวิจัยที่เกี่ยวข้อง

การทบทวนวรรณกรรมของงานวิจัยนี้ได้ทำการศึกษาค้นคว้างานวิจัยที่เกี่ยวข้องกับการสร้างแบบจำลองเพื่อจำแนกผู้ป่วยโรคหลอดเลือดสมอง โดยงานวิจัยที่เกี่ยวข้องมีรายละเอียด ดังนี้

2.6.1 **บทความวิจัยเรื่อง** A Study of Stroke Prevalence Prediction Based on Random Forest Algorithm (Shan et al., 2023)

งานวิจัยนี้ทำการศึกษาค้นคว้าการสร้างแบบจำลองจำแนกผู้ป่วยโรคหลอดเลือดสมอง มีการใช้คุณลักษณะ ที่ใช้ในสร้างแบบจำลองทั้งหมด 10 Feature แบ่ง Feature ได้ 2 ประเภท คือ 1.) ข้อมูลจัดเป็นกลุ่ม ประกอบด้วย heart disease, marital status, gender, smoking status, work type, type of residence 2.) ข้อมูลเชิงตัวเลข ประกอบด้วย age, average glucose level, body mass index มีกลุ่มข้อมูล คือ Stroke มีผลลัพธ์คือ ผู้ป่วยโรคหลอดเลือดสมอง และผู้ป่วยปกติ ข้อมูลมีลักษณะไม่สมดุลใช้วิธี SMOTE ในการแก้ปัญหา

การพัฒนาแบบจำลองจำแนกผู้มีความเสี่ยงโรคหลอดเลือดสมอง โดยใช้ Logistic regression, Random Forest, Support Vector Machine ทดสอบโดยการใช้ตัววัดประสิทธิภาพ ค่าความแม่นยำ, ค่าความถูกต้อง, ค่าความอ่อนไหว และ F1-score

ผลการวิจัยพบว่า Random Forest ได้ค่าตัวประสิทธิภาพความแม่นยำเท่ากับ 96.94%, ค่าความถูกต้อง 98.91%, ค่าความอ่อนไหว 94.96%, F1-score 97% ซึ่งเป็นค่าที่สูงเพียงพอต่อการตรวจจับ ผู้มีความเสี่ยงโรคหลอดเลือดสมอง

2.6.2 Stroke Risk Prediction with Machine Learning Techniques (Dritsas & Trigka, 2022)

งานวิจัยนี้ได้ทำการศึกษาการสร้างแบบจำลองเพื่อจำแนกผู้มีความเสี่ยงโรคหลอดเลือดสมอง และไม่มีความเสี่ยงโรคหลอดเลือดสมอง โดยใช้ชุดข้อมูลจาก Kaggle ชุดข้อมูลมีคุณลักษณะที่ใช้ในการสร้างแบบจำลอง 10 Feature คือ age, gender, hypertension, heart_disease, ever_married, work_type, residence type, avg glucose level, BMI, Smoking Status โดยมีกลุ่มข้อมูลเป็น Stroke มีผลลัพธ์คือ ผู้ป่วยโรคหลอดเลือดสมอง และผู้ป่วยปกติ โดยจะให้ความสนใจกับข้อมูลที่มีอายุมากกว่า 18 ปีขึ้นไป

มีการจัดการกับข้อมูล เช่น การจัดการข้อมูลที่ขาดหาย หรือข้อมูลรบกวน (Noisy data), ใช้ Feature selection เพื่อลดความซ้ำซ้อนของข้อมูล ฯลฯ ชุดข้อมูลมีลักษณะเป็นข้อมูลไม่สมดุล จึงใช้วิธี SMOTE ในการแก้ปัญหาข้อมูล

การพัฒนาแบบจำลองงานวิจัยใช้อัลกอริทึม 9 อัลกอริทึม ประกอบด้วย Naive Bayes, Random Forest, Logistic Regression, K-Nearest Neighbors, Stochastic Gradient Descent, Decision Tree, Multilayer Perceptron, Majority Voting และ Stacking เพื่อใช้ในการเปรียบเทียบประสิทธิภาพของแบบจำลอง ทดสอบโดยใช้ตัววัดประสิทธิภาพ ค่าความถูกต้อง, ค่าความอ่อนไหว, F-Measure, AUC Curve และ ค่าความแม่นยำ

ผลจากการวิจัยพบว่าตัววัดประสิทธิภาพ ค่าความถูกต้อง, ค่าความอ่อนไหว, F-Measure ของอัลกอริทึมให้ผลลัพธ์ใกล้เคียงกัน ยกเว้น Random Forest และ Stacking Classifier ที่ให้ค่ามากกว่าอัลกอริทึมอื่นอยู่มาก ซึ่ง AUC ของอัลกอริทึม Stacking classification ให้ผล AUC สูงสุดอยู่ที่ 98.9%

2.6.3 Finding the Best Classification Threshold in Imbalanced Classification (Zou et al., 2016)

งานวิจัยนี้ทำการศึกษาเกี่ยวกับการหาผลลัพธ์ที่เหมาะสมที่สุดของชุดข้อมูลที่มีลักษณะชุดขอมูลไม่สมดุล เนื่องจากแบบจำลองมักจะมีผลการจำแนกกลุ่มข้อมูลจำนวนมาก ได้แม่นยำกว่า กลุ่มข้อมูลจำนวนน้อย เพราะสัดส่วนจำนวนข้อมูลที่ใช้ในการฝึกฝนในการทำนายมีจำนวนน้อยกว่า

การสร้างแบบจำลองโดยมาตรฐานจะมีการตั้งเกณฑ์ผลลัพธ์ความน่าจะเป็นในการทำนายกลุ่มข้อมูลที่เท่ากัน Threshold อยู่ที่ 0.5 ทำให้แบบจำลองใช้งานได้ไม่ดีกับชุดข้อมูลไม่สมดุล ดังนั้นงานวิจัยนี้จึงได้มีการทดสอบใช้ค่า Threshold ความน่าจะเป็นของการจำแนกของแบบจำลอง ถ้ามากกว่า 0.5 จะอยู่ในกลุ่มข้อมูลจำนวนน้อย และถ้าน้อยกว่าเท่ากับ 0.5 เป็นกลุ่มข้อมูลจำนวนมาก

จากการทดลองโดยใช้ชุดข้อมูล Liao’s protein remote homology detection พบว่าชุดข้อมูลฝึกฝน ความน่าจะเป็น Threshold ที่ 0.79 เมื่อนำไปทดสอบกับข้อมูล Test set จะให้ค่าค่า Precision ที่สูงที่สุดเท่ากับ 1 ซึ่งมากกว่า จุดที่ Threshold ความน่าจะเป็น 0.5 มี Precision มีค่าเท่ากับ 0.7

2.6.4 The balanced accuracy and its posterior distribution(Brodersen et al., 2010)

งานวิจัยนี้ได้ทำการศึกษาข้อจำกัดของตัววัดประสิทธิภาพแบบจำลอง ที่ขาดความสามารถในการตรวจจับ และการประเมินที่ดีกับชุดข้อมูลที่เป็นชุดข้อมูลไม่สมดุล ดังนั้นจึงต้องมีการแทนที่ตัววัดประสิทธิภาพความแม่นยำ โดยการกระจายความแม่นยำด้วยความแม่นยำสมดุล

ตัววัดความแม่นยำของแบบจำลองบ่อยครั้งจะให้ค่าที่สูงเกินความเป็นจริง โดยเฉพาะกับชุดข้อมูลที่มีลักษณะชุดข้อมูลสมดุล ซึ่งในชุดข้อมูลฝึกสอนแบบจำลองจะให้ความสำคัญกับข้อมูลที่มีความถี่เยอะกว่า ส่งผลให้ค่าความแม่นยำในชุดข้อมูลทดสอบมีค่าที่สูงมาก ดังนั้นจึงต้องมีการใช้ความแม่นยำสมดุลเข้ามาทดแทน ซึ่งจะใช้งานได้ดีกับชุดข้อมูลไม่สมดุล และกรณีที่ชุดข้อมูลมีจำนวนกลุ่มข้อมูลที่สมดุลกัน จะทำให้ค่าความแม่นยำสมดุลให้ผลลัพธ์ที่ต่ำกว่าค่าความแม่นยำ

จากการทดสอบตัวอย่างแรก ชุดที่ใช้ในการเรียนรู้จะให้ชุดข้อมูลฝึกฝนมีความสมดุล โดยมีจำนวนข้อมูลที่สนใจ 70 รายการ และไม่สนใจ 70 รายการ ผลการทำนายของแบบจำลองที่ของข้อมูลที่สนใจคือ 69 รายการและไม่สนใจ 69 รายการ มีการจำแนกได้ถูกต้องเกือบ 100% คือทำให้ผลที่ได้จากการวัดประสิทธิภาพความแม่นยำ และความแม่นยำสมดุลให้ผลลัพธ์ใกล้เคียงกัน ซึ่งต่างจากการทดสอบตัวอย่างที่สอง ที่ให้ชุดข้อมูลไม่สมดุลมีจำนวนข้อมูลที่สนใจ 45 รายการ และไม่สนใจ 10 รายการ การทำนายของแบบจำลองมีการแบ่งข้อมูลฝึกฝนสำหรับทำนายไปที่ข้อมูลที่เราสนใจมากกว่าที่จำนวน 48 รายการ และไม่สนใจ 7 รายการ ผลการทำนายข้อมูลที่ถูกต้องคือ 40 รายการ และ 2 รายการ ส่งผลให้ความแม่นยำมีค่าที่ต่างจากความแม่นยำสมดุล

ผลจากการวิจัยพบว่าค่าของความแม่นยำ จะได้ประโยชน์จากการที่ข้อมูลไม่สมดุล ซึ่งทำให้ค่ามีค่าสูงกว่าที่ควรจะเป็น จึงมีความคิดที่จะแทนที่ด้วยการใช้ชุดข้อมูลไม่สมดุลในการวัดประสิทธิภาพของแบบจำลอง ในกรณีที่ชุดข้อมูลไม่สมดุล

2.7 **สรุปทฤษฎีและงานวิจัยที่เกี่ยวข้อง**

จากการศึกษาทฤษฎีและงานวิจัยที่เกี่ยวข้อง เพื่อนำมาใช้ในการสร้างแบบจำลองในงานวิจัยนี้ สามารถสรุปรายละเอียดจากหัวข้อต่าง ๆได้ ดังนี้

การศึกษาอัลกอริทึมของคอมมพิวเตอร์มี 3 ประเภท คือ การเรียนรู้ของเครื่อง, การเรียนรู้เชิงลึก และปัญญาประดิษฐ์ โดยการเลือกใช้อัลกอริทึมขึ้นอยู่กับประเภทของแบบจำลองที่สร้างและความซับซ้อนของปัญหาที่เจอ รวมถึงชุดข้อมูลที่นำมาใช้ในการสร้างแบบจำลอง ในงานวิจัยนี้

เป็นการสร้างแบบจำลองด้วยการเรียนรู้ของเครื่อง ซึ่งเป็นงานประเภทการเรียนรู้แบบ Supervised Learning รูปแบบการเรียนรู้เพื่อจำแนกข้อมูล

ชุดข้อมูลที่ใช้ในงานวิจัยนี้ มีลักษณะเป็นชุดข้อมูลไม่สมดุล ต้องใช้วิธีการแก้ปัญหาในการปรับสมดุลสัดส่วนจำนวนข้อมูลของแต่ละกลุ่มข้อมูลให้มีความเหมาะสมกับการนำไปใช้ในการสร้างแบบจำลอง โดยการใช้ 4 เทคนิค ประกอบด้วย การลดจำนวนข้อมูล, การเพิ่มจำนวนข้อมูล, SMOTE และ Class Weights หลังจากนั้นจึงนำข้อมูลที่ได้จากการปรับสมดุลของแต่ละกลุ่มข้อมูล ไปให้แบบจำลองเรียนรู้แล้วเปรียบเทียบประสิทธิภาพของแต่ละเทคนิคในแต่ละอัลกอริทึม

อัลกอริทึมที่นำมาใช้ในการสร้างแบบจำลองในงานวิจัยนี้ใช้ทั้งหมด 7 อัลกอริทึม ประกอบด้วย Logistic Regression, Decision Tree, Random Forest, XGBoost, AdaBoost, LightGBM และ CatBoost ข้อมูลที่นำไปใช้ให้แบบจำลองเรียนรู้จะใช้ข้อมูลที่ได้หลังจากการปรับสมดุลสัดส่วนของกลุ่มข้อมูลด้วยการจัดการข้อมูลไม่สมดุลแล้ว ทุก ๆแบบจำลองในแต่ละอัลกอริทึมจะถูกปรับหาค่า Parameter ที่ทำให้แบบจำลองมีประสิทธิภาพในการจำแนกสูงขึ้นด้วย RandomSearch

ขั้นตอนในการแบ่งข้อมูลที่ใช้ในการเรียนรู้ เพื่อคำนวณหาค่าประสิทธิภาพของแบบจำลองจะใช้ StratifiedKFold ในการแบ่งชุดข้อมูลเป็น 2 ส่วน คือ ชุดข้อมูลเรียนรู้ และชุดข้อมูลทดสอบ ซึ่งการใช้ StratifiedKFold จะทำให้ข้อมูลที่ถูกแบ่งเป็นชุดข้อมูลตามจำนวน Fold ที่เราตั้งไว้ให้เรียนรู้ และในทุกชุดข้อมูลที่ถูกแบ่ง จะมีจำนวนกลุ่มข้อมูลทุกกลุ่มในชุดข้อมูลที่เรียนรู้ ทำให้แบบจำลองได้เรียนรู้ทุกกลุ่มข้อมูลทุกกลุ่ม

การประเมินประสิทธิภาพของแบบจำลองที่ใช้สำหรับการจำแนกจะมีจุดเริ่มต้นจากการนำความถี่ที่เป็นผลลัพธ์ที่ได้จากการทำนายของแบบจำลอง มาประเมินประสิทธิภาพในแต่ละมุมมองต่าง ๆของกลุ่มข้อมูล ประกอบด้วย ค่าความแม่นยำ, ค่าความอ่อนไหว, F1 score, ค่าความจำเพาะ และ ROC curve

จากการศึกษางานวิจัยที่เกี่ยวข้องการสร้างแบบจำลองพบว่า แบบจำลองที่มีประสิทธิภาพในการจำแนกสูงจะมาจากการพัฒนาต้นแบบ Decision Tree ซึ่งจะทำให้ตัวแบบจำลองสามารถทำนายข้อมูลได้อย่างมีประสิทธิภาพ และชุดข้อมูลที่นำมาใช้กับแบบจำลองมีลักษณะข้อมูลไม่สมดุลส่วนใหญ่จะมีปัญหาเรื่องการเลือกใช้ตัววัดประสิทธิภาพและการทำนายข้อมูลจำนวนน้อย

ในส่วนปัญหาของชุดข้อมูลไม่สมดุลจะมี 2 ประเด็นหลัก คือ การที่แบบจำลองไม่ทำนายกลุ่มข้อมูลส่วนน้อย และการเลือกใช้ตัวประเมินประสิทธิภาพของแบบจำลอง จากการศึกษา

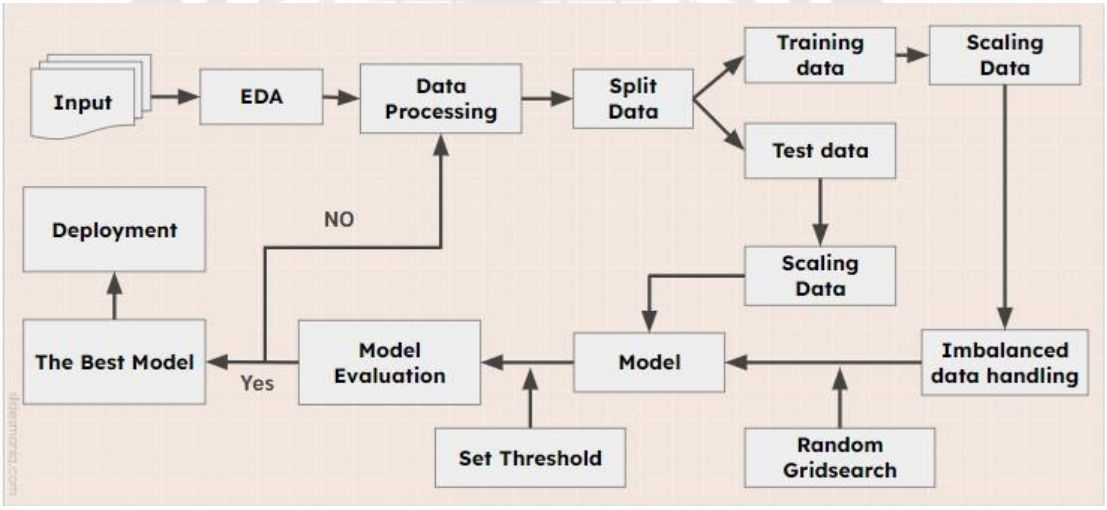
งานวิจัย Finding the Best Classification Threshold in Imbalanced Classification ใช้วิธีการปรับ Threshold ในการแก้ปัญหาการแบบจำลองไม่ทำนายข้อมูลจำนวนน้อย ในส่วนของการเลือกใช้ตัววัดประสิทธิภาพของแบบจำลองงานวิจัย The balanced accuracy and its posterior distribution หากมีการเลือกใช้ตัววัดประสิทธิภาพไม่เหมาะสมจะทำให้ได้ผลลัพธ์ของการประเมินออกมาสูงกว่าความเป็นจริง เพราะแบบจำลองจะให้ความสนใจในการทำนายกลุ่มข้อมูลจำนวนมากเพียงกลุ่มเดียวก็ทำให้ตัววัดประสิทธิภาพสูงแล้ว โดยไม่จำเป็นที่ต้องสนใจการทำนายข้อมูลจำนวนน้อย ทำให้การประเมินประสิทธิภาพของแบบจำลองไม่ได้สะท้อนความสามารถในการทำนายของแบบจำลองที่แท้จริง ดังนั้นในงานวิจัยนี้จึงมีการใช้ค่าความแม่นยำสมดุลในการวัดประสิทธิภาพ ที่มีการคำนวณจากกลุ่มข้อมูลจากกลุ่มข้อมูลจำนวนมาก และกลุ่มข้อมูลจำนวนน้อย ทำให้เห็นความสามารถในการจำแนกของแบบจำลองได้ดีกว่า

บทที่ 3
แนวคิดและวิธีวิจัย

ในงานวิจัยนี้ ผู้วิจัยได้ดำเนินการตามขั้นตอนดังนี้

1. กระบวนการทำงานของแบบจำลอง
2. การเก็บรวบรวมข้อมูล
3. การสำรวจข้อมูล (Exploratory Data Analysis: EDA) และการเตรียมข้อมูล (Pre-processing)
4. อัลกอริทึมของแบบจำลองการทำนาย
5. การประเมินผลแบบจำลอง
6. สรุปแนวคิดและวิธีวิจัย

3.1 กระบวนการทำงานของแบบจำลอง



ภาพประกอบ 26 กระบวนการทำงานของแบบจำลอง

จากภาพประกอบที่ 26 ได้อธิบายถึงกระบวนการสร้างแบบจำลองการทำนายโดนเริ่มจากขั้นตอนการนำเข้าข้อมูล การสำรวจข้อมูล เพื่อวิเคราะห์ความสัมพันธ์ระหว่างตัวแปรต่าง ๆ ที่เกี่ยวข้องกับการจำแนกผู้ป่วยโรคหลอดเลือดสมอง การเตรียมข้อมูล และคุณลักษณะที่เกี่ยวข้องกับการจำแนก โดยจะตัด Feature ID เนื่องจากไม่เกี่ยวข้องกับการทำนาย และลักษณะข้อมูลจัดเป็นกลุ่ม ประกอบด้วยคุณลักษณะ heart disease, marital status, gender, smoking status,

work type, type of residence จะถูกเปลี่ยนให้เป็นตัวเลขด้วยเทคนิค Encoder หลังจากนั้นจะถูกทำ Scaling เพื่อลดการกระจายตัวของข้อมูล พร้อมกับลักษณะข้อมูลเชิงตัวเลข ที่ประกอบด้วยคุณลักษณะ age, average glucose level, body mass index

ในการสร้างแบบจำลองทำนาย แบ่งข้อมูลออกเป็นชุดข้อมูลฝึกสอน และข้อมูลทดสอบ ในอัตราส่วน 80:20 สำหรับการจำแนกจะมีผลลัพธ์ 2 ประเภท คือ 0 : ผู้ป่วยปกติ และ 1 : ผู้ป่วยโรคหลอดเลือดสมอง จากนั้นนำข้อมูลฝึกสอนให้แบบจำลองเรียนรู้เพื่อสร้างแบบจำลองเรียนรู้เพื่อสร้างแบบจำลองการจำแนกผู้ป่วยโรคหลอดเลือดสมองในแบบการจำแนกสองกลุ่มข้อมูล ใช้เทคนิคการเรียนรู้ของเครื่องทั้งหมด 7 อัลกอริทึม คือ Logistic Regression, Decision Tree, Random Forest, XGBoost, Adaboost, LightGBM และCatboost และใช้เทคนิคการจัดการข้อมูลไม่สมดุล 3 เทคนิค เพื่อปรับสมดุลของข้อมูลในข้อมูลฝึกสอน ได้แก่ การสุ่มลดจำนวนตัวอย่าง, การสุ่มเพิ่มจำนวนตัวอย่าง และการเพิ่มจำนวนตัวอย่างด้วย SMOTE จากนั้นทำการปรับจูน Hyperparameter ของแบบจำลองให้ได้ประสิทธิภาพที่เหมาะสมด้วย Random search แบบจำลองทั้งหมดระหว่างฝึกสอนจะถูกตรวจสอบประสิทธิภาพด้วย Cross Validation ทั้งหมด 10-Fold โดยใช้การแบ่ง Fold ด้วยเทคนิค StratifiedKFold

สุดท้ายการทดสอบประสิทธิภาพของแบบจำลอง โดยใช้ Confusion Matrix ในการตรวจสอบผลลัพธ์ความถี่จากการทำนายของแบบจำลอง และใช้รายงาน Classification Report ประกอบด้วย ความถูกต้อง, ความอ่อนไหว, ความแม่นยำ, ROC Curve และ ความแม่นยำสมดุล ถูกสร้างขึ้นเพื่อประเมินผลแบบจำลองการทำนายด้วยข้อมูลทดสอบ ในงานวิจัยจะมุ่งเน้นผลลัพธ์ 2 ประการ ประการแรกคือ ค่าความแม่นยำสมดุล และประการที่ 2 คือผลลัพธ์ความถี่จากการจำแนกผู้ป่วยโรคหลอดเลือดสมองของแบบจำลอง หลังจากนั้นจะทำการปรับเกณฑ์ค่าความน่าจะเป็นของการทำนาย (Threshold) เพื่อให้แบบจำลองให้น้ำหนักกับการทำนายผู้ป่วยโรคหลอดเลือดสมองที่เป็นกลุ่มข้อมูลจำนวนน้อยมากขึ้น

3.2 การเก็บรวบรวมข้อมูล

ในงานวิจัยนี้ใช้ข้อมูล Stroke Prediction Dataset จากเว็บไซต์ Kaggle (Brodersen et al., 2010) ชุดข้อมูลมีคุณลักษณะ 12 รายการ ประกอบด้วย id, gender, age, hypertension, heart_disease, ever_married, work_type, Residence_type:, avg_glucose_level, bmi, smoking_status และมี stroke เป็นกลุ่มข้อมูล จำนวนข้อมูลมีทั้งหมด 5,110 แถว โดยรายละเอียดข้อมูลคุณลักษณะดังตาราง 4

ตาราง 4 รายละเอียดชุดข้อมูลเพื่อใช้ในการจำแนกผู้ป่วยโรคหลอดเลือดสมอง

ลำดับ	ข้อมูลตัวแปร (Variable)	คำอธิบายข้อมูล (Description)
1	id	ตัวระบุตำแหน่งข้อมูล
2	gender	เพศของผู้ป่วย
3	age	อายุของผู้ป่วย
4	hypertension	โรคความดันโลหิตสูง
5	heart_disease	โรคหัวใจ
6	ever_married	สถานะของผู้ป่วย
7	work_type	ประเภทอาชีพของผู้ป่วย
8	Residence_type	ตำแหน่งของที่อยู่อาศัยของผู้ป่วย
9	avg_glucose_level	ปริมาณน้ำตาลในเลือดของผู้ป่วย
10	bmi	ดัชนีความสมดุลร่างกายของผู้ป่วย
11	smoking_status	สถานะการสูบบุหรี่ของผู้ป่วย
12	stroke	อาการป่วยโรคหลอดเลือดสมองของผู้ป่วย

3.3 การสำรวจข้อมูลและการเตรียมข้อมูล

การสำรวจข้อมูลจะเริ่มจากวิเคราะห์จากตัวอย่างข้อมูลและสถิติเบื้องต้นของข้อมูล ดังภาพประกอบที่ 27

	id	age	hypertension	heart_disease	avg_glucose_level	bmi	stroke
count	5110.000000	5110.000000	5110.000000	5110.000000	5110.000000	4909.000000	5110.000000
mean	36517.829354	43.226614	0.097456	0.054012	106.147677	28.893237	0.048728
std	21161.721625	22.612647	0.296607	0.226063	45.283560	7.854067	0.215320
min	67.000000	0.080000	0.000000	0.000000	55.120000	10.300000	0.000000
25%	17741.250000	25.000000	0.000000	0.000000	77.245000	23.500000	0.000000
50%	36932.000000	45.000000	0.000000	0.000000	91.885000	28.100000	0.000000
75%	54682.000000	61.000000	0.000000	0.000000	114.090000	33.100000	0.000000
max	72940.000000	82.000000	1.000000	1.000000	271.740000	97.600000	1.000000

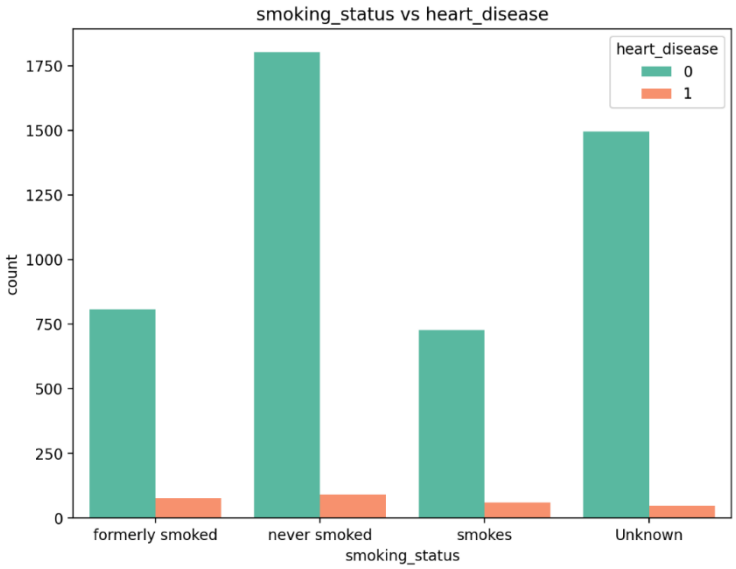
ภาพประกอบ 27 ข้อมูลสถิติพื้นฐานของชุดข้อมูล

จากภาพประกอบที่ 27 สถิติพื้นฐานแสดงข้อมูลของคุณลักษณะที่เป็นตัวเลขเท่านั้น และจากข้อมูลจะเห็นได้ว่าคุณลักษณะทุกตัวจะมีจำนวนข้อมูลทั้งหมด 5,110 แถว ยกเว้น BMI ที่มีเพียง 4,909 แถว เป็นผลจากการที่ข้อมูลมีข้อมูลไม่ครบทุกแถว

```
smoking_status
never smoked       1892
Unknown            1544
formerly smoked     885
smokes              789
Name: count, dtype: int64
```

ภาพประกอบ 28 ข้อมูลผู้สถานะการสูบบุหรี่ของผู้ป่วย

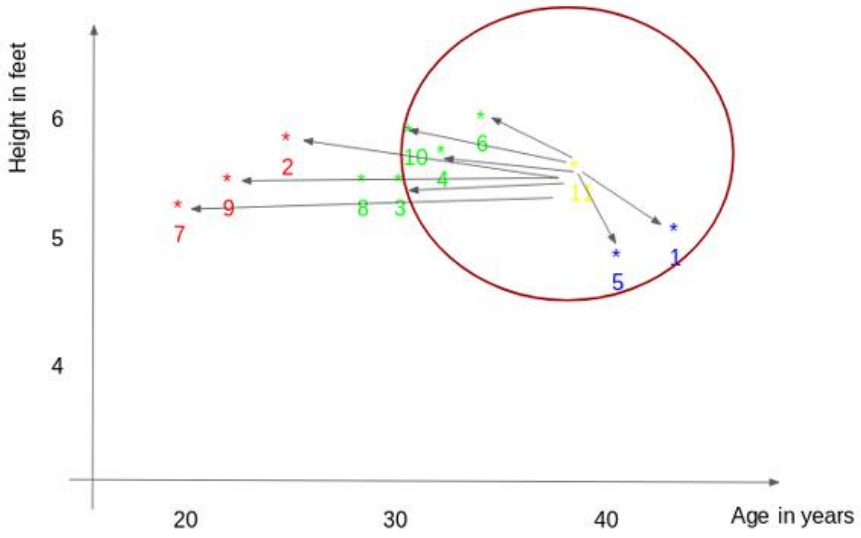
ภาพประกอบ 28 ข้อมูลสถานะการสูบบุหรี่ของผู้ป่วยมีจำนวน 1,544 รายการที่ไม่ทราบสถานะ (Unknown) ทางผู้วิจัยได้ทำการเปลี่ยนเป็นสถานะเป็นไม่สูบบุหรี่ (Never Smoked) เนื่องจากอัตราส่วนของผู้ที่ไม่ทราบสถานะและไม่โรคหัวใจ กับ ผู้ที่ไม่สูบบุหรี่และไม่เป็นโรคหัวใจมีผลลัพธ์ที่ใกล้เคียงกันมาก



ภาพประกอบ 29 ความสัมพันธ์ของจำนวนสถานะโรคหัวใจกับสถานะการสูบบุหรี่

จากภาพประกอบ 29 ในวิจัยนี้พบว่าผู้ที่สูบบุหรี่มีแนวโน้มจะเป็นโรคหัวใจมากกว่าผู้ไม่สูบบุหรี่ เนื่องจากบุหรี่มีนิโคติน และคาร์บอนมอนอกไซด์ ทำให้หัวใจหัวใจทำงานหนักขึ้น และทำให้เส้นเลือดเลี้ยงหัวใจตีบ เป็นเหตุให้เป็นโรคประจำตัวในระยะยาว (โรงพยาบาลศิริราชปิยมหาราชการุณย์, 2020)

จำนวนของ BMI มีจำนวนข้อมูลว่างอยู่ที่ 201 รายการ ซึ่งจะทำการเพิ่มเติมข้อมูลที่ขาดหายไปด้วยเทคนิค KNN imputation (K-nearest neighbor imputation) ทำให้ข้อมูลถูกแทนที่ด้วยค่าเฉลี่ยจำนวน (n) 100 ข้อมูลที่อยู่ตำแหน่งที่ใกล้ที่สุด



ภาพประกอบ 30 การหาข้อมูลที่อยู่ใกล้ด้วย KNN imputation

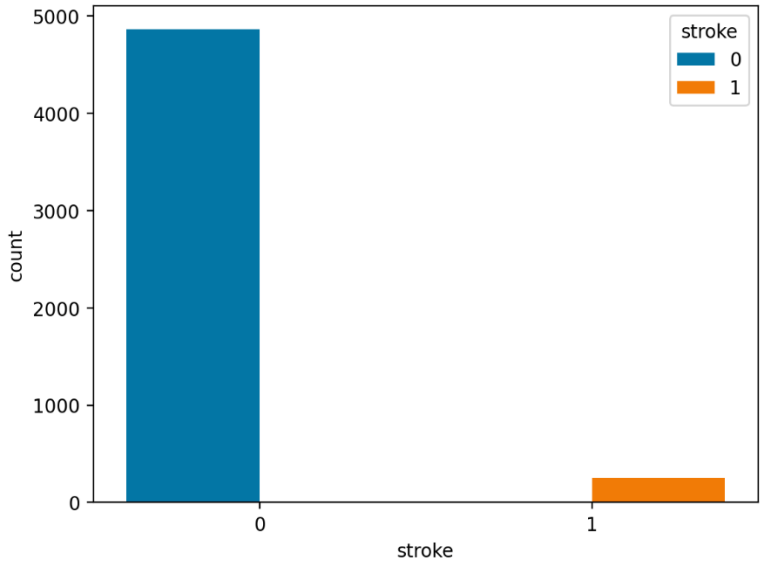
ที่มา: (A. Singh, 2023)

จากภาพประกอบ 30 การหาข้อมูลที่อยู่ใกล้จะใช้วิธีการคำนวณหาระยะทางด้วย Minkowski Distance หาค่าความเหมือนของข้อมูลผลลัพธ์ที่ได้จะมีความสมมาตรมากกว่าการคำนวณหาระยะทางด้วยวิธีอื่น เพราะมีนำจำนวนระยะทางที่วิ่งผ่านพร้อมกันมาใช้ในการคำนวณ (p) (RPG, 2021) ตามสมการที่ 14

$$Minkowski\ Distance = \sqrt[n]{(\sum_i^d |X_{1i} - X_{2i}|^p)} \quad (44)$$

การเตรียมข้อมูลสำหรับการพัฒนาแบบจำลองสำหรับจำแนกผู้ป่วยโรคหลอดเลือดสมอง ข้อมูลที่ใช้ในการสร้างแบบจำลองมีทั้งหมด 5,110 แถว ประกอบด้วย 12 Column ได้แก่ id, gender, age, hypertension, heart_disease, ever_married, work_type, Residence_type, avg_glucose_level, bmi, smoking_status และ stroke โดย id จะไม่ถูกนำมาใช้ในการสร้างแบบจำลอง เนื่องจากไม่มีความเกี่ยวข้องในการสร้างแบบจำลอง

งานวิจัยครั้งนี้เป็นการพัฒนาแบบจำลองทำนายผู้ป่วยโรคหลอดเลือดสมองเป็นปัญหาแบบจำแนกสองกลุ่ม คือ 0 : ผู้ป่วยปกติ และ 1 : ผู้ป่วยโรคหลอดเลือดสมอง



ภาพประกอบ 31 แสดงจำนวนของผู้ป่วยแต่ละกลุ่ม

จากภาพประกอบ 31 ปัญหาของชุดข้อมูลคือเรื่อง ชุดข้อมูลไม่สมดุล มีข้อมูลทั้งหมด 5,110 แถว แบ่งเป็นผู้ป่วยปกติ 4,861 และผู้ป่วยโรคหลอดเลือดสมอง 249 แถว ซึ่งข้อมูลผู้ป่วยปกติมีมากกว่าข้อมูลผู้ป่วยโรคหลอดเลือดสมองอยู่มาก

ก่อนปรับโครงสร้างข้อมูลต้องทำการแยกข้อมูลออกจากชุดข้อมูล เพื่อป้องกันปัญหาข้อมูลรั่วไหล (Data Leakage) คือ การที่มีข้อมูลซ้ำในชุดข้อมูลเรียนรู้ และข้อมูลทดสอบ เกิดจากการค่าผลลัพธ์ของข้อมูลและเพิ่มข้อมูล ดังภาพประกอบ 32

```
X_train.shape, X_test.shape, y_train.shape, y_test.shape
✓ 0.0s
((4088, 10), (1022, 10), (4088,), (1022,))
```

ภาพประกอบ 32 แบ่งชุดข้อมูลเรียนรู้และทดสอบ

ข้อมูลที่เป็นข้อมูลจัดเป็นกลุ่มของข้อมูลชุดนี้ประกอบด้วย hypertension, heart_disease, ever_married, work_type, Residence_type จะถูกแปลงให้เป็นตัวเลขด้วยเทคนิค LabelEncoder ดังภาพประกอบ 33

	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status
4351	1	1	0	0	1	2	0	88.10	29.1	0
12743	0	1	0	0	1	2	0	80.07	38.9	2
5906	0	0	0	0	0	0	1	89.11	23.3	2
13637	0	0	0	0	1	2	0	81.36	36.1	1
13131	1	1	0	0	1	2	0	82.59	29.6	1

ภาพประกอบ 33 ข้อมูลจัดเป็นกลุ่มที่ถูกแปลงด้วยเทคนิค LabelEncoder

หลังจากนั้นข้อมูลเชิงตัวเลขจะถูกนำมา ทำ Scaling เพื่อให้การกระจายตัวของข้อมูลอยู่ในมาตรฐานเดียวกัน โดยใช้เทคนิค StandardScaler ดังภาพประกอบ 34

	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status
845	-0.840344	-0.876887	-0.327962	-0.239061	0.71699	-0.161118	0.987843	-0.819973	0.527012	0.037625
3744	1.187594	-0.876887	-0.327962	-0.239061	-1.39472	-0.161118	-1.012307	0.352075	-0.999346	0.037625
4183	-0.840344	1.140398	-0.327962	-0.239061	0.71699	0.756224	-1.012307	0.090662	-0.507379	0.037625
3409	1.187594	-0.876887	-0.327962	-0.239061	0.71699	-0.161118	0.987843	-0.903944	-0.519994	1.785500
284	1.187594	-0.876887	-0.327962	-0.239061	-1.39472	-1.995803	0.987843	-0.529834	0.337794	0.037625

ภาพประกอบ 34 ข้อมูลเชิงตัวเลขถูกปรับอยู่ในมาตรฐานเดียวกันโดยใช้ StandardScaler

ใช้เทคนิคการแก้ปัญหาไม่สมดุลด้วย การลดจำนวนข้อมูลแบบสุ่ม, การเพิ่มจำนวนข้อมูลแบบสุ่ม และ SMOTE เพื่อให้ข้อมูลมีเพียงพอกับการนำไปใช้ในการเรียนรู้ให้กับแบบจำลอง

```
Original      : (4088, 10)
Undersampling : (398, 10)
Oversampling  : (7778, 10)
SMOTE         : (7778, 10)
```

ภาพประกอบ 35 ปริมาณข้อมูลที่เปลี่ยนไปหลังจากแก้ปัญหาข้อมูลไม่สมดุล

จากภาพประกอบ 35 จำนวนข้อมูลมีการเปลี่ยนแปลงไปจากเดิมจำนวนข้อมูลที่ใช้ในการเรียนรู้มีทั้งหมด 4,088 รายการ 10 คุณลักษณะ ในการแก้ปัญหาลดจำนวนข้อมูลแบบสุ่มทำให้ข้อมูลที่ใช้ในการเรียนรู้ลดลงเหลือ 398 รายการ ต่างจากการใช้การเพิ่มข้อมูลแบบสุ่มและ SMOTE ทำให้ข้อมูลเพิ่มขึ้นเป็น 7,778 รายการ

```
Original      : Counter({0: 3889, 1: 199})
Undersampling : Counter({0: 199, 1: 199})
Oversampling  : Counter({0: 3889, 1: 3889})
SMOTE         : Counter({0: 3889, 1: 3889})
```

ภาพประกอบ 36 ปริมาณข้อมูลที่เปลี่ยนไปหลังจากแก้ปัญหาข้อมูลไม่สมดุล

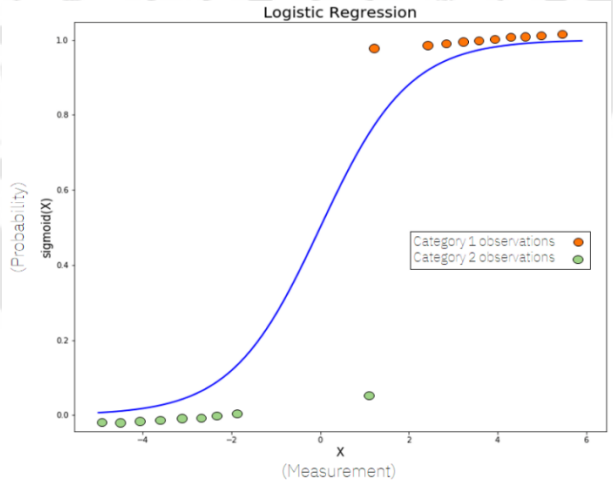
จากภาพประกอบ 36 การลดข้อมูลแบบสุ่มจะทำการลดข้อมูลที่มีจำนวนมากให้มีจำนวนเท่ากับกลุ่มข้อมูลจำนวนน้อย ทำให้มีการคำนวณที่เร็วกว่า แต่อาจทำให้บางกรณีจะสูญเสียข้อมูลที่สำคัญไปบ้าง ต่างจากการเพิ่มข้อมูลและ SMOTE ที่เป็นการเพิ่มข้อมูลกลุ่มจำนวนน้อยให้มีจำนวนเท่ากับกลุ่มข้อมูลจำนวนมาก ถึงแม้ทั้ง 2 วิธีนี้จะเป็นการเพิ่มข้อมูลเหมือนกัน แต่การได้มาซึ่งข้อมูลมีหลักการคำนวณที่แตกต่างกัน

3.5 อัลกอริทึมของแบบจำลองเพื่อจำแนก

ในการสร้างแบบจำลองงานวิจัยฉบับนี้จะเลือกใช้อัลกอริทึมพื้นฐาน (Basic Model) ได้แก่ Logistic Regression และ Decision Tree และอัลกอริทึมซับซ้อน (Complex Model) ได้แก่ Random Forest, XGBoost, Adaboost, LightGBM และ Catboost โดยอัลกอริทึม มีโครงสร้างดังต่อไปนี้

3.5.1 Logistic Regression

เทคนิคสถิติที่ใช้ในการพยากรณ์ความน่าจะเป็นที่จะเกิดเหตุการณ์หรือไม่เกิดเหตุการณ์ของเหตุการณ์หนึ่ง ๆที่สนใจ จะใช้เมื่อตัวแปรตาม (Dependent Variable) เป็นตัวแปรเชิงคุณภาพ ส่วนตัวแปรอิสระ (Independent Variable) หรือตัวแปรพยากรณ์ (Predictor) เป็นตัวแปรที่สนใจหรือเป็นปัจจัยที่สามารถพยากรณ์การเกิดเหตุการณ์หนึ่งที่สนใจ หรือเป็นปัจจัยที่มีอิทธิพลกับการเกิดเหตุการณ์หนึ่งที่สนใจได้ ซึ่งเป็นได้ทั้งตัวแปรเชิงปริมาณและตัวแปรเชิง คุณภาพ สามารถมีได้มากกว่า 1 ตัวแปร ผลลัพธ์ที่ได้จากการใช้อัลกอริทึมอยู่ที่ 0-1 กรณี Binary Classification ดังภาพประกอบที่ 37

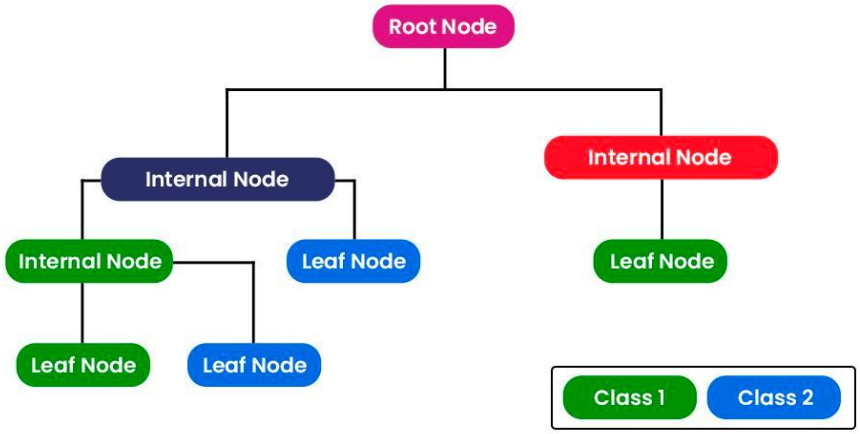


ภาพประกอบ 37 ผลลัพธ์ที่ได้จากแบบจำลองที่สร้างด้วย Logistic Regression

ที่มา: (Thorn, 2020)

3.5.2 Decision Tree

ต้นไม้ที่ใช้ในการสนับสนุนการตัดสินใจ ซึ่งมีลักษณะเป็นโครงสร้างต้นไม้กลับหัวที่มีรากอยู่ด้านบนและใบอยู่ด้านล่าง โดยที่ภายในต้นไม้จะประกอบไปด้วยโหนด ซึ่งแต่ละโหนดนั้น จะแสดงถึงการตัดสินใจบนข้อมูลของคุณสมบัติต่าง ๆ กิ่งของต้นไม้แสดงถึงค่าหรือผลลัพธ์ที่ได้จากการทดสอบ และใบซึ่งเป็นสิ่งที่อยู่ล่างสุดของต้นไม้ตัดสินใจจะแสดงถึงกลุ่มของข้อมูล



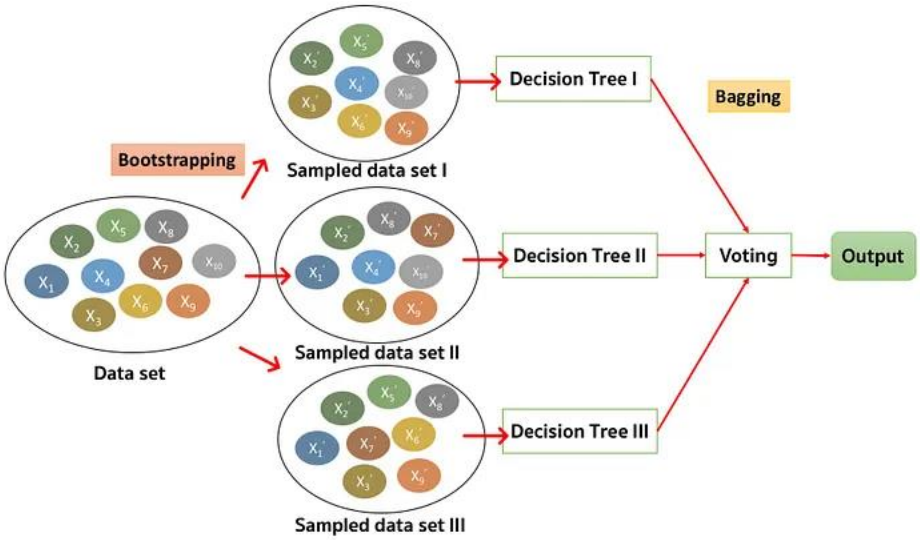
ภาพประกอบ 38 ตัวอย่างโครงสร้างของอัลกอริทึม Decision Tree

ที่มา: (tirumalachandraveni, 2022)

จากภาพประกอบ 38 ข้อมูลที่เป็นรากจะอยู่บนสุด ซึ่งโหนดจะถูกพิจารณาตามคุณสมบัติที่ดีที่สุด โดยใช้ Gini Index และจะถูกแบ่งไปจนกว่า Gini Index มีค่าเท่ากับ 0 เรียกว่า Leaf Node

3.5.3 Random Forest

อัลกอริทึมที่ต่อยอดมาจาก Decision Tree และเป็น Ensemble model ใช้เทคนิค Bootstrapping มีหลักการทำงานคือการสร้างแบบจำลอง Decision Tree หลาย ๆตัว ที่มีคุณลักษณะและข้อมูลที่ไม่เหมือนกัน ทำให้แบบจำลองมีความหลากหลาย และได้ผลลัพธ์ที่ดีกว่า Decision Tree แบบปกติ



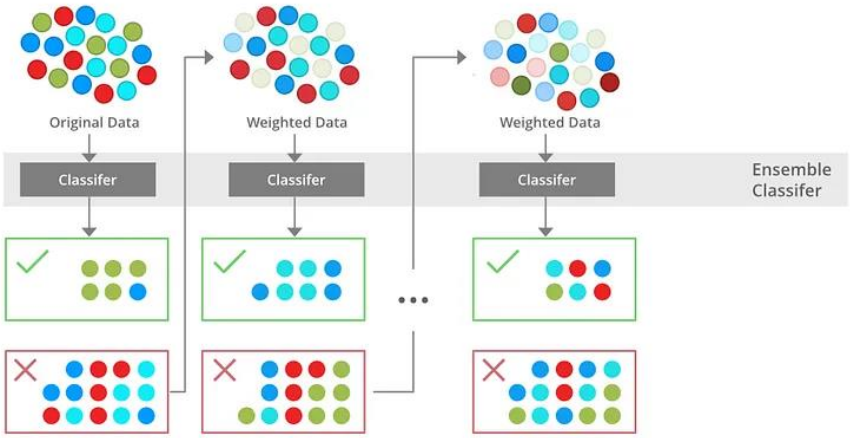
ภาพประกอบ 39 หลักการทำงาน Random forest

ที่มา: (tirumalachandraveni, 2022)

จากภาพประกอบที่ 39 อัลกอริทึม Random Forest จะทำการแบ่งข้อมูลจากชุดข้อมูลนำไปสร้างเป็น Decision Tree หลาย ๆแบบจำลอง ผลลัพธ์จากการทำนายที่ได้จาก Decision Tree จะนำมาทำการโหวตกลุ่มข้อมูลที่ได้จำนวนมากที่สุดจะเป็นผลลัพธ์ของ Random Forest

3.5.4 XGBoost

เป็น Ensemble model ใช้เทคนิค Boosting มีหลักการทำงานคือ สร้างแบบจำลอง Decision Tree ทำการจำแนกและทำซ้ำแบบเรียงลำดับ ข้อมูลที่จำแนกผิดจะถูกนำไปเรียนรู้ต่อในแบบจำลองถัดไป ส่งผลให้มีค่าความผิดพลาดลดลงเมื่อผ่านไปในแต่ละแบบจำลอง ทำให้มีความแม่นยำของแบบจำลองสูง



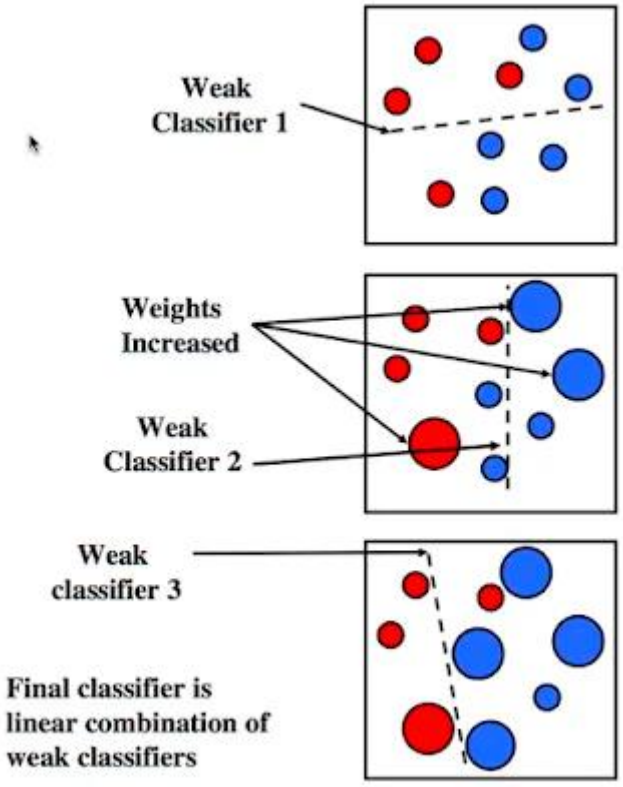
ภาพประกอบ 40 หลักการทำงานของ XGBoost

ที่มา: https://www.geeksforgeeks.org/xgboost/

จากภาพประกอบ 40 แบบจำลองแรกจะทำการทำนายผลลัพธ์ในครั้งแรก หลังจากนั้น แบบจำลองที่สองจะนำผลที่ทำนายผิดของแบบจำลองแรกมาทำการเรียนรู้ และจะทำซ้ำแบบเรียงลำดับจนกว่าค่าการทำนายผิดเหลือน้อยที่สุด

3.5.5 Adaboost

เทคนิคในกลุ่ม Boosting หลักการคือ การรวมแบบจำลองที่มีประสิทธิภาพต่ำ (weak learner) มารวมกันเกิดเป็นแบบจำลองที่มีประสิทธิภาพสูง (Strong learner) มีแนวคิดคือการปรับน้ำหนักให้กับข้อมูลที่แบบจำลองทำนายผิด ข้อมูลที่ทำนายผิดจะถูกปรับน้ำหนักให้สูงขึ้น และข้อมูลที่ทำนายถูกจะปรับให้มีน้ำหนักน้อยลง เพื่อให้ข้อมูลที่จำแนกผิดมีโอกาสได้ถูกนำไปเรียนรู้ในครั้งถัดไป



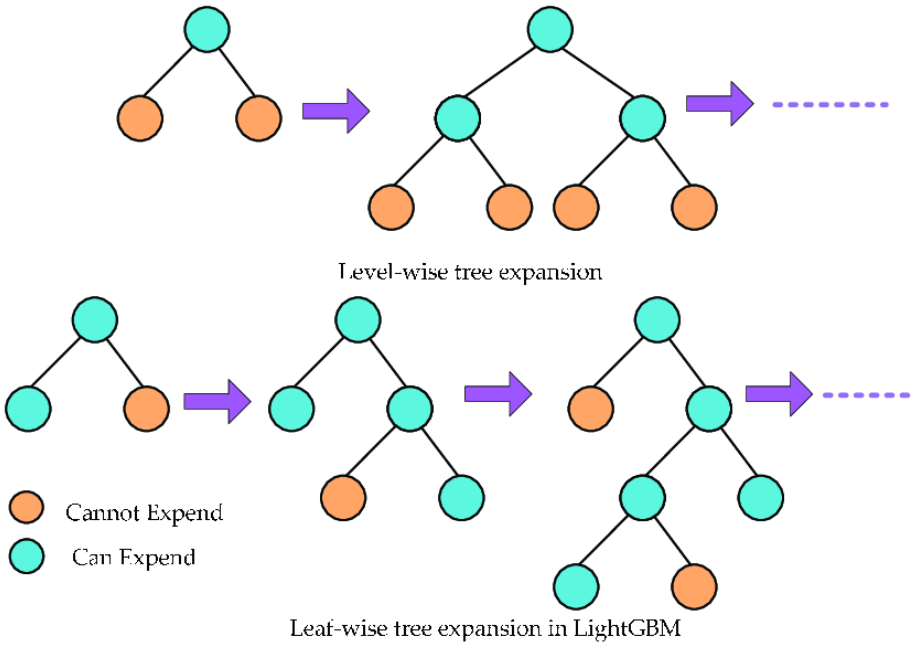
ภาพประกอบ 41 หลักการทำงานของ AdaBoost

ที่มา: https://sirawichjaichuen.medium.com/adaboost-algorithm-cfe6b58e60fa

ภาพประกอบที่ 41 การจำแนกของแบบจำลองแรกข้อมูลที่ทำนายผิด (สีน้ำเงิน) จะถูกนำไปปรับน้ำหนักในการจำแนกในแบบจำลองที่สอง แล้วทำการเรียนรู้ใหม่ ข้อมูลสีน้ำเงินจะถูกเรียนรู้มากกว่าสีแดงเป็นพิเศษ ผลลัพธ์สุดท้ายจะได้แบบจำลองที่ได้ประสิทธิภาพที่ดีที่สุด (investment, 2018)

3.5.6 LightGBM

เป็น Tree-based model เช่นเดียวกับ XGBoost ใช้เทคนิค Boosting มีหลักการสร้างแบบจำลองคือการนำ Decision Tree ที่มีประสิทธิภาพต่ำมาต่อกัน และDecision Tree ถัดไปจะทำการแก้ไขข้อผิดพลาดของ Decision Tree ก่อนหน้า



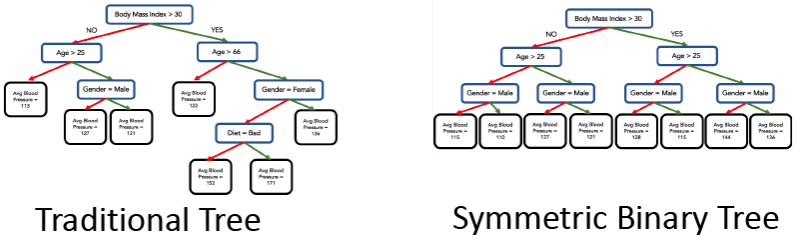
ภาพประกอบ 42 เปรียบเทียบหลักการทำงานของ LightGBM กับ อัลกอริทึมอื่น

ที่มา: https://www.researchgate.net/figure/Tree-expansion-in-LightGBM-Suppose-a-dataset-with-1-2-n-x-x-x-and-1-2-n-y_fig2_358974017

ภาพประกอบ 42 การประมวลผลของอัลกอริทึม LightGBM จะเร็วกว่าอัลกอริทึมอื่น เนื่องจากใช้เทคนิค leaf-wise growth จะทำการขยายกิ่งเฉพาะที่มีค่าความสูญเสียที่น้อยที่สุดเท่านั้น ทำให้สามารถประมวลผลได้เร็วกว่าอัลกอริทึมที่ใช้เทคนิค level-wise growth ที่ต้องประมวลผลทุกกิ่งในแต่ละระดับความลึกของแบบจำลอง (Dong et al., 2022)

3.5.7 CatBoost

เป็น Ensemble model ใช้เทคนิค Boosting มีหลักการทำงานคือการสร้างแบบจำลอง Decision tree ให้อยู่ในรูปแบบของ Symmetric tree หมายถึง จำนวนรากที่แตกโหนด จะมีเงื่อนไขเดียวกันหมด



ภาพประกอบ 43 เปรียบเทียบโครงสร้างของ Traditional Tree กับ Symmetric Tree

ที่มา: https://thaddeus-segura.com/catboost/

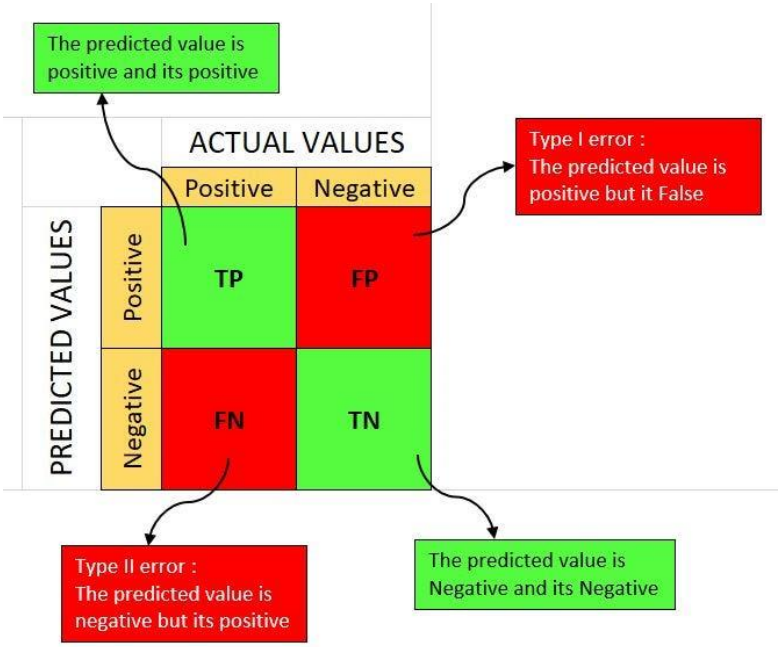
จากภาพประกอบ 43 การที่โหนดมีเงื่อนไขเดียวกันทั้งหมด ทำให้ไม่ต้องประมวลผลแต่ละเงื่อนไขใหม่ ทำให้การทำงานของ CatBoost รวดเร็วและแม่นยำ อีกทั้งมีการใช้หลักการตัดกิ่งไม้ (pruning) ทำให้มีจำนวนรากไม่ลึกส่งผลให้ช่วยลดการเกิด overfitting ของแบบจำลอง (Dong et al., 2022)

3.6 การประเมินประสิทธิภาพของแบบจำลอง

การใช้ตัวประเมินผลในงานวิจัยนี้จะใช้ค่าความแม่นยำสมดุล เนื่องจากชุดข้อมูลมีปัญหาข้อมูลไม่สมดุล ทำให้แบบจำลองให้ความสำคัญกับการจำแนกข้อมูลจำนวนที่มากกว่า ส่งผลให้ตัวประเมินประสิทธิภาพบางตัวมีค่าที่สูงเกินจริง ดังนั้นการใช้ความแม่นยำสมดุลจะช่วยให้การประเมินประสิทธิภาพให้ความสำคัญกับข้อมูลจำนวนน้อยด้วย

3.6.1 Confusion Matrix

Confusion Matrix เป็นตัววัดประสิทธิภาพในการแก้ปัญหา Classification โดยเป็นการแสดงความถี่เปรียบเทียบข้อมูลที่แบบจำลองได้จำแนก กับสิ่งที่เกิดขึ้นจริง



ภาพประกอบ 44 แสดงความถี่ผลลัพธ์ของแบบจำลอง

ที่มา: https://medium.com/analytics-vidhya/what-is-a-confusion-matrix-d1c0f8feda5

จากภาพประกอบ 44 รายละเอียดของตารางแสดงความถี่ในงานวิจัยนี้ มีความหมายดังนี้

True Positive (TP) คือ ผลตรวจพบว่า เป็นผู้ป่วยโรคหลอดเลือดสมอง และ แบบจำลองจำแนกว่าเป็นผู้ป่วยโรคหลอดเลือดสมอง

True Negative (TN) คือ ผลตรวจพบว่า เป็นผู้ป่วยปกติ และ แบบจำลองจำแนกว่าผู้ป่วยปกติ

False Positive (FP) คือ ผลตรวจพบว่า ผู้ป่วยปกติ และ แบบจำลองจำแนกว่า เป็นผู้ป่วยโรคหลอดเลือดสมอง

False Negative (FN) คือ ผลตรวจพบว่า เป็นผู้ป่วยโรคหลอดเลือดสมอง และแบบจำลองจำแนกว่าเป็นผู้ป่วยปกติ

จากการศึกษาทฤษฎีที่เกี่ยวข้องในบทที่ 2 เรื่องการประเมินประสิทธิภาพของแบบจำลอง ผู้วิจัยนำมาปรับใช้สำหรับการประเมินประสิทธิภาพแบบจำลองในงานวิจัย โดยใช้ความถี่ที่ได้จาก Confusion Matrix มาใช้ในการคำนวณ รายละเอียดดังตารางที่ 5

ตาราง 5 แสดงตัววัดประสิทธิภาพของแบบจำลองพร้อมคำอธิบาย

ตัววัดประสิทธิภาพ	คำอธิบาย
ค่าความแม่นยำ (Accuracy)	ใช้วัดค่าความแม่นยำของแบบจำลองที่สามารถจำแนกข้อมูลที่ถูกต้องได้
ค่าความอ่อนไหว (Recall)	เป็นการวัดความสามารถของแบบจำลองในการตรวจจับกลุ่มข้อมูลที่เราสนใจ
ค่าความถูกต้อง (Precision)	เป็นการวัดความสามารถของแบบจำลองในการจำแนกกลุ่มข้อมูลที่เราสนใจ
ค่าความจำเพาะ (Specificity)	ใช้วัดประสิทธิภาพของแบบจำลองในการทำนายกลุ่มข้อมูลที่เราไม่สนใจ
ค่าความแม่นยำสมดุล (Balanced Accuracy)	ใช้วัดประสิทธิภาพของแบบจำลองในการทำนายกลุ่มข้อมูลที่เราสนใจ และไม่สนใจ

3.6.2 **ค่าความแม่นยำสมดุล**

ใช้วัดประสิทธิภาพของแบบจำลองในการทำนายกลุ่มข้อมูลที่เราสนใจ และไม่สนใจ ดังสมการที่ 15

$$Balanced\ Accuracy\ = \frac{sensitivity + specficity}{2} \qquad (15)$$

ในงานวิจัยนี้ใช้ความแม่นยำสมดุลในการวัดประสิทธิภาพของแบบจำลอง เนื่องจาตัววัดนี้มีการนำค่าความจำเพาะที่เป็นตัววัดการทำนายกลุ่มข้อมูลที่เราไม่สนใจ และค่าความอ่อนไหวที่เป็นตัววัดการทำนายกลุ่มที่เราสนใจมาใช้ในการคำนวณร่วมกัน ทำให้ความแม่นยำสมดุลสามารถสะท้อนความสามารถในการทำนายของแบบจำลองได้ทั้งกลุ่มข้อมูลที่เราสนใจ และกลุ่มข้อมูลที่เราไม่สนใจ

ความแตกต่างของความแม่นยำสมดุล และค่าความอ่อนไหว คือ ค่าความอ่อนไหวจะคำนวณเฉพาะกลุ่มข้อมูลที่เราสนใจเท่านั้น ต่างจากความแม่นยำสมดุลที่มีการคำนวณการจำแนกกลุ่มข้อมูลที่เราสนใจ และกลุ่มข้อมูลที่เราไม่ได้สนใจร่วมกัน

ความแตกต่างของความแม่นยำสมดุล และค่า F1 score คือ F1 score คำนวณจากค่าความแม่นยำ และค่าความอ่อนไหว ที่ให้ความสำคัญกับการทำนายชุดข้อมูลที่เราสนใจ แต่ความแม่นยำสมดุลให้น้ำหนักกับการคำนวณความแม่นยำของทั้งสองกลุ่มข้อมูลเท่า ๆกัน

การเลือกใช้ตัววัดประสิทธิภาพของแบบจำลองอาจพิจารณาจากลักษณะของปัญหาและความสำคัญของงานที่ทำ หากชุดข้อมูลมีความสมดุลอาจพิจารณาการใช้ Recall และ F1 score ในการประเมินประสิทธิภาพของแบบจำลอง ซึ่งจะสะท้อนความสามารถของแบบจำลองในการพิจารณากลุ่มข้อมูลที่เราสนใจได้ดีกว่า แต่หากชุดข้อมูลมีความไม่สมดุลอาจพิจาณาการเลือกใช้ความแม่นยำสมดุล เป็นตัววัดประสิทธิภาพของแบบจำลอง เพราะมีการนำผลจากการจำแนกกลุ่มข้อมูลที่เราสนใจ และกลุ่มข้อมูลที่เราไม่สนใจ มาใช้ในการวัดประสิทธิภาพร่วมกัน

3.7 **สรุปแนวคิดและวิธีวิจัย**

งานวิจัยนี้มีเป้าหมายในการศึกษาสร้างแบบจำลองทำนายผู้ป่วยโรคหลอดเลือดสมองของผู้ป่วย โดยใช้ชุดข้อมูล Stroke Prediction Dataset ที่มีจำนวนข้อมูล 5,110 รายการ และ 11 คุณลักษณะมาใช้ในการสร้างแบบจำลอง การสร้างแบบจำลองมีขั้นตอนดังนี้

ทำการสำรวจชุดข้อมูลในการทำความเข้าใจข้อมูลและตรวจสอบความพร้อมของข้อมูลก่อนนำไปใช้ในการสร้างแบบจำลอง ข้อมูลที่ขาดหายจะถูกเติมด้วยค่าเฉลี่ยของกลุ่มข้อมูลที่อยู่ใกล้เคียงกันด้วยเทคนิค KNN Imputation เพื่อลดข้อจำกัดในกรณีที่ข้อมูลมีค่าผิดปกติ (Outlier)

ปรับให้คุณลักษณะให้เหมาะสมกับการนำแบบจำลองด้วย LabelEncoder ในการปรับข้อมูลจัดเป็นกลุ่มให้เป็นข้อมูลเชิงตัวเลข แล้วทำการ Scaling ข้อมูลเชิงตัวเลข เพื่อลดการกระจายตัวของข้อมูล

การจัดการข้อมูลไม่สมดุลก่อนนำข้อมูลไปใช้สร้างแบบจำลองจะใช้ 3 วิธี คือ การลดข้อมูลแบบสุ่ม การเพิ่มข้อมูลแบบสุ่ม และSMOTE แล้วนำข้อมูลไปใช้สร้างแบบจำลองจำแนกผู้ป่วยโรคหลอดเลือดสมอง

อัลกอริทึมที่ใช้ในการสร้างแบบจำลองใช้ 7 อัลกอริทึมประกอบด้วย Logistic Regression, Decision Tree, Random Forest, XGBoost, Adaboost, LightGBM และ Catboost ทุกอัลกอริทึมที่นำมาใช้สร้างแบบจำลองจะถูกปรับ Hyperparameter ทำให้มีประสิทธิภาพในการจำแนกที่ดีขึ้นด้วย RandomGridsearch

เปรียบเทียบประสิทธิภาพด้วยตัวประเมินประสิทธิภาพของแบบจำลอง ในงานวิจัยนี้มุ่งเน้นที่ความแม่นยำสมดุล เนื่องจากชุดข้อมูลไม่สมดุลเมื่อนำไปใช้กับความแม่นยำจะให้ค่าที่สูง

เกินจริง เหตุเกิดจากการที่แบบจำลองให้ความสำคัญในการจำแนกข้อมูลจำนวนมากที่เป็นกลุ่มข้อมูลที่เราไม่ได้สนใจ ระหว่างการสร้างแบบจำลองจะทำการปรับค่า Threshold เพื่อให้แบบจำลองให้ความคัญกับการจำแนกข้อมูลจำนวนน้อยมากขึ้น ซึ่งเป็นข้อมูลกลุ่มเป้าหมายที่เราสนใจ ผลลัพธ์ที่ได้จากการจำแนกจะได้แบบจำลองที่นำไปใช้ในการทำนายผู้ป่วยโรคหลอดเลือดสมองได้

บทที่ 4
ผลการดำเนินการวิจัย

ในการวิจัยศึกษาการสร้างแบบจำลองเพื่อจำแนกผู้ป่วยโรคหลอดเลือดสมอง ซึ่งใช้ข้อมูลคุณลักษณะผู้ป่วยในการเรียนรู้ให้แบบจำลอง โดยใช้เทคนิคการเรียนรู้ของเครื่อง ผู้วิจัยได้ดำเนินการวิจัยโดยการศึกษาตามขบวนการและขั้นตอนต่าง ๆตลอดจนการประเมินผลของแบบจำลองการจำแนก เพื่อให้สอดคล้องกับสมมติฐานที่ได้ตั้งไว้ดังนี้

1. ตัวประเมินประสิทธิภาพของแบบจำลอง มีปัญหาความไม่เหมาะสมในการคำนวณกับชุดข้อมูลไม่สมดุล สามารถแก้ปัญหาด้วยหน่วยวัดประสิทธิภาพ ความแม่นยำสมดุล (Balanced Accuracy)

2. แบบจำลองที่ซับซ้อนมีโครงสร้างการคำนวณที่ซับซ้อนกว่าแบบจำลองพื้นฐาน ทำให้การจำแนกมีประสิทธิภาพที่สูงกว่า

3. การปรับ Threshold สามารถช่วยแก้ปัญหาของแบบจำลองในการตรวจจับการจำแนกกลุ่มข้อมูลจำนวนมาก (Majority Class) และกลุ่มข้อมูลจำนวนน้อย (Minority Class) อย่างไร

4.1 ตัวประเมินประสิทธิภาพของแบบจำลอง มีปัญหาความไม่เหมาะสมในการคำนวณกับชุดข้อมูลไม่สมดุล สามารถแก้ปัญหาด้วยหน่วยวัดประสิทธิภาพ ความแม่นยำสมดุล (Balanced Accuracy)

จากผลการทดลองของงานวิจัย ได้ทำการใช้อัลกอริทึมทั้งหมด 7 อัลกอริทึมในการเปรียบเทียบประกอบด้วย Logistic Regression, Decision Tree, Random Forest, XGBoost, AdaBoost, LightGBM และ Catboost และชุดข้อมูลที่นำมาใช้ในการสร้างแบบจำลองเป็นชุดข้อมูลไม่สมดุลจึงได้มีการใช้เทคนิคการจัดการชุดข้อมูลไม่สมดุลประกอบด้วย การเพิ่มข้อมูลแบบสุ่ม, การลดข้อมูลแบบสุ่ม, SMOTE และ Class Re-weight อีกทั้งยังมีการใช้การปรับค่า Threshold ในการทำนายประเภทข้อมูล เพื่อให้แบบจำลองสามารถเรียนรู้ข้อมูลได้ทุกประเภทในแต่ละกลุ่มข้อมูลเรียนรู้ โดยแบบจำลองที่สร้างด้วยอัลกอริทึม Adaboost เป็นแบบจำลองที่ให้ประสิทธิภาพที่สุด ที่ใช้ความแม่นยำสมดุลในการเปรียบเทียบ ดังตารางที่ 6

ตาราง 6 แสดงความถี่และประสิทธิภาพของแบบจำลองที่สร้างด้วยอัลกอริทึม Adaboost ก่อนปรับน้ำหนักด้วย Class Re-weight

Adaboost Algorithm									
Imbalance Type	Threshold	TN	FN	TP	FP	Accuracy	Recall	ROC	Balance Accuracy
Original Data	0.1	581	8	42	391	0.61	0.84	0.70	0.72
	0.2	972	50	0	0	0.95	0.00	0.70	0.50
	0.3	972	50	0	0	0.95	0.00	0.70	0.50
	0.4	972	50	0	0	0.95	0.00	0.70	0.50
	0.5	972	50	0	0	0.95	0.00	0.70	0.50
SMOTE	0.1	0	0	50	972	0.05	1.00	0.60	0.50
	0.2	0	0	50	972	0.05	1.00	0.60	0.50
	0.3	0	0	50	972	0.05	1.00	0.60	0.50
	0.4	0	0	50	972	0.05	1.00	0.60	0.50
	0.5	0	0	50	972	0.05	1.00	0.60	0.50
Undersampling	0.1	581	8	42	391	0.61	0.84	0.71	0.72
	0.2	972	50	0	0	0.95	0.00	0.71	0.50
	0.3	972	50	0	0	0.95	0.00	0.71	0.50
	0.4	972	50	0	0	0.95	0.00	0.71	0.50
	0.5	972	50	0	0	0.95	0.00	0.71	0.50
Oversampling	0.1	0	0	50	972	0.05	1.00	0.60	0.50
	0.2	0	0	50	972	0.05	1.00	0.60	0.50
	0.3	0	0	50	972	0.05	1.00	0.60	0.50
	0.4	0	0	50	972	0.05	1.00	0.60	0.50
	0.5	959	50	0	13	0.94	0.00	0.60	0.49

ตาราง 7 แสดงความถี่และประสิทธิภาพของแบบจำลองที่สร้างด้วยอัลกอริทึม Adaboost หลังปรับน้ำหนักด้วย Class Re-weight

Adaboost Algorithm									
Imbalance Type	Threshold	TN	FN	TP	FP	Accuracy	Recall	ROC	Balance Accuracy
Original data + Class Re-weight	0.1	581	8	42	391	0.61	0.84	0.70	0.72
	0.2	972	50	0	0	0.95	0.00	0.70	0.50
	0.3	972	50	0	0	0.95	0.00	0.70	0.50
	0.4	972	50	0	0	0.95	0.00	0.70	0.50
	0.5	972	50	0	0	0.95	0.00	0.70	0.50
SMOTE + Class Re-weight	0.1	0	0	50	972	0.05	1.00	0.60	0.50
	0.2	0	0	50	972	0.05	1.00	0.60	0.50
	0.3	0	0	50	972	0.05	1.00	0.60	0.50
	0.4	0	0	50	972	0.05	1.00	0.60	0.50
	0.5	0	0	50	972	0.05	1.00	0.60	0.50
Undersampling + Class Re-weight	0.1	581	8	42	391	0.61	0.84	0.71	0.72
	0.2	972	50	0	0	0.95	0.00	0.71	0.50
	0.3	972	50	0	0	0.95	0.00	0.71	0.50
	0.4	972	50	0	0	0.95	0.00	0.71	0.50
	0.5	972	50	0	0	0.95	0.00	0.71	0.50
Oversampling + Class Re-weight	0.1	0	0	50	972	0.05	1.00	0.60	0.50
	0.2	0	0	50	972	0.05	1.00	0.60	0.50
	0.3	0	0	50	972	0.05	1.00	0.60	0.50
	0.4	0	0	50	972	0.05	1.00	0.60	0.50
	0.5	959	50	0	13	0.94	0.00	0.60	0.49

จากตารางที่ 6 และตารางที่ 7 ผลลัพธ์ที่ได้คือ หากเราเลือกใช้ตัววัดประสิทธิภาพไม่เหมาะสม ตัววัดประสิทธิภาพจะไม่ได้สะท้อนถึงความสามารถในการทำนายของแบบจำลอง ดังตารางตัวอย่าง อัลกอริทึม Adaboost กับ ชุดข้อมูลดั้งเดิม (Original Data) ค่าความแม่นยำจะมีค่าที่สูงกว่าความเป็นจริง แบบจำลองมีความอคติต่อกลุ่มข้อมูล ทำให้ไม่สามารถทำนายข้อมูลที่เราสนใจได้ถูกต้องทำให้ในงานวิจัยนี้ทางผู้วิจัยเลือกใช้ค่าความแม่นยำสมดุล เนื่องจากแสดงความสามารถในการทำนายข้อมูลทุกประเภท

เมื่อทำการใช้ความแม่นยำสมดุลเป็นตัววัดประสิทธิภาพของแบบจำลอง อัลกอริทึมที่ให้ประสิทธิภาพสูงที่สุด โดยการใช้ความแม่นยำสมดุลมีทั้งหมด 4 แบบจำลอง ซึ่งเป็นแบบจำลองที่สร้างด้วยอัลกอริทึม Adaboost ทั้งหมดแบบจำลองแรกคือ การใช้อัลกอริทึม Adaboost กับการใช้ชุดข้อมูลปกติ และแบบจำลองที่สองคือ การใช้อัลกอริทึม Adaboost ปรับน้ำหนักของข้อมูลด้วย Class Re-weight และแบบจำลองที่สาม อัลกอริทึม Adaboost ใช้ร่วมกับเทคนิค Undersampling ในการจัดการปัญหาชุดข้อมูลไม่สมดุล และแบบจำลองที่สี่ อัลกอริทึม Adaboost ใช้เทคนิค Undersampling ร่วมกับการใช้ Class Re-weight ที่ Threshold ที่ 0.1 มีประสิทธิภาพที่เท่ากัน สามารถจำแนกผู้ป่วยโรคหลอดเลือดสมองโดยได้ค่าตัววัดประสิทธิภาพความแม่นยำสมดุลที่ 0.72 จากข้อมูล Confusion Matrix แสดงให้เห็นว่าการจำแนกผู้ป่วยโรคหลอดเลือดสมองถูกต้อง 8 คน เทียบกับผู้ป่วยโรคหลอดเลือดสมองทั้งหมด 50 คน และการทำนายผู้ป่วยปกติถูกต้อง 581 เทียบกับจำนวนผู้ป่วยปกติทั้งหมด 972 คน ถึงแม้ว่าค่าความแม่นยำและค่าความถูกต้อง นั้นสูง แต่ไม่สามารถนำมาใช้ในการวัดประสิทธิภาพของแบบจำลองนี้ได้ เนื่องจากชุดข้อมูลฝึกสอนแบบจำลองยังมีปัญหาชุดข้อมูลไม่สมดุลเพราะจำนวนข้อมูลของผู้ป่วยปกติมากกว่าผู้ป่วยโรคหลอดเลือดสมอง ส่วนค่าความอ่อนไหวที่สามารถบอกผลการทำนายผู้ป่วยโรคหลอดเลือดสมองต่อการทำนายผู้ป่วยโรคหลอดเลือดสมองทั้งหมดมีค่าที่ต่ำ ทำให้ไม่สามารถนำไปใช้ในการวัดประสิทธิภาพของแบบจำลองได้ ดังนั้นจึงควรใช้ค่าความแม่นยำสมดุลในการวัดประสิทธิภาพของแบบจำลอง เพื่อให้ค่าวัดประสิทธิภาพของแบบจำลองสามารถสะท้อนความสามารถในการจำแนกได้ทั้งในส่วนของผู้ป่วยปกติ และผู้ป่วยโรคหลอดเลือดสมอง

สาเหตุที่การใช้เทคนิค Class Re-weight แล้วประสิทธิภาพได้ไม่ดีขึ้น เป็นเพราะ Adaboost จัดอยู่ในอัลกอริทึมประเภท Boosting model ที่มีความสามารถในการจัดการชุดข้อมูลไม่สมดุลอยู่แล้ว โดยมีหลักการทำงาน คือ การสร้างแบบจำลองที่มีประสิทธิภาพสูงจากแบบจำลองที่มีประสิทธิภาพ ด้วยการนำแบบจำลองประสิทธิภาพต่ำมาทำการเรียนรู้ Adaboost จะทำการปรับน้ำหนักของ Feature และข้อมูลไปจนกว่าจะได้แบบจำลองที่มีประสิทธิภาพสูงสุด

ดังนั้นการใช้ Class Re-weight ร่วมกับ Adaboost ในชุดข้อมูลนี้จะไม่ได้ผลลัพธ์ที่ดีขึ้นเพราะการปรับน้ำหนักของกลุ่มข้อมูลเป็นค่ามาตรฐานของ Adaboost อยู่แล้ว (Seiffert et al., 2008)

4.2 แบบจำลองที่ซับซ้อนมีโครงสร้างการคำนวณที่ซับซ้อนกว่าแบบจำลองพื้นฐาน ทำให้การจำแนกมีประสิทธิภาพที่สูงกว่า

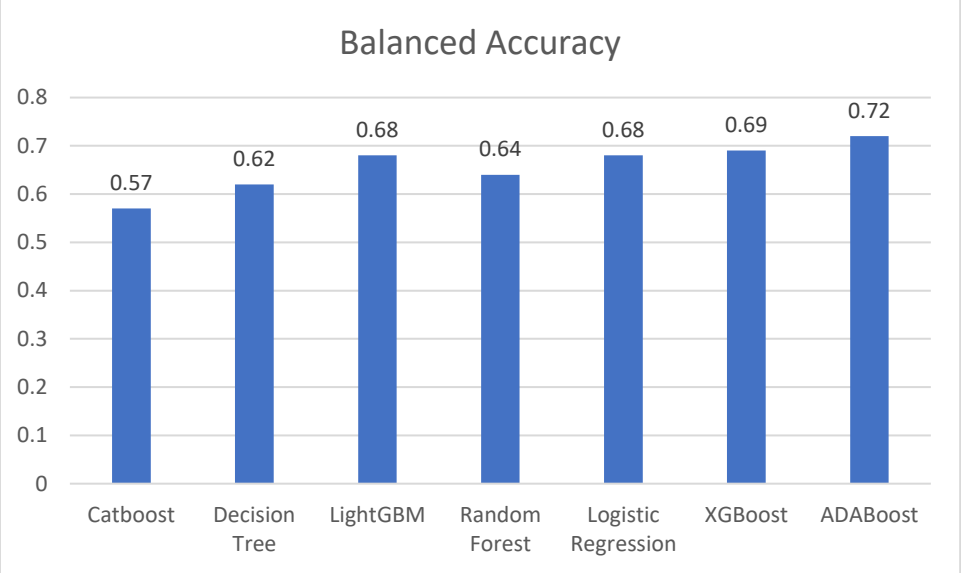
จากผลการศึกษาพบว่าในบางกรณีการใช้แบบจำลองพื้นฐานสามารถให้ความสามารถในการจำแนกได้สูงกว่าแบบจำลองที่ซับซ้อนได้เช่นกัน หากเทคนิคและอัลกอริทึมที่นำมาใช้ไม่เหมาะสมกับข้อมูล (M & Prakash, 2021)

ตาราง 8 เปรียบเทียบความสามารถในการทำนายผลของแบบจำลองที่สร้างด้วยอัลกอริทึม Decision Tree และ Catboost

Type	Threshold	TN	FN	TP	FP	Accuracy	Recall	ROC	Balance Accuracy
Decision Tree + Undersampling + Class Re-weight	0.1	844	33	17	128	0.84	0.34	0.61	0.60
	0.2	868	42	8	104	0.86	0.16	0.61	0.53
	0.3	849	33	17	123	0.85	0.30	0.61	0.59
	0.4	868	37	13	104	0.87	0.34	0.61	0.62
	0.5	865	37	13	107	0.86	0.26	0.61	0.57
Catboosting + SMOTE + Class Re-weight	0.1	788	34	6	184	0.79	0.32	0.64	0.57
	0.2	871	42	8	101	0.86	0.16	0.64	0.53
	0.3	915	42	8	57	0.90	0.16	0.64	0.55
	0.4	940	46	4	32	0.92	0.08	0.64	0.52
	0.5	941	48	2	31	0.92	0.04	0.64	0.50

ตารางที่ 8 แบบจำลองที่สร้างด้วยอัลกอริทึม Decision Tree หากสามารถนำมาใช้ร่วมกับเทคนิคการจัดการข้อมูลไม่สมดุลที่มีความเหมาะสมมากกว่าเมื่อวัดด้วยความแม่นยำสมดุลสามารถให้ประสิทธิภาพที่สูงกว่าอัลกอริทึม Catboost ที่มีอัลกอริทึมที่ซับซ้อนกว่าได้ เนื่องมาจากการเลือกใช้เทคนิคการจัดการข้อมูลที่เหมาะสม ทำให้การทำนายของ Decision Tree มีสัดส่วน

การทำนายที่ถูกต้องของกลุ่มข้อมูลทั้ง 2 ประเภทของแบบจำลอง Decision Tree ทำได้มีประสิทธิภาพที่สูงกว่า



ภาพประกอบ 45 กราฟแสดงประสิทธิภาพของการทำนายแบบจำลองที่สูงสุดของแต่ละอัลกอริทึม

จากภาพประกอบที่ 45 การทำงานของ Catboost มีความซับซ้อน ส่งผลให้ผลลัพธ์ที่ได้อาจไม่ดีกับชุดข้อมูลนี้มีค่าความแม่นยำสมดุลอยู่ที่ 0.57 ซึ่งการเลือกใช้ Decision Tree กับข้อมูลชุดนี้ซึ่งให้ผลลัพธ์ที่ดีกว่ามีค่าอยู่ที่ 0.62 แต่ในบางแบบจำลองที่มีความซับซ้อนก็ให้ประสิทธิภาพที่ดีกว่าแบบจำลองพื้นฐาน เช่น XGBoost และ AdaBoost ที่ให้ประสิทธิภาพของแบบจำลองได้ดีกว่า Decision Tree โดยมีค่าความแม่นยำสมดุลอยู่ที่ 0.69 และ 0.72

การเลือกใช้อัลกอริทึมขึ้นอยู่กับความเหมาะสมของชุดข้อมูล ตัวอย่างเช่น CatBoost ที่เป็นอัลกอริทึมที่ไม่เหมาะกับการใช้กับชุดข้อมูลที่กลุ่มของข้อมูลมีความคล้ายคลึงกัน (Hancock & Khoshgoftaar, 2020) ดังนั้นเมื่อใช้กับชุดข้อมูลผู้ป่วยโรคหลอดเลือดสมองกับผู้ป่วยปกติที่มีคุณลักษณะของข้อมูลคล้ายคลึงกัน ทำให้ได้ประสิทธิภาพที่ต่ำกว่าแบบจำลองพื้นฐานอย่าง Decision Tree ดังนั้นควรใช้อัลกอริทึมและเทคนิคการจัดการข้อมูลที่หลากหลายในการเปรียบเทียบ เพื่อหาแบบจำลองที่ให้ประสิทธิภาพที่ดีที่สุดกับชุดข้อมูลนั้น ๆ

4.3 การปรับ Threshold สามารถช่วยแก้ปัญหาของแบบจำลองในการตรวจจับการจำแนกกลุ่มข้อมูลจำนวนมาก (Majority Class) และกลุ่มข้อมูลจำนวนน้อย (Minority Class) อย่างไร

การปรับ Threshold เป็นการเปลี่ยนผลลัพธ์การทำนายให้อยู่ในรูปแบบของความน่าจะเป็น โดยค่ามาตรฐานในการให้ความน่าจะเป็นในการทำนายจะอยู่ที่ 0.5 หากเป็นการทำนายแบบ 2 กลุ่มของข้อมูลจะอยู่ที่ 50% : 50% ทำให้โอกาสที่จะทำนายข้อมูลทั้งสองกลุ่มมีค่าเท่ากัน และหากปรับค่าให้อยู่ที่ 0.1 โอกาสในการทำนายกลุ่มข้อมูลที่เราสนใจซึ่งเป็นกลุ่มข้อมูลจำนวนน้อยมีความน่าจะเป็นที่จะเป็นในการทำนายกลุ่มข้อมูลนี้หากมีค่าอยู่ระหว่าง 0-10% จะทำให้แบบจำลองทำนายเป็นกลุ่มข้อมูลที่เราสนใจ และหากเกิน 10% จะเป็นกลุ่มข้อมูลที่เราไม่สนใจ การปรับค่า Threshold ทำให้จะช่วยให้แบบจำลองมีความสามารถในการทำนายในการเรียนรู้กลุ่มข้อมูลจำนวนน้อยได้ดีขึ้น และมีโอกาสที่จะสามารถทำนายผู้ป่วยโรคหลอดเลือดสมองได้เพิ่มมากขึ้น ซึ่งเป็นกลุ่มข้อมูลที่เราสนใจที่มีจำนวนไม่มากในชุดข้อมูล

ตาราง 9 ผลการจำแนกแบบจำลองที่สร้างด้วย Logistic Regression ร่วมกับเทคนิค SMOTE

	Logistic Regression								
Imbalanced Type	Threshold	TN	FN	TP	FP	Accuracy	Recall	ROC	Balance Accuracy
SMOTE	0.1	857	26	24	115	0.86	0.48	0.75	0.68
	0.2	944	45	5	28	0.93	0.10	0.75	0.54
	0.3	965	46	4	7	0.95	0.08	0.75	0.54
	0.4	970	50	0	2	0.95	0.00	0.75	0.50
	0.5	972	50	0	0	0.95	0.00	0.75	0.50

ตารางที่ 9 แบบจำลองที่สร้างด้วยอัลกอริทึม Logistic Regression ร่วมกับเทคนิค SMOTE ทดลองโดยการปรับค่า Threshold โดยมีค่ามาตรฐานอยู่ที่ 0.5 จะมีการทำนายกลุ่มผู้ป่วยปกติอยู่ที่ 972 ตัวอย่าง และไม่สามารถทำนายผู้ป่วยโรคหลอดเลือดสมองได้เลย และมีค่าความแม่นยำสมดุลที่ 0.50 เมื่อทำการปรับ Threshold ไปจนถึง 0.1 จะเห็นว่าผลของการจำแนกของแบบจำลองให้ผลลัพธ์ในการทำนายที่ไม่เท่ากัน จุดที่ให้ค่าตัวประสิทธิภาพของแบบจำลองสูงที่สุดอยู่ที่ Threshold 0.1 มีค่าความแม่นยำสมดุลที่อยู่ 0.68 จากผลการทำนายถึงแม้ว่าจะทำนาย

ผลของกลุ่มข้อมูลที่เราไม่สนใจผิดเพิ่มมากขึ้น แต่จะได้ผลของการทำนายกลุ่มข้อมูลที่เราสนใจที่ถูกต้องสูงขึ้น ส่งผลให้การทำนายกลุ่มข้อมูลที่มีจำนวนน้อยได้ดีขึ้น ดังตารางตัวอย่างตารางที่ 9 Threshold ที่ 0.1 ให้ผลลัพธ์ของการทำนายกลุ่มผู้ป่วยปกติ 857 ตัวอย่าง และผู้ป่วยโรคหลอดเลือดสมอง 24 ตัวอย่าง ทำให้มีค่าความแม่นยำสมดุลเพิ่มขึ้นอยู่ที่ 0.68 ซึ่งมากกว่าจุด Threshold 0.50 ที่ให้ความแม่นยำสมดุลอยู่ที่ 0.5 เท่านั้น ซึ่งเป็นผลมาจากการที่จุดที่ Threshold สามารถทำนายกลุ่มข้อมูลจำนวนน้อยถูกต้องได้มากขึ้น

4.4 สรุปผล

งานวิจัยนี้เป็นการศึกษาการสร้างแบบจำลองเพื่อจำแนกโรคหลอดเลือด โดยใช้อัลกอริทึมพื้นฐานและซับซ้อนร่วมกับการใช้เทคนิคจัดการชุดข้อมูลไม่สมดุลเพื่อหาแบบจำลองที่ให้ค่าความแม่นยำสมดุลสูงที่สุด ซึ่งผลลัพธ์ที่ได้ต้องสอดคล้องกับสมมติฐานที่ได้ตั้งไว้ มีรายละเอียดดังนี้

ความเหมาะสมในการเลือกใช้ตัววัดประสิทธิภาพของแบบจำลอง ควรที่จะสะท้อนถึงการทำนายในทุกกลุ่มประเภทข้อมูล หากเลือกใช้ตัวประเมินประสิทธิภาพที่ไม่เหมาะสมจะทำให้ไม่สามารถชี้วัดประสิทธิภาพที่แท้จริงของแบบจำลองได้ ซึ่งในงานวิจัยนี้ได้มีการเลือกใช้ความแม่นยำสมดุลเป็นหลัก ถึงแม้ว่าจะมีค่าที่ใกล้เคียงกับค่า ROC เพราะจุดประสงค์ในการใช้งานและชุดข้อมูลที่ใช้ในงานวิจัยนี้เป็นชุดข้อมูลไม่สมดุล การคำนวณ ROC มาจากค่าความอ่อนไหว (True Positive Rate) และอัตราการทำนายกลุ่มข้อมูลที่เราไม่สนใจ (False Positive Rate) จะบอกค่าการแลกเปลี่ยนระหว่างค่าความอ่อนไหวและอัตราการทำนายกลุ่มที่เราไม่สนใจในทุก ๆ ตำแหน่ง Threshold โดยที่ไม่ได้สนใจความไม่เท่ากันของกลุ่มข้อมูล ต่างจากความแม่นยำสมดุลที่นำผลค่าความถี่ที่ได้จากการทำนายของแบบจำลองมาใช้ในการคำนวณ หากสามารถหาค่า Threshold ที่เหมาะสมในการทำนายทั้งกลุ่มข้อมูลที่เราสนใจ และกลุ่มที่เราไม่สนใจได้ถูกต้องในจำนวนความถี่ที่เหมาะสม การเลือกใช้ความแม่นยำสมดุลจะให้ผลลัพธ์ที่แสดงประสิทธิภาพของแบบจำลองได้เหม (Czakon, 2023)

แบบจำลองที่ซับซ้อนมีความสามารถในการจำแนกได้ดีกว่าแบบจำลองพื้นฐาน เป็นสิ่งที่ไม่เสมอไป ในงานวิจัยนี้ได้มีการศึกษาโดยสร้างแบบจำลองที่มาจากอัลกอริทึมที่มีความซับซ้อนและอัลกอริทึมพื้นฐาน แบบจำลองที่มีความซับซ้อนเช่น Catboost และ LightGBM รวมถึง Random Forest ให้ประสิทธิภาพที่น้อยกว่าแบบจำลองพื้นฐานอย่าง Decision Tree อาจเป็นเพราะชุดข้อมูลนี้ไม่เหมาะกับแบบจำลองที่สร้างด้วยอัลกอริทึมทั้ง 3 แต่แบบจำลองที่สร้างด้วยอัลกอริทึมซับซ้อน เช่น Adaboost และ XGBoost ให้ประสิทธิภาพที่สูงกว่า Decision Tree ดังนั้น

การเลือกใช้อัลกอริทึมที่สร้างแบบจำลองควรเลือกใช้ให้เหมาะสมกับชุดข้อข้อมูล เพราะไม่เสมอไปที่แบบจำลองที่สร้างด้วยอัลกอริทึมที่ซับซ้อนจะให้ผลลัพธ์ที่ดีกว่า

การปรับ Threshold สามารถช่วยให้แบบจำลองสามารถที่จะทำนายกลุ่มข้อมูลจำนวนน้อยได้เพิ่มมากขึ้น เพราะการที่ค่ามาตรฐานอยู่ที่ 0.5 จะทำให้การทำนายของแบบจำลองจะให้ความสำคัญกับกลุ่มข้อมูลจำนวนมาก ซึ่งในบางจุดของ Threshold จะไม่มีการทำนายกลุ่มข้อมูลจำนวนน้อยเลย เพื่อให้แบบจำลองสามารถทำนายกลุ่มข้อมูลจำนวนน้อยได้เพิ่มมากขึ้น จึงควรปรับค่า Threshold ให้น้อยลงเพื่อเพิ่มโอกาสที่จะทำนายกลุ่มข้อมูลจำนวนน้อยเพิ่มมากขึ้น

ดังนั้นการเลือกใช้เทคนิคการสร้างแบบจำลอง หรือการใช้ตัววัดประสิทธิภาพของแบบจำลอง ควรคำนึงถึงลักษณะของชุดข้อมูลที่นำมาใช้ในการสร้างแบบจำลอง หากเลือกใช้ไม่เหมาะสมจะทำให้แบบจำลองที่ได้มีประสิทธิภาพที่ไม่เพียงพอต่อการนำไปใช้งาน

บทที่ 5
สรุปผลการวิจัย อภิปรายผล และข้อเสนอแนะ

ในการวิจัยการสร้างแบบจำลองเพื่อจำแนกผู้ป่วยโรคหลอดเลือดสมอง ซึ่งใช้ข้อมูลคุณลักษณะของผู้ป่วยจากเว็บไซต์ Kaggle โดยการสร้างแบบจำลองใช้เทคนิคการเรียนรู้แบบมีผู้สอน (Supervised Learning) ในแบบ Classification Algorithm เพื่อมุ่งเน้นการแก้ปัญหาความไม่สมดุลของข้อมูลในการจำแนกประเภทแบบไบนารี (Binary Classification) ในส่วนของการประเมินผลแบบจำลองใช้ตัววัดประสิทธิภาพของแบบจำลองในการจำแนกผู้ป่วยปกติและผู้ป่วยโรคหลอดเลือดสมองด้วยค่าความแม่นยำสมดุล (Balanced Accuracy) โดยสามารถแบ่งหัวข้อในการสรุปผลได้ดังต่อไปนี้

1. สรุปผลการวิจัย
2. อภิปรายผลการวิจัย
3. ข้อเสนอแนะ

5.1 สรุปผลการวิจัย

แบบจำลองเพื่อจำแนกผู้ป่วยโรคหลอดเลือดสมองโดยใช้การเรียนรู้ของเครื่องในการสร้างแบบจำลองการจำแนก อย่างไรก็ตามการฝึกฝนแบบจำลองการทำนายมีแน้วโน้มที่จะเกิดอคติกับข้อมูลฝึกฝนแบบจำลองการ เนื่องจากลักษณะที่ไม่สมดุลของข้อมูลในข้อมูลฝึกสอน คือ ผู้ป่วยปกติและผู้ป่วยโรคหลอดเลือดสมอง หากไม่มีการจัดการข้อมูลที่เหมาะสม แบบจำลองการจำแนกจะทำนายกลุ่มข้อมูลจำนวนมากและไม่รู้จักกลุ่มข้อมูลจำนวนน้อย ซึ่งสิ่งที่งานวิจัยนี้ต้องการแก้ปัญหาคือ ความไม่สมดุลของชุดข้อมูล เนื่องจากค่าใช้จ่ายในการเก็บข้อมูลผู้ป่วยโรคหลอดเลือดสมองที่มีค่าใช้จ่ายสูง คือ การทุพพลภาพของร่างกายและการเสียชีวิตของผู้ป่วย

อัลกอริทึมการเรียนรู้ของเครื่องที่ใช้ในการเปรียบเทียบประสิทธิภาพของแบบจำลองประกอบด้วย 7 อัลกอริทึม คือ Logistic Regression, Decision Tree, Random Forest, XGBoost, AdaBoost, LightGBM และ CatBoost และใช้การจัดการกับชุดข้อมูลไม่สมดุล 2 ประเภท คือ ประเภทแรก คือ การสุ่มข้อมูลตัวอย่างใหม่ (Resampling Techniques) ประกอบด้วย การเพิ่มข้อมูลแบบสุ่ม (Random Oversampling), การลดข้อมูลแบบสุ่ม (Random Undersampling) และ SMOTE และประเภทที่สอง คือ การปรับน้ำหนักให้กับประเภทของข้อมูล (Reweighting Techniques) คือ Class Re-weight เพื่อใช้ในการปรับสมดุลของข้อมูลในข้อมูลฝึกสอน โดยรูปแบบของการเปรียบเทียบประสิทธิภาพในการแก้ปัญหาชุดข้อมูลไม่สมดุลจะทำการ

แยกแต่ละเทคนิค รวมถึงการนำทั้งสองวิธีมาใช้ในการแก้ปัญหาชุดข้อมูลไม่สมดุลร่วมกัน แล้วจึงนำมาเปรียบเทียบประสิทธิภาพของแบบจำลอง

จากผลการทดลองโดยการใช้ความแม่นยำสมดุลในการวัดประสิทธิภาพแบบจำลอง สามารถสรุปได้ว่าการสร้างแบบจำลองการทำนายโรคหลอดเลือดสมองด้วยอัลกอริทึม Adaboost ได้ประสิทธิภาพสูงที่สุด โดยมี 4 วิธีที่ให้ประสิทธิภาพที่เท่ากัน คือ การใช้ข้อมูลตั้งต้น (Original Data), การลดข้อมูลแบบสุ่ม, การปรับน้ำหนักให้กับประเภทของข้อมูลร่วมข้อมูลตั้งต้น (Class Re-weight with Original Data) และการปรับน้ำหนักให้กับประเภทของข้อมูลและการลดข้อมูลแบบสุ่ม (Class Re-weight with Undersampling) ในการทดลองครั้งนี้ให้ผลที่เท่ากันคือ ค่าความแม่นยำสมดุลอยู่ที่ 72% ซึ่งสาเหตุที่ให้ผลลัพธ์ที่เท่ากันคือ การที่ Adaboost เป็นอัลกอริทึมที่มีการปรับความสมดุลของข้อมูลอยู่แล้ว ทำให้ถึงแม้ว่าจะใช้เทคนิค Class Re-weight หรือไม่ใช้ ก็จะให้ประสิทธิภาพของแบบจำลองไม่ต่างจากเดิม

5.2 อภิปรายผลการวิจัย

งานวิจัยเรื่องนี้ได้ศึกษาการสร้างแบบจำลองเพื่อการจำแนกผู้ป่วยโรคหลอดเลือดสมองด้วยการแก้ปัญหาชุดข้อมูลไม่สมดุล ซึ่งชุดข้อมูลที่นำมาใช้มาจากเว็บไซต์ Kaggle โดยใช้เทคนิคการเรียนรู้แบบมีผู้สอน ในแบบ Classification Algorithm เพื่อฝึกฝนให้กับแบบจำลอง อย่างไรก็ตามชุดข้อมูลมีแนวโน้มที่จะสร้างความอคติให้กับแบบจำลอง เนื่องจากชุดข้อมูลที่มีความไม่สมดุลของข้อมูลผู้ป่วยปกและผู้ป่วยโรคหลอดเลือดสมอง ที่มีจำนวนของข้อมูลที่มีความแตกต่างกันมาก หากไม่มีการจัดข้อมูลที่ไม่สมดุลอย่างเหมาะสม แบบจำลองจะทำนายกลุ่มข้อมูลที่มีจำนวนมากและไม่รู้จักการกลุ่มข้อมูลจำนวนน้อย ซึ่งก็คือการทำนายของแบบจำลองมีแนวโน้มที่จะเลือกการทำนายกลุ่มข้อมูลผู้ป่วยปกติที่มีจำนวนมากกว่า ในการแก้ปัญหาความไม่สมดุลของข้อมูลการจำแนกประเภทแบบสองประเภทข้อมูล (Binary Classification) ในการสร้างแบบจำลองเพื่อจำแนกผู้ป่วยโรคหลอดเลือดสมอง อัลกอริทึมที่ใช้ในการเรียนรู้มีทั้งหมด 7 อัลกอริทึม คือ Logistic Regression, Decision Tree, Random Forest, XGBoost, AdaBoost, LightGBM และ CatBoost และใช้การจัดการกับชุดข้อมูลไม่สมดุล 2 ประเภท คือ ประเภทแรก คือ การสุ่มข้อมูลตัวอย่างใหม่ (Resampling Techniques) ประกอบด้วย การเพิ่มข้อมูลแบบสุ่ม (Random Oversampling), การลดข้อมูลแบบสุ่ม (Random Undersampling) และ SMOTE และประเภทที่สอง คือ การปรับน้ำหนักให้กับประเภทของข้อมูล (Reweighting Techniques) คือ Class Re-weight ในส่วนของการประเมินผลแบบจำลองใช้รายงาน Classification Report แต่ความไม่

สมดุลของชุดข้อมูลทำให้ตัววัดประสิทธิภาพของแบบจำลองไม่เหมาะสมที่จะนำมาใช้ในการวัดประสิทธิภาพของแบบจำลอง

ในงานวิจัยนี้เลือกใช้ค่าความแม่นยำสมดุล (Balanced Accuracy) แทนการใช้ค่า ROC&AUC ถึงแม้ว่าเป็นตัววัดประสิทธิภาพแบบจำลองของกลุ่มข้อมูลทั้งสองประเภทเหมือนกัน สิ่งที่ต่างกันของสองตัววัดนี้ คือ ROC&AUC จะเป็นการคำนวณค่าเฉลี่ยของผลลัพธ์ในทุก Threshold ที่แบบจำลองทายได้ จึงเหมาะกับการหาค่าที่สูงที่สุดในการทำนาย หรือหาผลลัพธ์ค่าเฉลี่ยของการทำนายของแบบจำลอง แต่ความแม่นยำสมดุล ใช้สำหรับการดู ณ จุด Threshold หนึ่ง ที่แบบจำลองทำนายได้ ซึ่งเหมาะกับการนำไปใช้ในการเปรียบเทียบเพียงบางจุดของ Threshold และเหมาะกับการนำไปใช้งานในการเลือกจุด Threshold ที่ให้ประสิทธิภาพสูงที่สุด

ดังนั้นในงานวิจัยนี้มีการเปรียบเทียบประสิทธิภาพของแบบจำลองเพียงบาง Threshold จึงเลือกใช้ตัววัดประสิทธิภาพความแม่นยำสมดุลเป็นตัววัดประสิทธิภาพหลัก เนื่องจากเหมาะสมกับการตอบคำถามตามสมมติฐานของงานวิจัยที่ตั้งไว้

จากผลการทดลองแบบจำลองการทำนายที่ถูกสร้างด้วยอัลกอริทึม Adaboost ได้ประสิทธิภาพสูงที่สุดด้วยการใช้ความแม่นยำสมดุลในการวัดประสิทธิภาพแบบจำลอง ร่วมกับการแก้ปัญหาชุดข้อมูลไม่สมดุลด้วย Undersampling และการใช้อัลกอริทึม Adaboost ร่วมกับชุดข้อมูลมาตรฐาน เป็นผลมาจากการที่การลดข้อมูลแบบสุ่มจะทำการลดข้อมูลจากกลุ่มข้อมูลจำนวนมาก เพื่อให้มีจำนวนของกลุ่มข้อมูลมีจำนวนที่เท่ากัน แต่การที่ทั้งสองวิธีให้ผลลัพธ์ที่เท่ากันเป็นผลมาจากจำนวนของข้อมูลที่ไม่เยอะและข้อมูลยังมีความคลายคลึงกันมาก ทำให้ไม่ส่งผลต่อประสิทธิภาพของ Adaboost และการใช้เทคนิค Class Re-weight ไม่มีผลต่ออัลกอริทึม Adaboost เพราะอัลกอริทึมมีการทำงานที่ปรับ Weight กับข้อมูลอยู่แล้ว ทำให้ถึงแม้ว่ามีการใช้เทคนิค Class Re-weight ร่วมด้วยก็จะมีประสิทธิภาพเท่าเดิม

5.3 ข้อเสนอแนะ

1. เนื่องจากในงานวิจัยนี้ชุดข้อมูลมาจากเว็บไซต์ Kaggle ที่เป็นชุดข้อมูลเปิดซึ่งอาจจะมาจากการสร้างข้อมูล ทำให้ชุดข้อมูลอาจไม่สะท้อนเหตุการณ์ที่เป็นจริง รวมถึงส่งผลให้การสร้างแบบจำลองก็อาจไม่สามารถนำไปใช้งานจริงได้เช่นกัน และหากนำไปใช้ในการสร้างแบบจำลองเพื่อใช้ในทางการแพทย์ ควรเลือกใช้ตัววัดประสิทธิภาพความอ่อนไหว (Recall)

2. คุณลักษณะของข้อมูลที่เกี่ยวข้องกับมนุษย์ควรคำนึงถึงกายภาพของแต่ละภูมิภาคด้วย เช่น คนเอเชีย คนแอฟริกา หรือคนยุโรป ที่มีค่ามาตรฐาน หรือตัววัดประสิทธิภาพทางด้านร่างกายที่ต่างกัน หมายความว่าการสร้างแบบจำลองที่ใช้คุณลักษณะทางกายภาพของคนเอเชียเพียงอย่างเดียว หากนำไปใช้กับคนยุโรป อาจทำให้ประสิทธิภาพของแบบจำลองมีค่าที่ต่ำ

3. การ Tuning parameter ของอัลกอริทึมในงานวิจัยนี้ใช้แบบ Random Search เนื่องจากทรัพยากรที่จำกัด ซึ่งการใช้วิธีนี้ในการคำนวณแต่ละครั้งจะให้ผลลัพธ์ที่ไม่เหมือนกัน แต่จะดีกว่าการที่ไม่ Tuning แน่นอน แต่หากผู้ศึกษามีทรัพยากรที่ไม่จำกัดสามารถใช้เทคนิค Grid Search ซึ่งเป็นเทคนิคที่ใช้ในการหา parameter ที่ดีที่สุดกับแบบจำลอง แต่จะใช้เวลาในการคำนวณและทรัพยากรที่เยอะกว่า Random Search อยู่มาก

4. ในอนาคตสามารถนำไปต่อยอดเกี่ยวกับการสร้างแบบจำลองในการทำนายในการจำแนกในเรื่องอื่น ๆได้ โดยการใช้ข้อมูลของสิ่งที่เราจะต้องการทำนายตัวอย่างเช่น การทำนายประเภทผลไม้ โดยการใช้ข้อมูลคุณลักษณะของผลไม้ต่าง ๆ เช่น ขนาดของผลไม้ สี รูปทรง เป็นต้น ซึ่งสามารถสร้างแบบจำลองได้ทั้งประเภทการจำแนกแบบข้อมูลสองประเภท และการจำแนกแบบข้อมูลหลายประเภท (Multiclassification) เป็นต้น รวมถึงการสร้างแบบจำลองเพื่อใช้ในการทำนายโรคหลอดเลือดสมองด้วยข้อมูลจริง แต่อาจต้องมีการใช้คุณลักษณะเพิ่มเติม เช่น เชื้อชาติ อุณหภูมิของร่างกาย หรือคุณลักษณะอื่น ๆ ที่เกี่ยวข้อง เพื่อให้แบบจำลองมีความสามารถมากพอที่จะนำไปใช้งาน

บรรณานุกรม

Abdullahi, F. H. (2023). Reinforcement Learning. https://medium.com/@farukhussainkbt/reinforcement-learning-d28543e01a4e

Brodersen, K. H., Ong, C. S., Stephan, K. E., & Buhmann, J. M. (2010). The Balanced Accuracy and Its Posterior Distribution 2010 20th International Conference on Pattern Recognition,

Czakon, J. (2023). F1 Score vs ROC AUC vs Accuracy vs PR AUC: Which Evaluation Metric Should You Choose? https://neptune.ai/blog/f1-score-accuracy-roc-auc-pr-auc

Dong, S., Khattak, A., Ullah, I., Zhou, J., & Hussain, A. (2022). Predicting and Analyzing Road Traffic Injury Severity Using Boosting-Based Ensemble Learning Models with SHAPley Additive exPlanations (20220302 ed., Vol. 19) https://doi.org/10.3390/ijerph19052925

Dritsas, E., & Trigka, M. (2022). Stroke Risk Prediction with Machine Learning Techniques (20220621 ed., Vol. 22) https://doi.org/10.3390/s22134670

Gatchalee, P. (2019). Confusion Matrix เครื่องมือสำคัญในการประเมินผลลัพธ์ของการทำนายใน Machine learning. https://medium.com/@pagongatchalee/confusion-matrix-เครื่องมือสำคัญในการประเมินผลลัพธ์ของการทำนาย-ในmachine-learning-fba6e3f9508c

Hancock, J. T., & Khoshgoftaar, T. M. (2020). CatBoost for big data: an interdisciplinary review (20201104 ed., Vol. 7) https://doi.org/10.1186/s40537-020-00369-8

investment, c. (2018). Adaptive Boosting Algorithm. https://medium.com/cw-quantlab/adaptive-boosting-algorithm-a761f0a0b264

Jamjumrat, C. (2022). รู้จักกับ Decision Tree มันคือต้นไม้อะไร ทำงานอย่างไร ? https://www.borntodev.com/2022/09/15/รู้จักกับ-decision-tree/

Kanraweekultana, N. (2019). เปรียบเทียบการลดมิติข้อมูล และเทคนิคการแสดงข้อมูล (work shop 3.1 & 3.2). https://www.medium.com/@natratanonkanraweekultana/การลดมิติข้อมูลด้วยเทคนิค-pca-และ-t-sne-work-shop-3-1-3-2-c79ec6df518e

M, A. R., & Prakash, D. T. R. D. (2021). A Simple Approach for Selecting the Best Machine Learning Algorithm. International Journal of Scientific & Engineering Research, 12(9).

Obiedat, M., Al-yousef, A., Khasawneh, A., Hamadneh, N., & Aljammal, A. (2020). Using Fuzzy c-Means for Weighting Different Fuzzy Cognitive Maps. International Journal of Advanced Computer Science and Applications, 11(5). https://doi.org/10.14569/ijacsa.2020.0110569

Pandian, S. (2022). A Comprehensive Guide on Hyperparameter Tuning and its Techniques. https://www.analyticsvidhya.com/blog/2022/02/a-comprehensive-guide-on-hyperparameter-tuning-and-its-techniques/

Pattayapon. (2023). มารู้จัก Catboost อัลกอริทึม ทำไมคนถึงใช้กันอย่างแพร่หลาย ?? https://www.medium.com/@pattayapon1311/มารู้จัก-catboost-อัลกอริทึมยอดนิยมใน-kaggle-938a1dd8d643

PradyaSin. (2019). Random Forest คืออะไร. In.

RPG, B. (2021). Day 04 – Minkowski Distance. https://bigdatarpg.com/2021/01/09/day-04-minkowski-distance/

Sachinsoni. (2023). Model Evaluation Techniques in Machine Learning. https://medium.com/@sachinsoni600517/model-evaluation-techniques-in-machine-learning-47ae9fb0ad33

Seiffert, C., Khoshgoftaar, T. M., Hulse, J. V., & Napolitano, A. (2008). Resampling or Reweighting: A Comparison of Boosting Implementations 2008 20th IEEE International Conference on Tools with Artificial Intelligence,

Shan, X., Chen, Y., & Qiao, Z. (2023). A Study of Stroke Prevalence Prediction Based on Random Forest Algorithm 2023 IEEE 3rd International Conference on Power, Electronics and Computer Applications (ICPECA),

Singh, A. (2023). KNN algorithm: Introduction to K-Nearest Neighbors Algorithm for Regression. https://www.analyticsvidhya.com/blog/2018/08/k-nearest-neighbor-introduction-regression-python/

Singh, K. (2023). How to Improve Class Imbalance using Class Weights in Machine

Learning? https://www.analyticsvidhya.com/blog/2020/10/improve-class-imbalance-class-weights/#h-2-logistic-regression-class-weight-balanced

Srimarong, S. (2020). 4 ประเภทของการแบ่งกลุ่มข้อมูล (Clustering). https://bigdata.go.th/big-data-101/4-types-of-clustering/

Team, M. C. (2019). กฎความสัมพันธ์ (Association Rule) คืออะไร ? https://www.mindphp.com/คู่มือ/73-คืออะไร/6852-what-is-a-association-rule.html

Thorn, J. (2020). Logistic Regression Explained. https://towardsdatascience.com/logistic-regression-explained-9ee73cede081

tirumalachandraveni. (2022). CART (Classification And Regression Tree) in Machine Learning. https://www.geeksforgeeks.org/cart-classification-and-regression-tree-in-machine-learning/

Zou, Q., Xie, S., Lin, Z., Wu, M., & Ju, Y. (2016). Finding the Best Classification Threshold in Imbalanced Classification. Big Data Research, 5, 2-8. https://doi.org/10.1016/j.bdr.2015.12.001

เทียมเก่า, ส. (2023). อุบัติการณ์โรคหลอดเลือดสมองประเทศไทย. Thai Journal of Neurology, 39.

โค้ววิลัยแสง, ณ. (2021). ตัวแบบการเรียนรู้จำแนกประเภทซัพพลายเออร์แบบมีผู้สอนสำหรับปัญหาการประเมินประสิทธิภาพของซัพพลายเออร์ในระบบ SAP ERP. Thai Journal of Operations Research:, 9.

โรงพยาบาลศิริราชปิยมหาราชการุณย์. (2020). บุหรี่ตัวร้าย ทำลายหัวใจ. https://www.siphhospital.com/th/news/article/share/620

ไกยวรรณ์, ย. (2012). หลักการและการใช้การวิเคราะห์การถดถอยโลจิสติคสำหรับการวิจัย. วารสารวิจัยมหาวิทยาลัยเทคโนโลยีราชมงคลศรีวิชัย, 4(1), 1-12.

จิตตนูนท์, ป. (2021). ความรู้โรคหลอดเลือดสมองและพฤติกรรมป้องกันของกลุ่มเสี่ยงโรคหลอดเลือดสมอง : กรณีศึกษาตำบลห้วยนาง จังหวัดตรัง. Songklanagarind Journal of Nursing, 41, 13-25.

ภาคผนวก

ตาราง 10 แสดงผลลัพธ์และค่าวัดประสิทธิภาพแบบจำลองที่สร้างด้วยอัลกอริทึม Logistic Regression ก่อนปรับ Class Re-weight

Logistic Regression									
Imbalanced Type	Threshold	TN	FN	TP	FP	Accuracy	Recall	ROC	Balance Accuracy
Original Data	0.1	854	30	20	118	0.86	0.40	0.74	0.64
	0.2	851	45	5	121	0.84	0.10	0.74	0.49
	0.3	964	48	2	8	0.95	0.04	0.74	0.52
	0.4	970	50	0	2	0.95	0.00	0.74	0.50
	0.5	971	50	0	1	0.95	0.00	0.74	0.50
SMOTE	0.1	857	26	24	115	0.86	0.48	0.75	0.68
	0.2	944	45	5	28	0.93	0.10	0.75	0.54
	0.3	965	46	4	7	0.95	0.08	0.75	0.54
	0.4	970	50	0	2	0.95	0.00	0.75	0.50
	0.5	972	50	0	0	0.95	0.00	0.75	0.50
Undersampling	0.1	858	29	21	114	0.86	0.42	0.73	0.65
	0.2	950	45	5	22	0.93	0.10	0.73	0.54
	0.3	965	48	2	7	0.95	0.04	0.73	0.52
	0.4	970	50	0	2	0.95	0.00	0.73	0.50
	0.5	971	50	0	1	0.95	0.00	0.73	0.50
Oversampling	0.1	854	30	20	118	0.86	0.40	0.74	0.64
	0.2	948	45	5	24	0.93	0.10	0.74	0.54
	0.3	966	50	0	6	0.95	0.00	0.74	0.50
	0.4	969	50	0	3	0.95	0.00	0.74	0.50
	0.5	972	50	0	0	0.95	0.00	0.74	0.50

ตาราง 11 แสดงผลลัพธ์และค่าวัดประสิทธิภาพแบบจำลองที่สร้างด้วยอัลกอริทึม Logistic Regression หลังปรับ Class Re-weight

Logistic Regression									
Imbalanced Type	Threshold	TN	FN	TP	FP	Accuracy	Recall	ROC	Balance Accuracy
Original Data + Class Re-weight	0.1	854	30	20	118	0.86	0.40	0.74	0.64
	0.2	851	45	5	121	0.84	0.10	0.74	0.49
	0.3	964	48	2	8	0.95	0.04	0.74	0.52
	0.4	970	50	0	2	0.95	0.00	0.74	0.50
	0.5	971	50	0	1	0.95	0.00	0.74	0.50
SMOTE + Class Re-weight	0.1	853	29	21	119	0.86	0.42	0.74	0.65
	0.2	952	44	6	20	0.94	0.12	0.74	0.55
	0.3	966	47	3	6	0.95	0.06	0.74	0.53
	0.4	968	50	0	4	0.95	0.00	0.74	0.50
	0.5	971	50	0	1	0.95	0.00	0.74	0.50
Undersampling + Class Re-weight	0.1	858	29	21	114	0.86	0.42	0.73	0.65
	0.2	950	45	5	22	0.93	0.10	0.73	0.54
	0.3	965	48	2	7	0.95	0.04	0.73	0.52
	0.4	970	50	0	2	0.95	0.00	0.73	0.50
	0.5	971	50	0	1	0.95	0.00	0.73	0.50
Oversampling + Class Re-weight	0.1	854	30	20	118	0.86	0.40	0.72	0.64
	0.2	948	45	5	24	0.93	0.10	0.72	0.54
	0.3	966	50	0	6	0.95	0.00	0.72	0.50
	0.4	969	50	0	3	0.95	0.00	0.72	0.50
	0.5	972	50	0	0	0.95	0.00	0.72	0.50

ตาราง 12 แสดงผลลัพธ์และค่าวัดประสิทธิภาพแบบจำลองที่สร้างด้วยอัลกอริทึม Decision Tree ก่อนปรับ Class Re-weight

Decision Tree									
Imbalanced Type	Threshold	TN	FN	TP	FP	Accuracy	Recall	ROC	Balance Accuracy
Original Data	0.1	854	30	12	133	0.83	0.24	0.55	0.55
	0.2	851	45	9	109	0.85	0.18	0.55	0.53
	0.3	964	48	9	53	0.91	0.18	0.55	0.56
	0.4	970	50	3	45	0.91	0.06	0.55	0.51
	0.5	971	50	3	23	0.93	0.06	0.55	0.52
SMOTE	0.1	913	48	2	59	0.90	0.04	0.51	0.49
	0.2	917	47	3	55	0.90	0.06	0.51	0.50
	0.3	926	44	6	46	0.91	0.12	0.51	0.54
	0.4	926	46	4	46	0.91	0.08	0.51	0.52
	0.5	921	46	4	51	0.91	0.08	0.51	0.51
Undersampling	0.1	856	41	9	116	0.85	0.18	0.53	0.53
	0.2	897	43	7	75	0.88	0.14	0.53	0.53
	0.3	903	47	3	69	0.89	0.06	0.53	0.49
	0.4	916	45	5	56	0.90	0.10	0.53	0.52
	0.5	933	44	6	39	0.92	0.12	0.53	0.54
Oversampling	0.1	886	39	11	86	0.88	0.22	0.53	0.57
	0.2	908	44	6	64	0.89	0.12	0.53	0.53
	0.3	900	43	7	72	0.89	0.14	0.53	0.53
	0.4	919	46	4	53	0.90	0.08	0.53	0.51
	0.5	938	47	3	34	0.92	0.06	0.53	0.51

ตาราง 13 แสดงผลลัพธ์และค่าวัดประสิทธิภาพแบบจำลองที่สร้างด้วยอัลกอริทึม Decision Tree หลังปรับ Class Re-weight

Decision Tree									
Imbalanced Type	Threshold	TN	FN	TP	FP	Accuracy	Recall	ROC	Balance Accuracy
Original Data + Class Re-weight	0.1	931	43	7	41	0.92	0.14	0.52	0.55
	0.2	933	46	4	39	0.92	0.08	0.52	0.52
	0.3	932	46	4	40	0.92	0.08	0.52	0.52
	0.4	929	47	3	43	0.9	0.06	0.52	0.5
	0.5	935	45	5	37	0.92	0.1	0.52	0.53
SMOTE + Class Re-weight	0.1	919	46	4	53	0.89	0.08	0.51	0.51
	0.2	917	40	10	55	0.91	0.2	0.51	0.57
	0.3	917	46	4	55	0.9	0.08	0.51	0.51
	0.4	924	49	1	48	0.91	0.02	0.51	0.49
	0.5	916	45	5	56	0.91	0.09	0.51	0.52
Undersampling + Class Re-weight	0.1	844	33	17	128	0.84	0.34	0.61	0.60
	0.2	868	42	8	104	0.86	0.16	0.61	0.53
	0.3	849	33	17	123	0.85	0.30	0.61	0.59
	0.4	868	37	13	104	0.87	0.34	0.61	0.62
	0.5	865	37	13	107	0.86	0.26	0.61	0.57
Oversampling + Class Re-weight	0.1	932	48	2	40	0.91	0.04	0.51	0.5
	0.2	919	46	4	53	0.9	0.08	0.51	0.51
	0.3	934	48	2	38	0.92	0.04	0.51	0.5
	0.4	939	48	2	33	0.92	0.04	0.51	0.50
	0.5	930	46	4	42	0.91	0.08	0.51	0.52

ตาราง 14 แสดงผลลัพธ์และค่าวัดประสิทธิภาพแบบจำลองที่สร้างด้วยอัลกอริทึม Random Forest ก่อนปรับ Class Re-weight

Random Forest									
Imbalanced Type	Threshold	TN	FN	TP	FP	Accuracy	Recall	ROC	Balance Accuracy
Original Data	0.1	813	28	22	159	0.82	0.44	0.71	0.64
	0.2	929	43	7	43	0.92	0.14	0.71	0.55
	0.3	960	48	2	12	0.94	0.04	0.71	0.51
	0.4	966	48	2	6	0.95	0.04	0.71	0.52
	0.5	972	50	0	0	0.95	0.00	0.71	0.50
SMOTE	0.1	820	30	20	152	0.82	0.40	0.67	0.62
	0.2	909	45	5	63	0.89	0.10	0.67	0.52
	0.3	951	46	4	21	0.93	0.08	0.67	0.53
	0.4	962	50	0	10	0.94	0.00	0.67	0.49
	0.5	970	50	0	2	0.95	0.00	0.67	0.50
Undersampling	0.1	816	31	19	156	0.82	0.38	0.72	0.61
	0.2	929	44	6	43	0.91	0.12	0.72	0.54
	0.3	958	49	1	14	0.94	0.02	0.72	0.50
	0.4	968	50	0	4	0.95	0.00	0.72	0.50
	0.5	972	50	0	0	0.95	0.00	0.72	0.50
Oversampling	0.1	815	32	18	157	0.82	0.36	0.70	0.60
	0.2	920	44	6	52	0.91	0.12	0.70	0.53
	0.3	949	46	4	23	0.93	0.08	0.70	0.53
	0.4	962	50	0	10	0.94	0.00	0.70	0.49
	0.5	969	50	0	3	0.95	0.00	0.70	0.50

ตาราง 15 แสดงผลลัพธ์และค่าวัดประสิทธิภาพแบบจำลองที่สร้างด้วยอัลกอริทึม Random Forest หลังปรับ Class Re-weight

Random Forest									
Imbalanced Type	Threshold	TN	FN	TP	FP	Accuracy	Recall	ROC	Balance Accuracy
Original Data + Class Re-weight	0.1	836	34	16	136	0.83	0.32	0.70	0.59
	0.2	918	45	5	54	0.9	0.10	0.70	0.52
	0.3	947	49	1	25	0.93	0.02	0.70	0.50
	0.4	929	50	0	43	0.94	0	0.70	0.49
	0.5	935	50	0	37	0.95	0.00	0.70	0.50
SMOTE + Class Re-weight	0.1	845	35	15	127	0.84	0.30	0.68	0.58
	0.2	922	44	6	50	0.91	0.12	0.68	0.53
	0.3	951	48	2	21	0.93	0.04	0.68	0.51
	0.4	964	50	0	8	0.94	0.00	0.68	0.5
	0.5	971	50	0	1	0.95	0.00	0.68	0.5
Undersampling + Class Re-weight	0.1	682	22	28	290	0.69	0.56	0.71	0.63
	0.2	830	31	19	142	0.83	0.38	0.71	0.62
	0.3	905	43	7	67	0.89	0.14	0.71	0.54
	0.4	964	50	0	8	0.92	0.02	0.71	0.49
	0.5	971	50	0	1	0.94	0.00	0.71	0.49
Oversampling + Class Re-weight	0.1	708	24	26	264	0.72	0.52	0.71	0.62
	0.2	863	34	16	109	0.86	0.32	0.71	0.60
	0.3	923	44	6	49	0.91	0.12	0.71	0.53
	0.4	949	47	3	23	0.93	0.06	0.71	0.52
	0.5	967	50	0	5	0.95	0.00	0.71	0.50

ตาราง 16 แสดงผลลัพธ์และค่าวัดประสิทธิภาพแบบจำลองที่สร้างด้วยอัลกอริทึม XGBoost ก่อนปรับ Class Re-weight

XGBoost									
Imbalanced Type	Threshold	TN	FN	TP	FP	Accuracy	Recall	ROC	Balance Accuracy
Original Data	0.1	860	34	16	112	0.86	0.32	0.68	0.60
	0.2	931	46	4	41	0.91	0.08	0.68	0.52
	0.3	946	47	3	26	0.93	0.06	0.68	0.52
	0.4	959	48	2	13	0.94	0.04	0.68	0.51
	0.5	966	50	0	6	0.95	0.00	0.68	0.50
SMOTE	0.1	918	46	4	54	0.90	0.08	0.60	0.51
	0.2	935	46	4	37	0.92	0.08	0.60	0.52
	0.3	943	50	0	29	0.92	0.00	0.60	0.49
	0.4	955	49	1	17	0.94	0.02	0.60	0.50
	0.5	959	47	3	13	0.94	0.06	0.60	0.52
Undersampling	0.1	647	14	36	325	0.67	0.72	0.74	0.69
	0.2	917	40	10	55	0.91	0.20	0.74	0.57
	0.3	960	46	4	12	0.94	0.08	0.74	0.53
	0.4	972	50	0	0	0.95	0.00	0.74	0.50
	0.5	972	50	0	0	0.95	0.00	0.74	0.50
Oversampling	0.1	921	44	6	51	0.91	0.12	0.64	0.53
	0.2	934	46	4	38	0.92	0.08	0.64	0.52
	0.3	949	47	3	23	0.93	0.06	0.64	0.52
	0.4	952	49	1	20	0.93	0.02	0.64	0.50
	0.5	953	50	0	19	0.93	0.00	0.64	0.49

ตาราง 17 แสดงผลลัพธ์และค่าวัดประสิทธิภาพแบบจำลองที่สร้างด้วยอัลกอริทึม XGBoost ก่อนปรับ Class Re-weight

XGBoost									
Imbalanced Type	Threshold	TN	FN	TP	FP	Accuracy	Recall	ROC	Balance Accuracy
Original Data + Class Re-weight	0.1	911	42	8	61	0.90	0.16	0.63	0.55
	0.2	922	45	5	50	0.91	0.10	0.63	0.52
	0.3	938	50	0	34	0.92	0.00	0.63	0.48
	0.4	941	47	3	31	0.92	0.06	0.63	0.51
	0.5	944	48	2	28	0.93	0.04	0.63	0.51
SMOTE + Class Re-weight	0.1	904	42	8	68	0.89	0.16	0.62	0.55
	0.2	931	44	6	41	0.92	0.12	0.62	0.54
	0.3	939	48	2	33	0.92	0.04	0.62	0.50
	0.4	937	47	3	35	0.92	0.06	0.62	0.51
	0.5	948	47	3	24	0.93	0.06	0.62	0.52
Undersampling + Class Re-weight	0.1	748	24	26	224	0.76	0.52	0.70	0.64
	0.2	834	37	13	138	0.83	0.26	0.70	0.56
	0.3	871	35	15	101	0.87	0.30	0.70	0.60
	0.4	895	43	7	77	0.88	0.14	0.70	0.53
	0.5	922	46	4	50	0.91	0.08	0.70	0.51
Oversampling + Class Re-weight	0.1	912	42	8	60	0.90	0.16	0.62	0.55
	0.2	929	46	4	43	0.91	0.08	0.62	0.52
	0.3	932	48	2	40	0.91	0.04	0.62	0.50
	0.4	941	47	3	31	0.92	0.06	0.62	0.51
	0.5	948	45	5	24	0.93	0.10	0.62	0.54

ตาราง 18 แสดงผลลัพธ์และค่าวัดประสิทธิภาพแบบจำลองที่สร้างด้วย Adaboost ก่อนปรับ Class Re-weight

Adaboost									
Imbalanced Type	Threshold	TN	FN	TP	FP	Accuracy	Recall	ROC	Balance Accuracy
Original Data	0.1	581	8	42	391	0.61	0.84	0.70	0.72
	0.2	972	50	0	0	0.95	0.00	0.70	0.50
	0.3	972	50	0	0	0.95	0.00	0.70	0.50
	0.4	972	50	0	0	0.95	0.00	0.70	0.50
	0.5	972	50	0	0	0.95	0.00	0.70	0.50
SMOTE	0.1	0	0	50	972	0.05	1.00	0.60	0.50
	0.2	0	0	50	972	0.05	1.00	0.60	0.50
	0.3	0	0	50	972	0.05	1.00	0.60	0.50
	0.4	0	0	50	972	0.05	1.00	0.60	0.50
	0.5	0	0	50	972	0.05	1.00	0.60	0.50
Undersampling	0.1	581	8	42	391	0.61	0.84	0.71	0.72
	0.2	972	50	0	0	0.95	0.00	0.71	0.50
	0.3	972	50	0	0	0.95	0.00	0.71	0.50
	0.4	972	50	0	0	0.95	0.00	0.71	0.50
	0.5	972	50	0	0	0.95	0.00	0.71	0.50
Oversampling	0.1	0	0	50	972	0.05	1.00	0.60	0.50
	0.2	0	0	50	972	0.05	1.00	0.60	0.50
	0.3	0	0	50	972	0.05	1.00	0.60	0.50
	0.4	0	0	50	972	0.05	1.00	0.60	0.50
	0.5	959	50	0	13	0.94	0.00	0.60	0.49

ตาราง 19 แสดงผลลัพธ์และค่าวัดประสิทธิภาพแบบจำลองที่สร้างด้วย Adaboost หลังปรับ Class Re-weight

Adaboost									
Imbalanced Type	Threshold	TN	FN	TP	FP	Accuracy	Recall	ROC	Balance Accuracy
Original Data + Class Re-weight	0.1	581	8	42	391	0.61	0.84	0.70	0.72
	0.2	972	50	0	0	0.95	0.00	0.70	0.50
	0.3	972	50	0	0	0.95	0.00	0.70	0.50
	0.4	972	50	0	0	0.95	0.00	0.70	0.50
	0.5	972	50	0	0	0.95	0.00	0.70	0.50
SMOTE + Class Re-weight	0.1	0	0	50	972	0.05	1.00	0.60	0.50
	0.2	0	0	50	972	0.05	1.00	0.60	0.50
	0.3	0	0	50	972	0.05	1.00	0.60	0.50
	0.4	0	0	50	972	0.05	1.00	0.60	0.50
	0.5	0	0	50	972	0.05	1.00	0.60	0.50
Undersampling + Class Re-weight	0.1	581	8	42	391	0.61	0.84	0.71	0.72
	0.2	972	50	0	0	0.95	0.00	0.71	0.50
	0.3	972	50	0	0	0.95	0.00	0.71	0.50
	0.4	972	50	0	0	0.95	0.00	0.71	0.50
	0.5	972	50	0	0	0.95	0.00	0.71	0.50
Oversampling + Class Re-weight	0.1	0	0	50	972	0.05	1.00	0.60	0.50
	0.2	0	0	50	972	0.05	1.00	0.60	0.50
	0.3	0	0	50	972	0.05	1.00	0.60	0.50
	0.4	0	0	50	972	0.05	1.00	0.60	0.50
	0.5	959	50	0	13	0.94	0.00	0.60	0.49

ตาราง 20 แสดงผลลัพธ์และค่าวัดประสิทธิภาพแบบจำลองที่สร้างด้วย LightGBM ก่อนปรับ Class Re-weight

LightGBM									
Imbalanced Type	Threshold	TN	FN	TP	FP	Accuracy	Recall	ROC	Balance Accuracy
Original Data	0.1	865	34	16	107	0.86	0.32	0.74	0.60
	0.2	949	48	2	23	0.93	0.04	0.74	0.51
	0.3	971	50	0	1	0.95	0.00	0.74	0.50
	0.4	972	50	0	0	0.95	0.00	0.74	0.50
	0.5	972	50	0	0	0.95	0.00	0.74	0.50
SMOTE	0.1	935	44	6	37	0.92	0.12	0.65	0.54
	0.2	943	47	3	29	0.93	0.06	0.65	0.52
	0.3	948	50	0	24	0.93	0.00	0.65	0.49
	0.4	955	49	1	17	0.94	0.02	0.65	0.50
	0.5	950	50	0	22	0.93	0.00	0.65	0.49
Undersampling	0.1	935	30	20	37	0.93	0.40	0.73	0.68
	0.2	943	47	3	29	0.93	0.06	0.73	0.52
	0.3	948	49	1	24	0.93	0.02	0.73	0.50
	0.4	955	49	1	17	0.94	0.02	0.73	0.50
	0.5	950	50	0	22	0.93	0.00	0.73	0.49
Oversampling	0.1	937	48	2	35	0.92	0.04	0.53	0.50
	0.2	952	50	0	20	0.93	0.00	0.53	0.49
	0.3	914	48	2	58	0.90	0.04	0.53	0.49
	0.4	944	48	2	28	0.93	0.04	0.53	0.51
	0.5	858	43	7	114	0.85	0.14	0.53	0.51

ตาราง 21 แสดงผลลัพธ์และค่าวัดประสิทธิภาพแบบจำลองที่สร้างด้วย LightGBM หลังปรับ Class Re-weight

LightGBM									
Imbalanced Type	Threshold	TN	FN	TP	FP	Accuracy	Recall	ROC	Balance Accuracy
Original Data + Class Re-weight	0.1	941	49	1	31	0.92	0.02	0.61	0.49
	0.2	943	47	3	29	0.93	0.06	0.61	0.52
	0.3	954	48	2	18	0.94	0.04	0.61	0.51
	0.4	945	48	2	27	0.93	0.04	0.61	0.51
	0.5	952	48	2	20	0.93	0.04	0.61	0.51
SMOTE + Class Re-weight	0.1	816	29	21	156	0.82	0.42	0.65	0.63
	0.2	875	37	13	97	0.87	0.26	0.65	0.58
	0.3	899	44	6	73	0.89	0.12	0.65	0.52
	0.4	917	42	8	55	0.91	0.16	0.65	0.55
	0.5	929	42	8	43	0.92	0.16	0.65	0.56
Undersampling + Class Re-weight	0.1	830	33	17	142	0.83	0.34	0.67	0.60
	0.2	858	38	12	114	0.85	0.24	0.67	0.56
	0.3	903	40	10	69	0.89	0.20	0.67	0.56
	0.4	914	44	6	58	0.90	0.12	0.67	0.53
	0.5	934	45	5	38	0.92	0.10	0.67	0.53
Oversampling + Class Re-weight	0.1	879	39	11	93	0.87	0.22	0.62	0.56
	0.2	909	41	9	63	0.90	0.18	0.62	0.56
	0.3	926	42	8	46	0.91	0.16	0.62	0.56
	0.4	925	45	5	47	0.91	0.10	0.62	0.53
	0.5	935	48	2	37	0.92	0.04	0.62	0.50

ตาราง 22 แสดงผลลัพธ์และค่าวัดประสิทธิภาพแบบจำลองที่สร้างด้วย Catboost ก่อนปรับ Class Re-weight

Catboost									
Imbalanced Type	Threshold	TN	FN	TP	FP	Accuracy	Recall	ROC	Balance Accuracy
Original Data	0.1	822	35	15	150	0.82	0.30	0.72	0.57
	0.2	949	46	4	23	0.93	0.08	0.72	0.53
	0.3	970	50	0	2	0.95	0.00	0.72	0.50
	0.4	968	49	1	4	0.95	0.02	0.72	0.51
	0.5	972	50	0	0	0.95	0.00	0.72	0.50
SMOTE	0.1	933	48	2	39	0.91	0.04	0.68	0.50
	0.2	947	49	1	25	0.93	0.02	0.68	0.50
	0.3	947	47	3	25	0.93	0.06	0.68	0.52
	0.4	961	49	1	11	0.94	0.02	0.68	0.50
	0.5	963	50	0	9	0.94	0.00	0.68	0.50
Undersampling	0.1	0	0	50	972	0.05	1.00	0.74	0.50
	0.2	0	0	50	972	0.05	1.00	0.74	0.50
	0.3	968	49	1	4	0.95	0.02	0.74	0.51
	0.4	972	50	0	0	0.95	0.00	0.74	0.50
	0.5	972	50	0	0	0.95	0.00	0.74	0.50
Oversampling	0.1	887	40	10	85	0.88	0.20	0.68	0.56
	0.2	936	47	3	36	0.92	0.06	0.68	0.51
	0.3	947	49	1	25	0.93	0.02	0.68	0.50
	0.4	957	49	1	15	0.94	0.02	0.68	0.50
	0.5	961	49	1	11	0.94	0.02	0.68	0.50

ตาราง 23 แสดงผลลัพธ์และค่าวัดประสิทธิภาพแบบจำลองที่สร้างด้วย Catboost หลังปรับ Class Re-weight

Catboost									
Imbalanced Type	Threshold	TN	FN	TP	FP	Accuracy	Recall	ROC	Balance Accuracy
Original Data + Class Re-weight	0.1	896	46	4	76	0.88	0.08	0.65	0.50
	0.2	909	43	7	63	0.90	0.14	0.65	0.54
	0.3	928	46	4	44	0.91	0.08	0.65	0.52
	0.4	936	45	5	36	0.92	0.10	0.65	0.53
	0.5	950	49	1	22	0.93	0.02	0.65	0.50
SMOTE + Class Re-weight	0.1	788	34	6	184	0.79	0.32	0.64	0.57
	0.2	871	42	8	101	0.86	0.16	0.64	0.53
	0.3	915	42	8	57	0.90	0.16	0.64	0.55
	0.4	940	46	4	32	0.92	0.08	0.64	0.52
	0.5	941	48	2	31	0.92	0.04	0.64	0.50
Undersampling + Class Re-weight	0.1	848	41	9	124	0.84	0.18	0.61	0.53
	0.2	900	48	2	72	0.88	0.04	0.61	0.48
	0.3	915	48	2	57	0.90	0.04	0.61	0.49
	0.4	931	46	4	41	0.91	0.08	0.61	0.52
	0.5	937	50	0	35	0.92	0.00	0.61	0.48
Oversampling + Class Re-weight	0.1	898	42	8	74	0.89	0.16	0.65	0.54
	0.2	915	44	6	57	0.90	0.12	0.65	0.53
	0.3	922	47	3	50	0.91	0.06	0.65	0.50
	0.4	928	47	3	44	0.91	0.06	0.65	0.51
	0.5	930	49	1	42	0.91	0.02	0.65	0.49

ประวัติผู้เขียน