# การศึกษาอัลกอริทึมระดับข้อมูลสำหรับชุดข้อมูลที่ไม่สมดุล

# THE STUDY OF DATA LEVEL ALGORITHMS FOR IMBALANCED DATASET

**สิริรัตน์ ไชยธงรัตน์**

**ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญา**
**วิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์)**
**ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์**
**สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง**
**ปีการศึกษา 2566**

# THE STUDY OF DATA LEVEL ALGORITHMS FOR IMBALANCED DATASET

SIRIRAT CHAITHONGRAT

SPECIAL PROBLEM SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT
FOR THE DEGREE OF BACHELOR OF SCIENCE (COMPUTER SCIENCE)
DEPARTMENT OF COMPUTER SCIENCE, SCHOOL OF SCIENCE
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2023

| | |
|---|---|
| **หัวข้อปัญหาพิเศษ** | การศึกษาอัลกอริทึมระดับข้อมูลสำหรับชุดข้อมูลที่ไม่สมดุล<br>The Study Of Data Level Algorithms For Imbalanced Dataset |
| **ชื่อนักศึกษา** | นางสาว สิริรัตน์ ไชยธงรัตน์ รหัสนักศึกษา 63050201 |
| **ปริญญา** | วิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์) |
| **ภาควิชา** | วิทยาการคอมพิวเตอร์ |
| **ปีการศึกษา** | 2566 |
| **อาจารย์ที่ปรึกษา** | ผศ.ดร.อนันตพร หรรษคุณาฒัย |

คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง (สจล.) อนุมัติให้โครงงานพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์) ประจำปีการศึกษา 2566

| คณะกรรมการสอบ | ลายมือชื่อ |
|---|---|
| ผศ.ดร.ศรัณย์ อินทโกสุม<br>ประธานกรรมการ | |
| ดร.จักรพันธ์ เตไชยา<br>กรรมการ | |
| ผศ.ดร.อนันตพร หรรษคุณาฒัย<br>กรรมการและอาจารย์ที่ปรึกษา | |

**หัวข้อปัญหาพิเศษ** การศึกษาอัลกอริทึมระดับข้อมูลสำหรับชุดข้อมูลที่ไม่สมดุล
**ชื่อนักศึกษา** นางสาว สิริรัตน์ ไชยธงรัตน์ รหัสนักศึกษา 63050201
**ปริญญา** วิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์)
**ภาควิชา** วิทยาการคอมพิวเตอร์
**คณะ** วิทยาศาสตร์
**มหาวิทยาลัย** สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง (สจล.)
**ปีการศึกษา** 2566
**อาจารย์ที่ปรึกษา** ผศ.ดร.อนันตพร หรรษคุณาฒัย

## บทคัดย่อ

ความไม่สมดุลของข้อมูล เป็นปัญหาอย่างหนึ่งในการเรียนรู้ของเครื่องที่ส่งผลกระทบต่อประสิทธิภาพในการจำแนกประเภทของโมเดล วิธีการหรือเทคนิคการแก้ไขปัญหาของชุดข้อมูลที่ไม่สมดุลอย่าง เทคนิคการสุ่มตัวอย่างที่เป็นวิธีการที่นิยมอย่างมากในการแก้ไขปัญหาความไม่สมดุลของชุดข้อมูล โดยในงานวิจัยนี้ได้ทำการทดลองเพื่อเปรียบเทียบประสิทธิภาพของเทคนิคการสุ่มตัวอย่าง 3 เทคนิค ได้แก่ SMOTE Tomek Links และ RUSBoostClassifier ร่วมกับอัลกอริทึมการเรียนรู้ของเครื่อง ซึ่งได้แก่ Decision Trees, Naïve Bayes, k-Nearest Neighbors, Support Vector Machines และ Artificial Neural Networks จากการทดลองพบว่าโมเดลที่มีการเรียนรู้โดยการใช้หลักการความน่าจะเป็นในการจำแนกข้อมูลอย่าง Decision Trees และ Naïve Bayes จะทำงานได้ดีกับเทคนิค RUSBoostClassifier ส่วนโมเดลที่ใช้การคำนวณเชิงคณิตศาสตร์หรือใช้ระยะทางระหว่างข้อมูลพิจารณาในการจำแนกข้อมูล เช่น Artificial Neural Networks, Support Vector Machines และ k-Nearest Neighbors จะทำงานได้ดีกับเทคนิค SMOTE นอกจากนี้ยังพบว่าจำนวนคุณลักษณะมีผลต่อเทคนิคการสุ่มข้อมูล ซึ่งข้อมูลในการทดลองได้มากจากชุดข้อมูลบนเว็บไซต์ Kaggle จำนวน 10 ชุด

**คำสำคัญ** : ความไม่สมดุลของชุดข้อมูล เทคนิคการสุ่มตัวอย่าง เทคนิค RUSBoostClassifier เทคนิค SMOTE

**Title** The Study Of Date Level Algorithms For Imbalanced Dataset
**Students** Miss Sirirat Chaithongrat Student ID 63050201
**Degree** Bachelor of Science (Computer Science)
**Department** Computer Science
**School** Science
**University** King Mongkut's Institute of Technology Ladkrabang (KMITL)
**Academic Year** 2023
**Advisor** Asst.Prof.Dr. Anantaporn Hanskunatai

## Abstract

Data imbalance is one of the challenges in machine learning that impacts the performance of classification models. Random sampling techniques are widely used to address this issue. This research experimentally compared the effectiveness of three random sampling techniques: SMOTE, Tomek Links, and RUSBoostClassifier, in combination with machine learning algorithms including Decision Trees, Naïve Bayes, Support Vector Machines, k-Nearest Neighbors, and Artificial Neural Networks. The experiment revealed that models learning through probability-based methods such as Decision Trees and Naïve Bayes performed well with the RUSBoostClassifier technique. On the other hand, models that utilize mathematical computations or consider distances between data points, such as Artificial Neural Networks, Support Vector Machines, and k-Nearest Neighbors, worked effectively with the SMOTE technique. Additionally, it was found that the number of features had an impact on the choice of sampling technique. The experimental data were obtained from 10 datasets available on the Kaggle website.

**Keywords** : imbalanced data sampling technique, RUSBoostClassifier, SMOTE

# กิตติกรรมประกาศ

ในการทำวิจัยเล่มนี้ สามารถสำเร็จได้ด้วยดีจากการช่วยเหลือและการสนับสนุนบุคคลหลายท่าน ผู้จัดทำจึงขอขอบพระคุณบุคคลดังต่อไปนี้

ผศ.ดร.อนันตพร หรรษคุณาฒัย อาจารย์ที่ปรึกษาโครงพิเศษที่กรุณาให้คำปรึกษาในขั้นตอนการดำเนินงานวิจัยและตรวจสอบความเรียบร้อยของงานมาตลอดในการทำวิจัยเล่มนี้

ผศ.ดร.ศรัณย์ อินทโกสุม และดร.จักรพันธ์ เตไชยา ประธานกรรมการและกรรมการที่เสียสละเวลาในการชี้แนะแนวทางการพัฒนา ที่ควรจะพิจารณาและแก้ไข

และเพื่อนร่วมภาควิชาทุกคนในสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังที่ให้คำปรึกษา และการช่วยเหลือตลอดมา นอกจากนี้อาจมีบุคคลที่ไม่ได้กล่าวถึง ณ ที่นี้ จึงขอขอบพระคุณทุกท่านที่มีส่วนร่วมในการช่วยเหลือในการทำวิจัยปัญหาพิเศษเล่มนี้

สิริรัตน์ ไชยธงรัตน์

# สารบัญ

**หน้า**

# สารบัญ(ต่อ)

หน้า

# สารบัญ(ต่อ)

**หน้า**

# สารบัญตาราง

**ตารางที่** **หน้า**

# สารบัญรูป

**รูปที่** **หน้า**

# สารบัญรูป(ต่อ)

**รูปที่** **หน้า**

# สารบัญรูป(ต่อ)

**รูปที่** **หน้า**

# สารบัญรูป(ต่อ)

**รูปที่** **หน้า**

# คำย่อ/สัญลักษณ์

| คำย่อ/สัญลักษณ์ | คำอธิบาย |
|---|---|
| RUS | Random Under-sampling |
| AdaBoost | Adaptive Boosting |
| RUSBoost | Random Under-sampling integrated in the learning of AdaBoost. |
| SMOTE | Synthetic Minority Over-sampling Technique |
| DT | Decision Trees algorithm |
| NB | Naïve Bayes algorithm |
| SVM | Support Vector Machines algorithm |
| kNN | k-Nearest Neighbors algorithm |
| ANNs | Artificial Neural Networks algorithm |

# บทที่ 1

# บทนำ

## 1.1 ความเป็นมาและความสำคัญของปัญหา

ชุดข้อมูลที่ไม่สมดุล (Dataset Imbalanced) คือ ชุดข้อมูลที่มีการกระจายตัวของเซตคำตอบคลาสไม่สม่ำเสมอ หรือกล่าวได้ว่า ชุดข้อมูลที่มีจำนวนคลาสใดคลาสหนึ่งมีจำนวนเซตคำตอบที่มากกว่า หรือ น้อยกว่าคลาสอื่น ตัวอย่างที่เห็นได้ชัดเจนในการใช้งานแอปพลิเคชันจริง เช่น ข้อมูลของผู้ป่วย ข้อมูลการทำบัตรเครดิต ข้อมูลการทำนายอาชีพเกี่ยวกับสายงานเทคโนโลยี เป็นต้น ส่วนใหญ่จะพบว่าข้อมูลบางส่วนมีข้อมูลที่ขาดหายไป หรือ ข้อมูลที่เก็บจำนวนคลาสเซตคำตอบด้วยจำนวนที่ไม่เท่ากัน ทำให้ชุดข้อมูลเกิดความไม่สมดุลกัน ชุดข้อมูลที่ไม่สมดุลส่งผลให้เกิดการทำนายที่ผิดพลาดโดยเฉพาะในด้านการแพทย์ส่งผลกระทบเป็นอันตรายต่อชีวิตมนุษย์โดยตรงได้ และสามารถนำไปสู่การตัดสินใจที่ผิดพลาดได้อีกด้วย

ชุดข้อมูลที่ไม่สมดุลส่งผลให้โมเดลเกิดการเรียนรู้แบบอคติเอนเอียงไปทางคลาสส่วนใหญ่ ส่งผลให้ประสิทธิภาพโมเดล ความแม่นยำ ความถูกต้องของโมเดลไม่มีความน่าเชื่อถือ ด้านการนำโมเดลไปใช้การทำนายในคลาสส่วนใหญ่ถูกต้องแต่ในทางกลับกันกับคลาสส่วนน้อยกลับทำนายที่ผิดพลาด ส่งผลให้ไม่สามารถสรุปหาข้อมูลได้อย่างตรงไปตรงมา และไม่สามารถหาข้อมูลใหม่เชิงลึกได้ การตีความตัววัดประสิทธิภาพการประเมินโมเดลอาจจะได้ผลลัพธ์ที่ไม่ถูกต้องอีกด้วย จึงเป็นเหตุผลที่ต้องมีกระบวนการในเตรียมข้อมูลให้สมดุล เหมาะกับการนำไปใช้งาน และการสร้างโมเดลเลือกใช้อัลกอริทึมให้เหมาะกับจุดประสงค์การใช้งานในแต่ละชุดข้อมูลอย่างมีประสิทธิภาพ

โดยสรุปกล่าวได้ว่า ชุดข้อมูลเป็นสิ่งที่สำคัญสำหรับการพัฒนาโมเดลให้มีประสิทธิภาพ และสามารถตัดสินใจได้อย่างเป็นกลาง ไม่มีอคติ นี่จึงเป็นที่มาในการศึกษาหาวิธีการที่สามารถหาวิธีแก้ไขเรื่องของความไม่สมดุลของข้อมูล และการเลือกใช้อัลกอริทึมที่เหมาะให้กับโมเดล ทำให้โมเดลมีความถูกต้อง ไม่อคติ และสามารถนำไปประยุกต์ใช้งานจริงได้อย่างเหมาะสม หรือ การสร้างองค์ความรู้ใหม่โดยมีเหตุผลมารองรับองค์ความรู้ได้น่าเชื่อถือ

## 1.2 วัตถุประสงค์ของงานวิจัย

1) ศึกษาอัลกอริทึมที่เกี่ยวข้องกับการปรับชุดข้อมูลให้มีความสมดุล (data level algorithms)
2) เปรียบเทียบประสิทธิภาพการทำงานของเทคนิคการสุ่มตัวอย่างที่แตกต่างกันเมื่อใช้ร่วมกับอัลกอริทึมเรียนรู้ของเครื่อง (Machine Learning) กับชุดข้อมูลไม่สมดุล
3) เปรียบเทียบและวิเคราะห์ประสิทธิภาพของเทคนิคการสุ่มตัวอย่างกับจำนวนคุณลักษณะของข้อมูลในชุดข้อมูลที่ไม่สมดุล

## 1.3 ขอบเขตของงานวิจัย

1) ชุดข้อมูลที่ใช้จะเป็นชุดข้อมูลแบบ Binary Classification เท่านั้น
2) ชุดข้อมูลที่ใช้ในการเก็บผลการทดลองในการวิจัยมาจากเว็บไซต์ Kaggle
3) อัลกอริทึมที่ใช้ในการสร้างโมเดลประกอบด้วย 5 อัลกอริทึม ดังนี้ Decision Trees, Naïve Bayes, k-Nearest Neighbors, Support Vector Machines และ Artificial Neural Networks

## 1.4 ประโยชน์ที่คาดว่าจะได้รับ

1) เข้าใจหลักการของอัลกอริทึมเทคนิคการสุ่มตัวอย่างแบบต่างๆ และสามารถเลือกใช้ให้เหมาะสมกับอัลกอริทึมการเรียนรู้ในแต่ละแบบได้
2) ได้ทราบถึงจำนวนคุณลักษณะมีผลต่อเทคนิคการสุ่มตัวอย่างในการจำแนกประเภทของโมเดลอย่างไร

# บทที่ 2
# ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ชุดข้อมูลที่ไม่สมดุลเป็นปัญหาทั่วไปที่สามารถพบได้จากการเก็บชุดข้อมูลจากที่มีการใช้งานจริง ปัจจุบันการจัดการกับชุดข้อมูลไม่สมดุลนั้นถือว่าเป็นสิ่งที่ท้าทายที่จะหาวิธีการจัดการชุดข้อมูลอย่างไรให้เกิดความสมดุล และมีประสิทธิภาพต่อการนำไปใช้งาน วิเคราะห์ข้อมูล หรือการนำชุดข้อมูลเหล่านี้ไปใช้ฝึกสอนโมเดลให้มีการเรียนรู้ทำนายข้อมูลได้อย่างมีความแม่นยำ สำหรับทฤษฎีที่เกี่ยวข้องที่จะนำมากล่าวถึงในบทนี้ ทฤษฎี Decision Trees ทฤษฎี Naïve Bayes ทฤษฎี Support Vector Machines ทฤษฎี k-Nearest Neighbors และทฤษฎี Neural Network โดยจะเน้นไปที่การศึกษาการสร้างแบบ ANNs (Artificial Neural Networks) ที่ใช้อัลกอริทึม Backpropagation ทฤษฎีที่ใช้ในการแก้ไขปัญหาชุดข้อมูลไม่สมดุล ทฤษฎีการสุ่มตัวอย่างใหม่เช่น เทคนิคการสุ่มตัวอย่างแบบ SMOTE (Synthetic Minority Over-sampling Technique) เทคนิคการสุ่มตัวอย่างแบบ Tomek Links และเทคนิคที่ Ensemble Methods แบบ Boosting โดยใช้อัลกอริทึม AdaBoost (Adaptive Boosting) ที่ผสมผสานเทคนิคการสุ่มตัวอย่างแบบ Random Under Sampling และกล่าวถึงงานวิจัยที่เกี่ยวข้องเป็นแนวทางในการศึกษาของการแก้ไขปัญหาชุดข้อมูลที่ไม่สมดุล

เพื่อการศึกษาในงานวิจัยนี้ จึงได้รวบรวมทฤษฎี และงานวิจัยที่เกี่ยวข้องเป็นกรณีศึกษาสำหรับการดำเนินการทำงานวิจัย

## 2.1 Classification Model

Classification Model โมเดลการจำแนกประเภทเป็นส่วนหนึ่งในกระบวนการเรียนรู้ของเครื่อง (Machine Learning) ในกระบวนการเรียนรู้ของโมเดลประเภทนี้ คือ การวิเคราะห์ข้อมูล และจำแนกข้อมูลออกเป็นแต่ละประเภท โดยจะต้องมีเซตคำตอบของชุดข้อมูลนั้น โดยโมเดลจะเรียนรู้จากชุดข้อมูลฝึกสอนที่เตรียมไว้สำหรับการฝึกสอน เพื่อให้โมเดลทำนายชุดข้อมูล ให้ได้ตรงตามเซตคำตอบที่เตรียมไว้ ยิ่งโมเดลสามารถทำนาย หรือจำแนกประเภทข้อมูลได้ตรงเซตคำตอบ แสดงว่าโมเดลมีประสิทธิภาพมีความสามารถในเรียนรู้จดจำ และจำแนกประเภทได้เป็นอย่างดีสามารถตรวจวัดประสิทธิภาพความแม่นยำของโมเดลได้จาก Confusion Matrix

## 2.2 ความไม่สมดุลของชุดข้อมูล

ความไม่สมดุลของชุดข้อมูล คือ ชุดข้อมูลมีการกระจายตัวของข้อมูลมีขนาดไม่เท่าเทียมกัน เช่น การแบ่งกลุ่มชุดข้อมูลออกเป็นจำนวนสองกลุ่ม หรือมากกว่าสองกลุ่ม ถ้าในแต่ละกลุ่มมีจำนวนข้อมูลที่ไม่เท่ากัน จึงกล่าวได้ว่าชุดข้อมูลนั้นไม่สมดุล ชุดข้อมูลที่จำแนกประเภทออกมีจำนวนข้อมูลเป็นส่วนใหญ่ของประเภทคลาสทั้งหมด ถือว่าเป็นกลุ่มข้อมูลคลาสส่วนใหญ่ ชุดข้อมูลที่จำแนกประเภทออกมีจำนวนข้อมูลเป็นส่วนน้อยของประเภทคลาสทั้งหมด ถือว่าเป็นกลุ่มข้อมูลคลาสส่วนน้อย นำกลุ่มข้อมูลเหล่านี้มาใช้ในการวิเคราะห์ข้อมูล เพื่อให้โมเดลสามารถจำแนกประเภทข้อมูลได้ จึงจำเป็นที่จะต้องทำให้ชุดข้อมูลเกิดความสมดุลกัน โดยเราจะพิจารณาว่าควรจะลดจำนวนข้อมูลของกลุ่มคลาสส่วนใหญ่ หรือ เพิ่มจำนวนข้อมูลของกลุ่มคลาสส่วนน้อย เพื่อให้ชุดความข้อมูลเกิดความสมดุลง่ายต่อการนำไปสู่การวิเคราะห์ และการฝึกสอนโมเดลได้อย่างมีประสิทธิภาพ สามารถวัดอัตราส่วนความไม่สมดุลของชุดข้อมูลได้จากสมการ IR (Imbalanced Ratio)

$$IR = \frac{\text{number of majority class}}{\text{number of minority class}} \tag{2.1}$$

## 2.3 การสุ่มตัวอย่างข้อมูล (Resampling)

การสุ่มตัวอย่างข้อมูลเป็นวิธีสุ่มตัวอย่างเพื่อปรับเปลี่ยนขนาดข้อมูล หรือการกระจายตัวความถี่ของข้อมูล โดยวิธีการสุ่มตัวอย่างจะแบ่งออกเป็นสองวิธีหลัก คือ การที่สุ่มตัวอย่างกลุ่มคลาสส่วนน้อยเป็นจำนวนมาก คือ การเพิ่มตัวอย่างคลาสส่วนน้อยโดยการสุ่มตัวอย่าง การลดตัวอย่างจากคลาสส่วนใหญ่โดยการสุ่มตัวอย่าง คือ การลดขนาดข้อมูลในกลุ่มคลาสส่วนใหญ่ เพื่อที่จะสร้างวิธีการสุ่มตัวอย่างสร้างความสมดุลให้กับชุดข้อมูลให้โมเดลมีสามารถเรียนรู้ได้อย่างมีประสิทธิภาพมากยิ่งขึ้น โดยจะมีวิธีการสุ่มตัวอย่างได้หลากหลายวิธี ขึ้นอยู่กับการชุดข้อมูล และวัตถุประสงค์ของการดำเนินงาน

## 2.4 การคัดเลือกคุณลักษณะที่เกี่ยวข้อง (Feature Selection)

การเลือกคุณลักษณะเป็นกระบวนการในการเรียนรู้ของเครื่องและสถิติโดยเลือกคุณลักษณะที่เกี่ยวข้องและสำคัญที่สุด จากชุดข้อมูลเพื่อสร้างแบบโมเดลทำการวิเคราะห์ โดยเทคนิคหรือวิธีการเลือกคุณลักษณะ คือ การปรับปรุงประสิทธิภาพของโมเดลโดยการลดขนาด หรือจำนวนคุณลักษณะ โดยจะคัดเลือกคุณลักษณะที่เกี่ยวข้อง เพื่อปรับปรุงประสิทธิภาพโมเดลให้มีความสามารถตีความและจำแนกประเภทข้อมูล

### 2.4.1 การคัดเลือกคุณลักษณะแบบ ANOVA

การคัดเลือกคุณลักษณะแบบ ANOVA จะคำนวณจากค่า F-statistic และ p-value โดยจะจัดลำดับความสำคัญของคุณลักษณะที่มีค่า F-statistic สูงกว่า หรือค่า p-value ที่ต่ำจะถือว่ามีนัยสำคัญต่อการจำแนกประเภท การคัดเลือกคุณลักษณะจะใช้สำหรับการปรับปรุงประสิทธิภาพโมเดล หรือการสร้างโมเดลเพื่อวิเคราะห์และตีความเพิ่มเติมในการจำแนกประเภทข้อมูล

## 2.5 เทคนิคการสุ่มตัวอย่างแบบเพิ่มขนาดข้อมูล (Oversampling)

### 2.5.1 เทคนิค SMOTE

เทคนิคการสุ่มตัวอย่างแบบ SMOTE (Synthetic Minority Oversampling Technique) เป็นหนึ่งในวิธีที่ได้รับความนิยมมากที่สุด แตกต่างจากวิธีการสุ่มข้อมูลแบบสุ่มซ้ำจากกลุ่มตัวอย่างคลาสส่วนน้อย แนวคิดหลักของ SMOTE คือ การสร้างตัวอย่างตามระยะห่างของแต่ละข้อมูลในคลาสกลุ่มน้อยจากตัวอย่างเพื่อนบ้านที่ใกล้เคียงมากที่สุด โดยส่วนใหญ่จะคำนวณระยะห่างแบบ Euclidean ตามสมการ 2.2 ต่อไปนี้

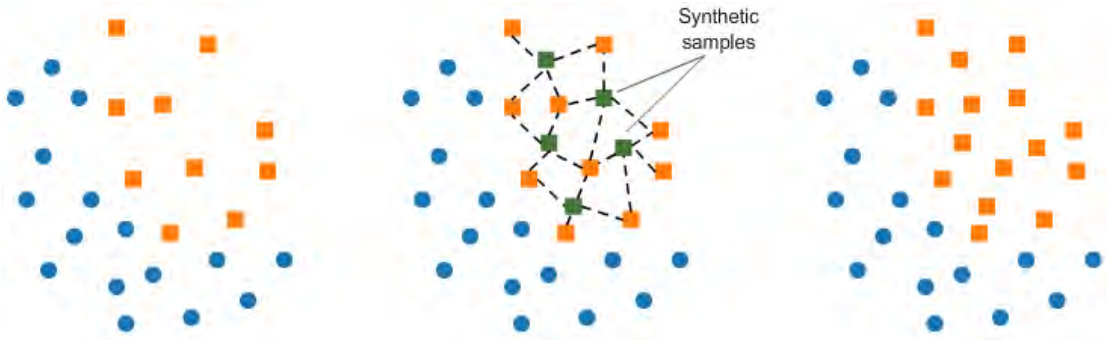
$$d(x, y) = \sqrt{\sum_{i=1}^{n} \left( x_i - y_i \right)^2} \qquad (2.2)$$

โดยที่ d คือ ระยะห่างของจุดข้อมูลระหว่างจุดข้อมูลสองจุด

n คือ จำนวนตัวอย่างคลาสส่วนน้อยทั้งหมด เมื่อกำหนดให้ i = 1, 2, 3, ..., n

$x_i$ คือ ค่าจุดข้อมูลที่ i ของข้อมูลตัวอย่างคลาสส่วนน้อย

$y_i$ คือ ค่าจุดข้อมูลที่ i ของข้อมูลตัวอย่างคลาสส่วนน้อยที่ใกล้เคียงกับจุด $x_i$




(ก) (ข) (ค)

**รูปที่ 2.1 แนวคิดกระบวนการอัลกอริทึมของ SMOTE**

ที่มา: https://www.kaggle.com/code/rafjaa/resampling-strategies-for-imbalanced-datasets/notebook

รูปที่ 2.1 แสดงแนวคิดกระบวนการอัลกอริทึมของ SMOTE แบบอย่างง่าย โดยที่รูป (ก) แสดงการกระจายตัวของข้อมูลของคลาสส่วนใหญ่ใช้สัญลักษณ์วงกลมแทนตัวอย่างคลาสส่วนใหญ่ และใช้สัญลักษณ์รูปสี่เหลี่ยมแทนตัวอย่างคลาสส่วนน้อย รูป (ข) แสดงการทำงานของ SMOTE จะสร้างตัวอย่างสังเคราะห์ใหม่ขึ้นมาจากการหาระยะห่างระหว่างจุดข้อมูลในคลาสส่วนน้อยที่ใกล้เคียง ซึ่งตัวอย่างใหม่ที่เพิ่มเข้ามาจะแทนที่ด้วยจุดสี่เหลี่ยมที่สร้างขึ้นมาใหม่ที่อยู่ระหว่างจุดสี่เหลี่ยมเดิม หรือจุดข้อมูลในคลาสเดิม รูป (ค) หลังจากที่ตัวอย่างสังเคราะห์ถูกเพิ่มเข้าในคลาสส่วนน้อยจะเห็นได้ว่าจำนวนตัวอย่างในแต่ละคลาสมีจำนวนเท่ากัน โดยการทำงานของ SMOTE มีจุดประสงค์เพื่อให้ชุดข้อมูลในแต่ละคลาสสมดุลกัน

ขั้นตอนการทำงานของอัลกอริทึม SMOTE แสดงดังรูปภาพที่ 2.2

**Algorithm** *SMOTE*(T, N, k)
**Input:** Number of minority class samples $T$; Amount of SMOTE $N$%; Number of nearest neighbors $k$
**Output:** $(N/100) * T$ synthetic minority class samples
1. (* *If N is less than 100%, randomize the minority class samples as only a random percent of them will be SMOTEd.* *)
2. **if** $N < 100$
3. **then** Randomize the $T$ minority class samples
4. $T = (N/100) * T$
5. $N = 100$
6. endif
7. $N = (int)(N/100)$ (* *The amount of SMOTE is assumed to be in integral multiples of 100.* *)
8. $k$ = Number of nearest neighbors
9. *numattrs* = Number of attributes
10. *Sample*[ ][ ]: array for original minority class samples
11. *newindex*: keeps a count of number of synthetic samples generated, initialized to 0
12. *Synthetic*[ ][ ]: array for synthetic samples
(* *Compute k nearest neighbors for each minority class sample only.* *)
13. **for** $i \leftarrow 1$ to $T$
14. Compute $k$ nearest neighbors for $i$, and save the indices in the *nnarray*
15. Populate($N$, $i$, *nnarray*)
16. **endfor**

*Populate*($N$, $i$, *nnarray*) (* *Function to generate the synthetic samples.* *)
17. **while** $N \neq 0$
18. Choose a random number between 1 and $k$, call it $nn$. This step chooses one of the $k$ nearest neighbors of $i$.
19. **for** $attr \leftarrow 1$ to *numattrs*
20. Compute: $dif = Sample[nnarray[nn]][attr] - Sample[i][$
21. Compute: $gap$ = random number between 0 and 1
22. $Synthetic[newindex][attr] = Sample[i][attr] + gap *$ $dif$
23. **endfor**
24. $newindex$++
25. $N = N - 1$
26. **endwhile**
27. **return** (* *End of Populate.* *)

**รูปที่ 2.2 ขั้นตอนการทำงานของอัลกอริทึม SMOTE**

ขั้นตอนการทำงานของอัลกอริทึม SMOTE มีขั้นตอนดังนี้

1. Input รับตัวแปรเข้ามา 3 ค่า ประกอบด้วย ตัวแปร T คือ จำนวนตัวอย่างที่ถูกสุ่มเลือกมาจากคลาสส่วนน้อย ตัวแปร k คือ จำนวนเพื่อนบ้านที่ใกล้เคียงกันมากที่สุดในการสร้างกลุ่มตัวอย่างใหม่ และตัวแปร N คือ จำนวนตัวอย่างสังเคราะห์ใหม่ โดยใช้เปอร์เซ็นต์ในการคำนวณ โดย Output คือ ตัวอย่างสังเคราะห์ที่สร้างขึ้นมาใหม่ (N/100) * T (เปอร์เซ็นต์ N คูณกับ T คือ ตัวอย่างสังเคราะห์ใหม่)

2. (บรรทัดที่ 1-7) กำหนดให้เงื่อนไขว่า ถ้า N น้อยกว่า 100 ให้ทำการสุ่มตัวอย่าง T ในคลาสส่วนน้อยเพื่อสุ่มเลือกตัวอย่างเดิมขึ้นมาเป็นจำนวน (N/100)*T ตัวอย่างในการสร้างตัวอย่างใหม่
3. (บรรทัดที่ 13-16) การวนรอบซ้ำโดยเข้าไปวนซ้ำใน T โดยให้การทำงานวนซ้ำคำนวณค่า k เพื่อนบ้านที่ใกล้เคียงเก็บไว้ในตัวแปร nnarray ที่เป็นอาร์เรย์หนึ่งมิติ และส่งค่า i, N และ nnarray ไปที่ Populate เป็นฟังก์ชันของการสร้างตัวอย่างสังเคราะห์ใหม่
4. (บรรทัดที่ 17-26) ฟังก์ชันการสร้างตัวอย่างสังเคราะห์ใหม่ Populate โดยการสร้างรอบการทำงานวนซ้ำโดยกำหนดเงื่อนไขเมื่อ N ไม่เท่ากับ 0 ภายในรอบการทำงานแต่ละรอบจะทำการสุ่มเลือกตัวเลข 1 ถึง k โดยเก็บค่าที่ได้ในตัวแปร nnarray ขั้นตอนนี้จะเป็นคำนวณหาค่าระยะห่างระหว่างสมาชิกและเพื่อนบ้านที่ถูกสุ่มเลือก วิธีการคำนวณสามารถคำนวณได้จากวิธี Euclidean เก็บค่าไว้ในตัวแปร diff ต่อมาทำการสุ่มเลือกค่า gap ระหว่าง 0-1 โดยค่า gap นี้จะถูกนำไปใช้คูณกับค่าความต่างระยะห่างบวกเข้ากับตำแหน่งตัวอย่างที่เดิม เพื่อสร้างตัวอย่างสังเคราะห์ใหม่ ทำวนซ้ำจนกว่าจะครบทุกค่าในตัวแปร nnarray โดยจะคืนค่าจำนวนตัวอย่างสังเคราะห์ใหม่ที่ได้เป็นผลลัพธ์ที่ได้จากอัลกอริทึมนี้

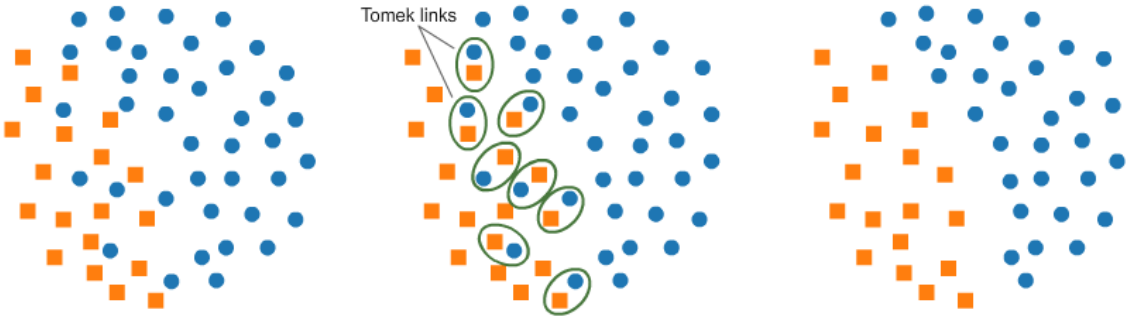
## 2.6 เทคนิคการสุ่มตัวอย่างแบบลดขนาดข้อมูล (Undersampling)

### 2.6.1 เทคนิค Random Under Sampling

เทคนิคการสุ่มตัวอย่างแบบ RUS (Random Under-sampling) เป็นเทคนิคที่สุ่มตัวอย่างลดจำนวนกลุ่มตัวอย่างคลาสส่วนใหญ่ให้มีขนาดเท่ากับจำนวนกลุ่มตัวอย่างคลาสส่วนน้อย กล่าวได้ว่าเป็นวิธีการสุ่มตัวอย่างเพื่อให้เกิดความสมดุลจากการกระจายตัวของข้อมูลในคลาสโดยการสุ่มตัวอย่างคลาสส่วนใหญ่ วิธีการสุ่มตัวอย่างออกจะเป็นการสุ่มแบบสลับลำดับข้อมูลในคลาสส่วนใหญ่เพื่อให้ได้ข้อมูลที่มีการกระจายตัวที่หลากหลาก เทคนิคนี้จะมีประสิทธิภาพก็ต่อเมื่อมีข้อมูลเป็นจำนวนมาก

### 2.6.2 เทคนิค Tomek Links

เทคนิค Tomek Links เป็นวิธีที่เหมาะกับการแก้ไขปัญหาแบบ Binary Classification แนวคิดของ Tomek Links ประกอบด้วยข้อมูลสองจุดที่อยู่ใกล้กันที่สุดของคลาสที่ตรงกันข้ามกันลิงค์จับคู่เข้าด้วยกัน และลบจุดที่เป็นข้อมูลของคลาสส่วนใหญ่ออกไปในแต่ละคู่ที่ลิงก์กัน เพื่อเพิ่มช่องว่างให้ข้อมูลกระจายตัวอย่างสมดุล และง่ายต่อการจำแนกประเภทข้อมูลของแต่ละคลาส




(ก) (ข) (ค)

**รูปที่ 2.3 แนวคิดกระบวนการอัลกอริทึมของ Tomek Links**

ที่มา: https://www.kaggle.com/code/rafjaa/resampling-strategies-for-imbalanced-datasets/notebook

รูปที่ 2.3 แสดงแนวคิดการทำงานของ Tomek Links รูป (ก) เป็นการแสดงการจะกระจายตัวของข้อมูลที่ไม่สมดุลระหว่างคลาสส่วนน้อย ใช้สัญลักษณ์แทนด้วยสี่เหลี่ยม และคลาสส่วนใหญ่ ใช้สัญลักษณ์แทนด้วยวงกลม ที่มีการทับซ้อนกันของจุดข้อมูล รูป (ข) เป็นการเข้าสู่กระบวนการของ Tomek Links โดยการจับคู่คลาสตรงข้ามที่ใกล้เคียงกัน ต้องการที่จะลดจำนวนกลุ่มตัวอย่างคลาสส่วนใหญ่ เพื่อให้ข้อมูลทั้งสองคลาสสมดุลกัน รูป (ค) เมื่อลบข้อมูลจุดข้อมูลคลาสส่วนใหญ่ในแต่ละคู่ของ Tomek Links จะเห็นได้ว่าจุดข้อมูลระหว่างคลาสสองคลาสมีการกระจายตัวที่แบ่งแยกกันอย่างเห็นได้ชัดเจน และง่ายต่อการจำแนกประเภทคลาส

ขั้นตอนการทำงานของอัลกอริทึม Tomek links มีขั้นตอนดังนี้

1. การระบุตัวอย่างที่ลิงก์คู่กันของ Tomek links โดยจะต้องมีหนึ่งตัวอย่างจากคลาสส่วนใหญ่ และหนึ่งตัวอย่างคลาสส่วนน้อยที่เป็นเพื่อนบ้านกัน โดยเงื่อนไขในการจับคู่ของ Tomek link สมมติว่ามีตัวอย่าง A และ B จะเป็น Tomek links ได้จะต้องไม่ตัวอย่าง C ที่ระยะห่างระหว่าง A และ C น้อยกว่าหรือเท่ากับ A และ B
2. การลบตัวอย่างในคลาสส่วนใหญ่ของ Tomek links หลังจากที่จับคู่ตัวอย่างได้ ตัวอย่างจากคลาสส่วนใหญ่ในแต่ละคู่ของ Tomek links จะถูกลบออก ทำให้ชุดข้อมูลมีขนาดเล็กลง เนื่องจากจำนวนตัวอย่างคลาสส่วนใหญ่ลดลง

## 2.7 เทคนิค Ensemble Method

เทคนิคที่รวบรวมการทำนายของโมเดลหลายโมเดล เพื่อทำการทำนายผลลัพธ์มีความแม่นยำและมีประสิทธิภาพมากยิ่งขึ้น การใช้ ensemble จะเข้ามาช่วยในการจัดการปรับปรุงงประสิทธิภาพโมเดล โดยจะมีสองวิธีส่วนใหญ่ที่นิยมใช้ดังนี้

### 2.7.1 เทคนิคการเพิ่มประสิทธิภาพโมเดลแบบ Bagging (Bootstrap Aggregating)

เป็นเทคนิคที่ใช้ในฝึกสอนโมเดลหลายโมเดลที่เหมือนกันแต่ใช้ชุดข้อมูลการฝึกสอนที่แตกต่างกัน ชุดข้อมูลที่นำมาฝึกสอนจะถูกสุ่มโดยใช้วิธีการสุ่มแบบแทนที่ (Bootstrap) หลังจากที่มีการฝึกสอนและทำนายของแต่ละโมเดลครบทุกโมเดล จะทำการเลือกโมเดลที่ดีสุดโดยใช้วิธีการหาค่าเฉลี่ยออก หรือทำการลงมติเสียงส่วนใหญ่ (Majority vote) ในการเลือกโมเดล โดยอัลกอริทึมที่นิยมใช้ในเทคนิค bagging คือ Random Forest

**Algorithm 1** Random Forest

**Precondition:** A training set $S := (x_1, y_1), \ldots, (x_n, y_n)$, features $F$, and number of trees in forest $B$.

```
function RandomForest(S, F)
    H ← ∅
    for i ∈ 1, ..., B do
        S^(i) ← A bootstrap sample from S
        h_i ← RandomizedTreeLearn(S^(i), F)
        H ← H ∪ {h_i}
    end for
    return H
end function
function RandomizedTreeLearn(S, F)
    At each node:
        f ← very small subset of F
        Split on best feature in f
    return The learned tree
end function
```

รูปที่ 2.4 ขั้นตอนการทำงานของอัลกอริทึม Random Forest

ขั้นตอนการทำงานของอัลกอริทึม Random Forest มีขั้นตอนดังนี้

1. สุ่มข้อมูลแบบ bootstrapping จากชุดข้อมูลทั้งหมด ให้ได้จำนวนชุดข้อมูล n ชุดข้อมูล หรือตามจำนวนโมเดล Decision Treesที่กำหนดไว้
2. Random Forest จะสร้างโมเดล Decision Trees โดยที่แต่ละโมเดล Decision Trees มีประสิทธิภาพการเรียนรู้ที่ต่ำ แต่เมื่อนำมาทำนายร่วมกันเพื่อสร้างโมเดลที่มีประสิทธิภาพการทำนายที่แม่นยำกว่าการใช้โมเดล Decision Trees เพียงโมเดลเดียว
3. ประเภทคลาสที่ได้รับการจำแนกในแต่ละโมเดล Decision Trees ลงมติเสียงส่วนใหญ่ ประเภทคลาสที่ได้รับเสียงข้างมากจะถือว่าเป็นคลาสที่จำแนกได้จากการทำนายของโมเดล Random Forest

### 2.7.2 เทคนิคการเพิ่มประสิทธิภาพโมเดลแบบ Boosting

เป็นหนึ่งในวิธีการที่ใช้ใน Machine Learning เพื่อเพิ่มประสิทธิภาพการทำนายของโมเดลให้มีความแม่นยำมากยิ่งขึ้น โดยการปรับปรุงโมเดลที่มีประสิทธิภาพการเรียนรู้ระดับต่ำให้กลายเป็นโมเดลที่มีการเรียนรู้ที่มีประสิทธิภาพในระดับสูง แนวคิดหลักของวิธีนี้ คือ การกำหนดน้ำหนักในการถ่วงน้ำหนักแต่ละจุดข้อมูลตัวอย่างน้ำหนักเท่ากัน หรือน้ำหนักตามการกระจายตัวข้อมูล เมื่อข้อมูลเหล่านี้ถูกนำเข้าไปฝึกสอนในโมเดลที่มีการเรียนรู้ประสิทธิภาพต่ำ อัลกอริทึม Boosting จะเพิ่มน้ำหนักของจุดข้อมูลตัวอย่างที่ทำนายผิดพลาด และจุดข้อมูลตัวอย่างที่ถูกปรับการจากถ่วงน้ำหนักจะถูกส่งไปยังโมเดลถัดไป คือ การทำซ้ำกระบวนการนี้จะถูกกำหนดการทำซ้ำไว้ล่วงหน้า หรือจนกว่าจะได้ประสิทธิภาพโดยรวมที่ดีในระดับหนึ่ง ยกตัวอย่างอัลกอริทึมของ Boosting ที่ได้รับความนิยมมากที่สุดอย่างหนึ่งคือ AdaBoost (Adaptive Boosting)

#### 1) อัลกอริทึม AdaBoost

เป็นอัลกอริทึมที่ได้รับความนิยมใช้สำหรับข้อมูลประเภท Classifier และ Regression เป็นอัลกอริทึมที่ทำงานได้ดีกับปัญหาประเภทการจำแนก Binary และ Multiclass classification โดยจะเน้นไปที่การจัดการตัวอย่างที่จำแนกคลาสได้ยาก และให้น้ำหนักกับตัวอย่างที่ทำนายผลผิดพลาดสูงกว่าตัวอย่างที่ทำนายผลได้แม่นยำและปรับปรุงประสิทธิภาพการจำแนกคลาสโดยรวม

**Algorithm 1** AdaBoost for binary classification

**Precondition:** A training set $S := (x_1, y_1), \ldots, (x_n, y_n)$, hypothesis space $\mathcal{H}$, and number of iterations $T$.

1 **for** $i \in \{1, 2 \ldots, n\}$ **do**
2 $\quad \mathcal{D}_1(i) \leftarrow \frac{1}{n}$
3 **end for**
4 $H \leftarrow \emptyset$
5 **for** $t = 1, \ldots, T$ **do**
6 $\quad h_t \leftarrow \operatorname{argmin}_{h \in \mathcal{H}} P_{i \sim \mathcal{D}_t}(h(x_i) \neq y_i)$ ▷ find good hypothesis on weighted training set
7 $\quad \epsilon_t \leftarrow P_{i \sim \mathcal{D}_t}(h_t(x_i) \neq y_i)$ ▷ compute hypothesis's error
8 $\quad \alpha_t \leftarrow \frac{1}{2} \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$ ▷ compute hypothesis's weight
9 $\quad H \leftarrow H \cup \{(\alpha_t, h_t)\}$ ▷ add hypothesis to the ensemble
10 $\quad$ **for** $i \in \{1, 2 \ldots, n\}$ **do** ▷ update training set distribution
11 $\quad\quad \mathcal{D}_{t+1}(i) \leftarrow \frac{\mathcal{D}_t(i)\, e^{-\alpha_t y_i h_t(x_i)}}{\sum_{j=1}^{n} \mathcal{D}_t(j)\, e^{-\alpha_t y_j h_t(x_j)}}$
12 $\quad$ **end for**
13 **end for**
14 **return** $H$

**รูปที่ 2.5 ขั้นตอนการทำงานของอัลกอริทึม AdaBoost**

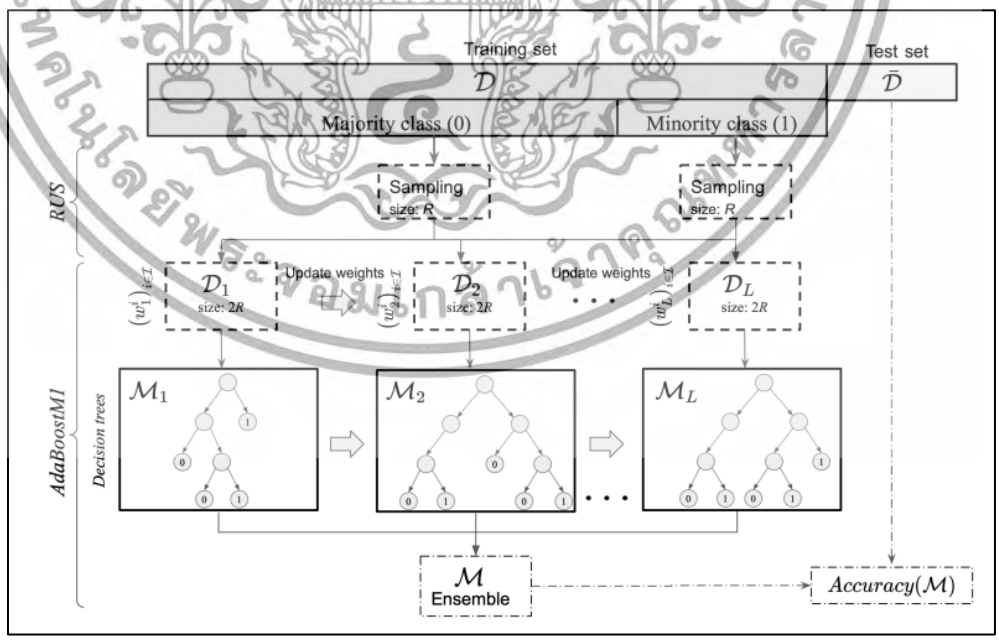
ที่มา: https://mbernste.github.io/files/notes/AdaBoost.pdf

ขั้นตอนการทำงานของอัลกอริทึม AdaBoost มีขั้นตอนดังนี้

1. กำหนดน้ำหนักสำหรับแต่ละตัวอย่างในการฝึกสอนโมเดล ให้น้ำหนักที่จุดข้อมูล $D_i = 1/n$ โดยที่ n คือจำนวนตัวอย่างการฝึกสอนโมเดล
2. ฝึกสอนโมเดลที่มีประสิทธิภาพต่ำโดยใช้การฝึกข้อมูลที่มีน้ำหนัก และคำนวณค่า error ทำนายผลที่ผิดพลาด ของโมเดลประสิทธิภาพต่ำ และคำนวณผลรวมน้ำหนักของโมเดลประสิทธิภาพต่ำ เพื่อที่จะถ่วงน้ำหนักใหม่ในการฝึกสอนรอบถัดไป
3. ปรับเปลี่ยนน้ำหนักใหม่สำหรับการฝึกสอนโมเดลในรอบถัดไปจนกว่าจะได้โมเดลที่มีประสิทธิภาพการทำนายที่สูง

**2) อัลกอริทึม RUSBoostClassifier**

อัลกอริทึมการจำแนกประเภทที่รวมสองเทคนิคระหว่าง Random Under sampling และ AdaBoost เป็นอัลกอริทึมที่ถูกออกแบบเพื่อใช้สำหรับแก้ไขปัญหาความไม่สมดุลของคลาสในประเภทการจำแนกแบบ Binary classification แนวคิดหลักของอัลกอริทึมนี้ คือ การรวมหลักการของ boosting และการสุ่มเลือกข้อมูลในชุดข้อมูล (random under-sampling) เพื่อสร้างวิธีการที่เพิ่มประสิทธิภาพการทำนาย RUSBoost เป็นเทคนิคที่นำอัลกอริทึม AdaBoost (Adaptive Boosting) เข้ามาขยายเพื่อเพิ่มประสิทธิภาพการทำงานของโมเดลอัลกอริทึมนี้เข้ามาช่วยในการจัดการกับปัญหาความไม่สมดุลและการจำแนกข้อมูลที่ไม่สมดุล




**รูปที่ 2.6 แผนภาพแสดงขั้นตอนการทำงานของอัลกอริทึม RUSBoostClassifier**

ที่มา: https://www.biorxiv.org/content/10.1101/2021.10.06.463434v1.full.pdf

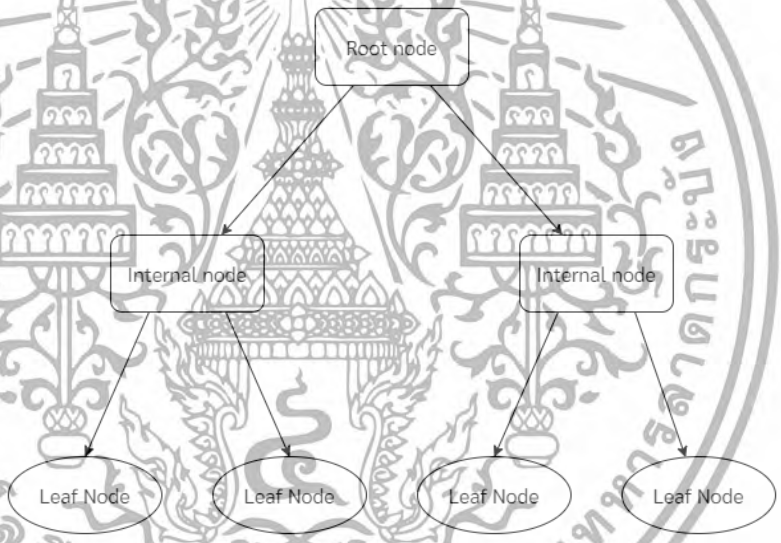
ขั้นตอนการทำงานของอัลกอริทึม RUSBoostClassifier มีขั้นตอนดังนี้

1. การกำหนดน้ำหนักที่เท่ากันให้กับแต่ละตัวอย่างในชุดข้อมูลสำหรับฝึกสอนโมเดล ให้ชุดข้อมูลมีน้ำหนัก หรือกล่าวได้ว่าเป็นการถ่วงน้ำหนักชุดข้อมูล และสร้าง ensemble สำหรับเก็บโมเดลที่มีประสิทธิภาพการทำนายที่ต่ำ
2. ทำการ Boosting ในแต่ละรอบเพื่อฝึกให้โมเดลที่มีประสิทธิภาพการจำแนกที่ต่ำ โดยมุ่งเน้นไปที่ตัวอย่างของคลาสส่วนน้อย โดยในแต่ละรอบจะโมเดลจะถูกฝึกด้วยชุดข้อมูลฝึกสอนที่ถ่วงน้ำหนัก โดยตัวโมเดลที่มีประสิทธิภาพการจำแนกที่ต่ำ หรือโมเดลจำแนกอย่างง่าย เช่น Decision Trees มีการแบ่งแบบเดียวไม่ซับซ้อน (Decision Stumps)
3. เมื่อโมเดลทำนายผลลัพธ์จะการคำนวณน้ำหนักของโมเดลที่มีประสิทธิภาพต่ำ เป็นการวัดว่าโมเดลสามารถทำงานได้ดีเพียงใดในชุดข้อมูลถ่วงน้ำหนักในปัจจุบัน โดยอ้างอิงจากอัตราความผิดพลาด โมเดลที่จำแนกได้อย่างแม่นยำจะได้รับน้ำหนักที่สูงกว่า หลังจากที่คำนวณน้ำหนักจะทำการเปลี่ยนแปลงน้ำหนักในการถ่วงชุดข้อมูลที่จำแนกประเภทไม่ถูกต้อง เพื่อสร้างแนวโน้มที่จะถูกเลือกในรอบการวนซ้ำถัดไป
4. ทำการสุ่มตัวอย่างจากคลาสส่วนใหญ่เพื่อสร้างชุดข้อมูลที่ถ่วงน้ำหนักสมดุลกัน จำนวนตัวอย่างในคลาสส่วนใหญ่ที่จะถูกเลือกเป็นชุดข้อมูลใหม่จะถูกกำหนดจำนวนจากอัตราความสมดุลในปัจจุบันซึ่งสามารถเปลี่ยนแปลงไปได้ตามแต่ละรอบของการวนซ้ำ Boosting ทำการเพิ่มโมเดลที่มีประสิทธิภาพการจำแนกที่ต่ำที่ผ่านการฝึกสอนเข้าไปใน ensemble
5. Ensemble สุดท้ายถูกสร้างขึ้นจากการรวมการทำนายของโมเดลที่มีประสิทธิภาพการจำแนกที่ต่ำทั้งหมดใน ensemble

## 2.8 Decision Trees

Decision Trees แนวคิดของอัลกอริทึม Decision Trees คือ การใช้หลักการInformation Gain การสร้างกฎขึ้นมาโดยใช้เงื่อนไข if-else ถ้าเป็นจริงกับกฎข้อเงื่อนไขที่กำหนดให้ดำเนินการต่อ แต่ถ้าไม่เป็นจริงให้พิจารณากฎที่ข้อเงื่อนไขถัดไปจนกว่าจะถึงเงื่อนไขสุดท้ายในประเภทที่จำแนกได้ Decision Trees เป็นหนึ่งในอัลกอริทึมที่นิยมใช้กับโมเดลจำแนกประเภท (Classification Model) เพราะสามารถบอกที่มาของเหตุผลในการตัดสินใจในการทำนายผลลัพธ์

Decision Trees จำแนกข้อมูลในรูปแบบโครงสร้างต้นไม้แบบลำดับชั้น ซึ่งประกอบด้วย โหนดรากจุดเริ่มต้นของ Decision Trees หรือคุณลักษณะที่ต้องพิจารณาก่อนที่แยกกิ่งก้านสาขาไป โหนดถัดไปตามเงื่อนไขที่กำหนด กิ่งก้านสาขา คือค่าคุณลักษณะที่สอดคล้องในแต่ละเงื่อนไข กิ่งก้านสาขา โหนดภายใน คือกฎเงื่อนไขที่จะทดสอบค่าข้อมูลแต่ละคุณลักษณะ และโหนดใบ กำหนดการจำแนกประเภท




**รูปที่ 2.7 แผนภาพแสดงแนวคิดการทำงานของอัลกอริทึม Decision Trees**

รูปที่ 2.7 แผนภาพแสดงแนวคิดการทำงานของอัลกอริทึม Decision Trees โดยจะประกอบด้วย โหนดรากที่แบ่งแยกกิ่งก้านสาขาเป็นไปตามความสอดคล้องค่าข้อมูลแต่ละคุณลักษณะมาพิจารณาเงื่อนไขถัดในโหนดภายในโหนดภายในจะคอยทดสอบค่าของแต่ละคุณลักษณะควรจะจำแนกเป็นประเภทใดในโหนดใบที่กำหนดการจำแนกประเภท

ขั้นตอนการทำงานของอัลกอริทึม Decision Trees มีขั้นตอนดังนี้

1. กำหนดโหนดรากคุณลักษณะที่ดีที่สุดสำหรับการพิจารณาตัดสินใจสำหรับโหนดถัดไป อัลกอริทึมจะเลือกคุณลักษณะที่ดีที่สุด จากการคำนวณ Entropy การแบ่งคุณลักษณะที่เป็นไปได้ในแต่ละโหนด คำนวณได้จากสมการ 2.3 ดังนี้

$$\text{Entropy}(S) = -\sum_{i=1}^{c} p_i \cdot \log_2 p_i \qquad (2.3)$$

โดยที่ Entropy คือ การวัดความไม่แน่นอนในชุดข้อมูล

c คือ จำนวนประเภทคลาสในชุดข้อมูล

$p_i$ คือ สัดส่วนของตัวอย่างในคลาส i ในชุดข้อมูล

Entropy ถูกคำนวณสำหรับการแบ่งที่เป็นไปได้จากสัดส่วนคลาสต่อจำนวนตัวอย่างทั้งหมด ขั้นต่อไปจะเป็นการพิจารณาหาคุณลักษณะใดเหมาะสมกับการเป็นโหนดราก โดยใช้การ Information Gain สำหรับการแบ่งว่าคุณลักษณะใดควรเป็นโหนดราก และคุณลักษณะใดควรเป็นโหนดภายในหรือโหนดลูก จากสมการ IG ดังสมการที่ 2.4 ดังนี้

$$\text{IG}(S, A) = \text{Entropy}(S) - \sum_{j=1}^{A} \frac{S_j}{S} \cdot \text{Entropy}(S_j) \qquad (2.4)$$

โดยที่ Entropy(S) คือ Entropy ของโหนดราก

Entropy($S_j$) คือ Entropy ของโหนดลูกที่ j

S คือ จำนวนตัวอย่างทั้งหมดในโหนดราก

$S_j$ คือ จำนวนตัวอย่างในโหนดภายใน หรือโหนดลูกที่ j

A คือ จำนวนค่าคุณลักษณะที่กำลังพิจารณา

Information Gain ตัดสินใจเลือกคุณลักษณะที่ดีที่สุดได้จาก โหนดที่มีค่า IG สูงที่สุดในแต่ละคุณลักษณะ

2. เมื่อได้โหนดรากค่าคุณลักษณะจะแยกกิ่งสาขาในการสร้างโหนดภายในสำหรับการพิจารณาคุณลักษณะถัดไปที่สอดคล้องกันกับค่าคุณลักษณะจากโหนดราก โดยจะทำวนซ้ำจนกว่าจะสามารถจำแนกกำหนดโหนดใบได้อย่างสมบูรณ์ และจะวนซ้ำไปยังกิ่งสาขาอื่นจนกว่าจะครบสาขา ไปจนถึงกำหนดโหนดใบให้สาขาได้สมบูรณ์

## 2.9 Naïve Bayes

Naïve Bayes เป็นอัลกอริทึมที่ใช้หลักการความน่าจะเป็นในการจำแนกประเภทคลาส โดยจะพิจารณาข้อมูลต่อไปนี้ในการหาความน่าจะเป็นในการจำแนกคลาสข้อมูล สามารถอธิบายหลักการได้ด้วยสมการ 2.5

$$P(class \mid attribute) = \frac{P(attribute \mid class)\, P(class)}{P(attribute)} \tag{2.5}$$

โดยที่ P(class | attribute) คือ ความน่าจะเป็นของ class คลาสใดคลาสใดคลาสหนึ่งที่พิจารณาจาก attribute ที่ต้องการ

P(attribute | class) คือ ความน่าจะเป็นของข้อมูลที่อยู่ใน attribute ที่อยู่ในคลาสใดคลาสหนึ่ง

P(class) คือ ความน่าจะเป็นทั้งหมดของ class

P(attribute) คือ ความน่าจะเป็นทั้งหมดของ attribute

ในกรณีที่มีคุณลักษณะหรือปัจจัยที่มีผลต่อการจำแนกประเภทคลาสมากกว่าหนึ่งสามารถเขียนให้อยู่ในรูปแบบสมการ 2.6

$$P(class \mid a_1, a_2, a_3, \ldots, a_n) = \frac{P(a_1, a_2, a_3, \ldots, a_n \mid class)\, P(class)}{P(a_1, a_2, a_3, \ldots, a_n)} \tag{2.6}$$

โดยที่ $a_i$ แทนคุณลักษณะแต่ละคุณลักษณะที่มีผลต่อการจำแนก เมื่อ i = 1, 2, 3, ... , n

การจำแนกประเภทคลาสจากการหาความน่าจะเป็นจากทฤษฎีเบย์ สามารถจำแนกประเภทคลาสได้ง่าย ในกรณีที่มีการทราบข้อมูลระหว่างสองคลาสที่ระบุไว้ชัดเจน ตัวอย่างเช่น ชุดข้อมูลที่ต้องการทำนายจะออกเป็นไปว่ายน้ำในกรณีที่ฝนไม่ตกให้เป็น class can และจะไม่ออกไปว่ายน้ำในกรณีที่ฝนตก class cannot สามารถคำนวณหาความน่าจะเป็นภายใต้เงื่อนไขของคุณลักษณะหรือปัจจัยที่จะเกิดขึ้นในคลาสใดคลาสหนึ่งได้จากสมการ 2.7

$$Classify(data)_{NB} = \operatorname{argmax}_{c_j \in C} P(c_j) \prod_{i=1}^{n} P(a_i \mid c_j) \tag{2.7}$$

โดยที่ $c_j$ แทนจำนวนคลาสที่สามารถจำแนกได้ เมื่อ j = 1, 2, 3, ... , n

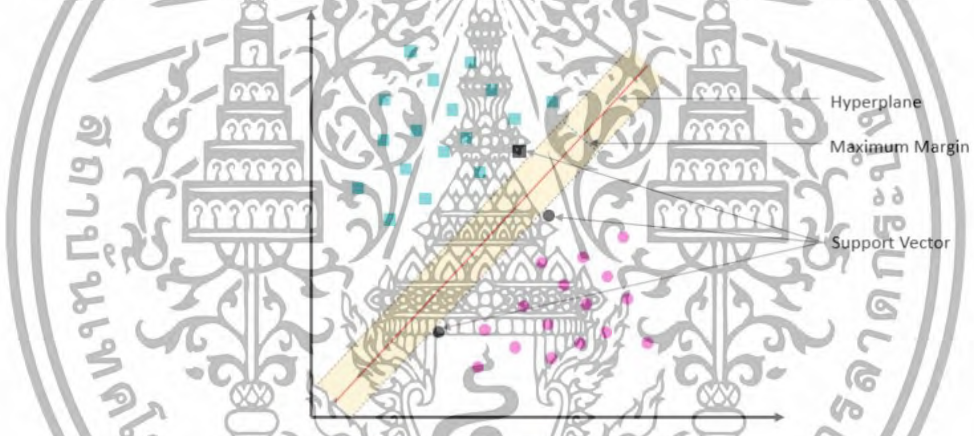
จากสมการที่ 2.7 สามารถจำแนกประเภทคลาสได้จากการหาความน่าจะเป็นระหว่าง 2 คลาสว่าคลาสใดมีความน่าจะเป็นสูงสุด จึงจำแนกประเภทเป็นคลาสที่มีความน่าจะเป็นสูง

## 2.10 Support Vector Machines

SVM (Support Vector Machines) เป็นอัลกอริทึมการเรียนรู้ของเครื่อง (Machine Learning) ที่ใช้สำหรับโมเดลการจำแนกประเภท Classification หรือการทำนายค่าตัวเลข Regression เหมาะสมกับชุดข้อมูลที่มีคุณลักษณะจำนวนมากแนวคิดของ SVM คือการค้นหาเส้นตรงที่ดีที่สุดในการแบ่งข้อมูลจำแนกข้อมูลเป็นแต่ละกลุ่มคลาส หรือประเภท

ขั้นตอนกระบวนการทำงานของอัลกอริทึม SVM มีขั้นตอนดังนี้

1. เส้นตรงแบ่งข้อมูล (Hyperplane) จะเป็นเส้นตรงที่แบ่งข้อมูลออกเป็นกลุ่มคลาส จุดข้อมูลที่อยู่ใกล้เส้นแบ่งจะถูกใช้กำหนดเส้นขอบเขตของแต่ละกลุ่มข้อมูลคลาส จะเรียกจุดที่อยู่ใกล้เส้นแบ่งขอบว่าเวกเตอร์ที่สนับสนุนเส้นแบ่งข้อมูล
2. ระยะขอบ (Margin) คือระยะห่างระหว่าง Hyperplane กับจุดข้อมูลที่อยู่ใกล้เส้นแบ่งที่สุดจากในแต่กลุ่มคลาสข้อมูลโดยจะหาเส้นที่มีระยะห่างขอบมากที่สุดเพื่อให้ครอบคลุมในการทำนายข้อมูลที่ไม่เคยเห็นมาก่อนในการจำแนกแบ่งกลุ่มคลาส ดังรูปที่ 2.8

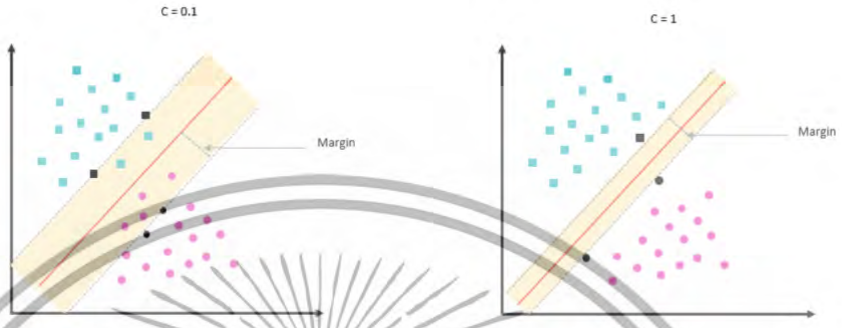



**รูปที่ 2.8 แสดงชุดข้อมูลที่จำแนกได้ 2 กลุ่มคลาสผ่านอัลกอริทึม SVM**

ที่มา: https://www.nectec.or.th/wp-content/uploads/2022/08/CPS-ML-manufacturing.pdf

รูปที่ 2.8 แสดงชุดข้อมูลที่จำแนกได้ 2 กลุ่มคลาสผ่านอัลกอริทึม SVM โดยสัญลักษณ์สี่เหลี่ยมแทน Positive class และสัญลักษณ์วงกลมแทน Negative class โดยจะมี Hyperplane ข้อมูลแต่ละคลาส และจุดข้อมูลที่อยู่ใกล้เส้นตรงแบ่งข้อมูลที่มีระยะเส้นขอบมากที่สุดในการครอบคลุมถึงจุดข้อมูลที่ไม่เคยเห็น เพื่อให้สามารถทำนายการจำแนกกลุ่มคลาสได้

3. พารามิเตอร์ C คือตัวแปรสำคัญที่ใช้ในการกำหนดระยะขอบ ถ้ายิ่งกำหนดให้ค่าพารามิเตอร์ C มีค่ามาก ระยะขอบจะมีระยะที่แคบลง ทำให้การแบ่งจุดข้อมูลในแต่ละกลุ่มคลาสมีความแม่นยำมากยิ่งขึ้น ในกรณีที่พารามิเตอร์ C มีค่าน้อยระยะขอบมีระยะที่กว้างมากขึ้น ทำให้การแบ่งจุดข้อมูลในแต่ละกลุ่มคลาสมีความแม่นยำน้อยลง




(ก) ข)

**รูปที่ 2.9 แสดงค่าพารามิเตอร์ C ที่มีผลต่อการกำหนดระยะขอบขนาดเส้นแบ่ง**

ที่มา: https://www.nectec.or.th/wp-content/uploads/2022/08/CPS-ML-manufacturing.pdf

รูปที่ 2.9 แสดงค่าพารามิเตอร์ C ที่มีผลต่อการกำหนดระยะขอบขนาดเส้นแบ่งข้อมูลโดยรูป (ก) ปรับให้ค่าพารามิเตอร์ C มีค่าน้อย ระยะขอบมีขนาดที่กว้างมากขึ้น และรูป (ข) ปรับให้คาพารามิเตอร์ C มีค่ามากขึ้น ระยะขอบมีขนาดที่แคบลง เห็นได้ชัดจากทั้ง 2 รูปว่าระยะขอบ คือขนาดเส้นแบ่งข้อมูลมีผลต่อการแบ่งจำแนกข้อมูลในแต่ละกลุ่มคลาส

4. ในกรณีที่จุดข้อมูลแต่ละจุดมีความซับซ้อนไม่สามารถจะแก้ปัญหาผ่านการสร้างเส้นตรงในการแบ่งข้อมูลแต่ละกลุ่มคลาส SVM จะใช้เทคนิค Kernel เป็นเทคนิคที่ช่วยในการจำแนกข้อมูลที่ไม่สามารถแยกได้ชัดว่าควรจะจำแนกกลุ่มคลาสใดได้อย่างมีประสิทธิภาพ

## 2.11 k-Nearest Neighbors

k-Nearest Neighbors เป็นวิธีการจำแนกประเภทคลาสข้อมูลที่สนใจโดยใช้ความคล้ายคลึงของข้อมูล หรือเพื่อนบ้านที่ใกล้เคียงจำนวน k ตัว โดยกระบวนการเรียนรู้ของอัลกอริทึมนี้ คือการเรียนรู้จากชุดข้อมูลตัวอย่างที่มีอยู่ นิยมใช้กับโมเดลประเภทการเรียนรู้แบบถดถอย Regression โดยการค่าค่าเฉลี่ยที่ใกล้เคียงที่สุดจากเพื่อนบ้านจำนวน k ตัว และโมเดลแบบ Classification จะใช้เสียงมีข้างมากในการจำแนกข้อมูลควรไปอยู่คลาสใด

ขั้นตอนกระบวนการทำงานของอัลกอริทึม k-Nearest Neighbors มีขั้นตอนดังนี้

1. กำหนดค่า k จำนวนเพื่อนบ้านที่ใกล้เคียงกับจุดข้อมูลที่ต้องการจำแนกประเภทคลาส
2. คำนวณหาระยะห่างระหว่างจุดข้อมูลที่ต้องการจำแนกประเภทคลาสกับกลุ่มตัวอย่างข้อมูลในแต่ละคลาสที่เป็นเพื่อนบ้านที่ใกล้เคียงที่สุด สามารถคำนวณหาระยะห่างระหว่างจุดข้อมูลได้จากสมการ Euclidean ดังสมการ 2.8

$$\text{distance} = \sqrt{\sum_{i=1}^{n} \left( p_i - q_i \right)^2} \qquad (2.8)$$

โดยที่ distance คือ ระยะห่างของจุดข้อมูลระหว่างจุดข้อมูลสองจุด

n คือ จำนวนคุณลักษณะทั้งหมด เมื่อกำหนดให้ i = 1, 2, 3, ..., n

$p_i$ คือ ค่าจุดข้อมูลที่ i ของข้อมูลกลุ่มตัวอย่างที่ใกล้เคียงกับจุด $q_i$

$q_i$ คือ ค่าจุดข้อมูลที่ i ของข้อมูลต้องการทำนายจำแนกประเภทคลาส

3. เมื่อได้ระยะห่างจุดข้อมูลเพื่อนบ้านที่ใกล้เคียงทั้งหมด จะทำการเรียงค่าระยะห่างจากน้อยไปมาก และสังเกตค่า k ว่ากำหนดเท่าใด ตัวอย่างเช่น หากกำหนดค่า k = 5 ค่าระยะห่างระหว่างจุดข้อมูลที่ต้องการทำนายกับจุดข้อมูลกลุ่มตัวอย่างที่มีระยะห่างน้อยที่สุด 5 อันดับแรก
4. จุดข้อมูลเพื่อนบ้านที่ใกล้เคียงมากที่สุดตามจำนวนค่า k กลุ่มเพื่อนบ้านที่กำหนดนำมาเทียบกับกลุ่มตัวอย่างข้อมูลในแต่ละคลาสว่าจุดข้อมูลที่ต้องการทำนายควรจำแนกอยู่คลาสใด โดยทำการโหวตเสียงข้างมากที่สุดในการกำหนดประเภทคลาสให้กับจุดข้อมูลที่ต้องการทำนายจำแนกประเภทคลาส

## 2.12 Artificial Neural Networks

Neural Network เป็นการสร้างอัลกอริทึมให้มีความสามารถในการเรียนรู้ที่ซับซ้อนเหมือนระบบประสาทของมนุษย์ การสร้างประสาทเทียมจากโมเดล Perceptron โดยการนำ Perceptron หลากหลายจำนวนรวมเข้าด้วยกันเป็นพื้นฐาน Neural Networks โมเดล Artificial Neural Networks เหมาะสมกับชุดข้อมูลที่เป็นตัวเลข เพราะสามารถคำนวณข้อมูลที่มีความซับซ้อนแบบขนานพร้อมกันได้ขณะเวลาเดียวกัน และมีการจดจำการเรียนรู้ที่รวดเร็ว

แนวคิดการทำงานของ Perceptron คือ คุณลักษณะข้อมูลที่นำเข้าโดยค่าในแต่ละคุณลักษณะที่นำเข้ามาจะถูกถ่วงน้ำหนักที่เกี่ยวข้อง ในตอนเริ่มต้นน้ำหนัก และค่า Bias จะถูกสุ่มค่าขึ้นมา โดยจะถูกปรับน้ำหนักตลอดการเรียนรู้ของข้อมูลที่ผิดพลาด เมื่อค่าคุณลักษณะถูกถ่วงน้ำหนักจะถูกคำนวณผลรวมน้ำหนักแต่ละค่าคุณลักษณะที่ถ่วงน้ำหนัก เพื่อให้โมเดลเรียนรู้กำหนดผลลัพธ์ของ Perceptron ผลลัพธ์ คือค่า 0 และ 1 สามารถเขียนรูปแบบสมการได้ดังนี้

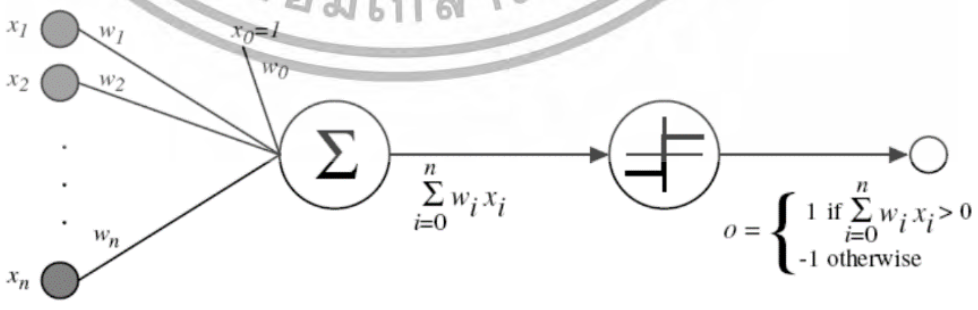
$$f(x) = \begin{cases} 1, & \text{if } \sum_{i=1}^{n} w_i \cdot x_i + b > 0 \\ 0, & \text{otherwise} \end{cases} \tag{2.9}$$

โดยที่ $w_i$ คือ ค่าน้ำหนักที่ i ต้องถ่วงน้ำหนักค่าคุณลักษณะ

$x_i$ คือ ค่าคุณลักษณะที่ i ในชุดข้อมูล

b คือ ค่าเอนเอียง (Bias) ที่สุ่มขึ้นมา

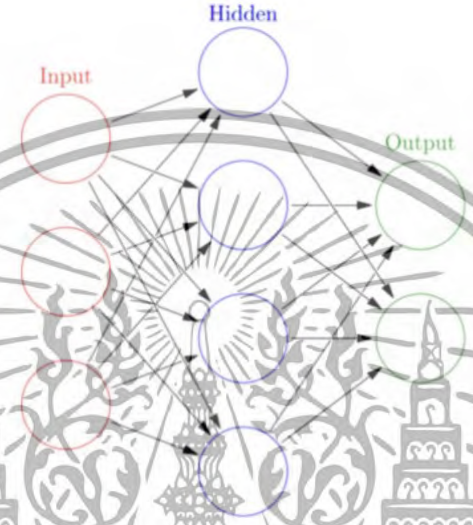
Perceptron สามารถเพิ่มประสิทธิภาพการเรียนรู้ของโมเดล Perceptron ได้โดยการนำผลรวมน้ำหนักแต่ละค่าคุณลักษณะ ส่งผ่านฟังก์ชันกระตุ้น เช่น ฟังก์ชัน sigmoid, ฟังก์ชัน ReLU หรือฟังก์ชัน Bipolar วิธีนี้เป็นการเพิ่ม Layer ในการคำนวณฟังก์ชัน f(x) เมื่อคำนวณค่า f(x) ได้จะถูกส่งไปคำนวณ f(x) ใน layer ถัดไปจนกว่าจะได้ผลการทำนายออกเป็น Binary ค่า 0 หรือ 1 เป็นกระบวนการเชื่อมต่อ Perceptron เรียกกระบวนการนี้ว่า Multilayer Perceptron




**รูปที่ 2.10 แนวคิดกระบวนการของ Perceptron ผ่านฟังก์ชันกระตุ้น**

รูปที่ 2.10 แสดงแนวคิดกระบวนการของ Perceptron ผ่านฟังก์ชันกระตุ้น โดยให้นำผลรวมน้ำหนักที่ถ่วงในแต่ละค่าจุดข้อมูลคุณลักษณะ ส่งผ่านไปฟังก์ชันกระตุ้นที่ Bipolar จนได้ค่าทำนายออกเป็น 2 ค่าคือ 1 หรือ -1

Multilayer Perceptron หรือเรียกอีกชื่อว่า Multilayer Neural Network ซึ่งประกอบด้วย Input layer ที่นำคุณลักษณะเข้ามาคำนวณ Hidden layer ที่ใช้ในการคำนวณพิจารณาคุณลักษณะที่นำเข้ามา และผลลัพธ์ที่ได้ Output layer



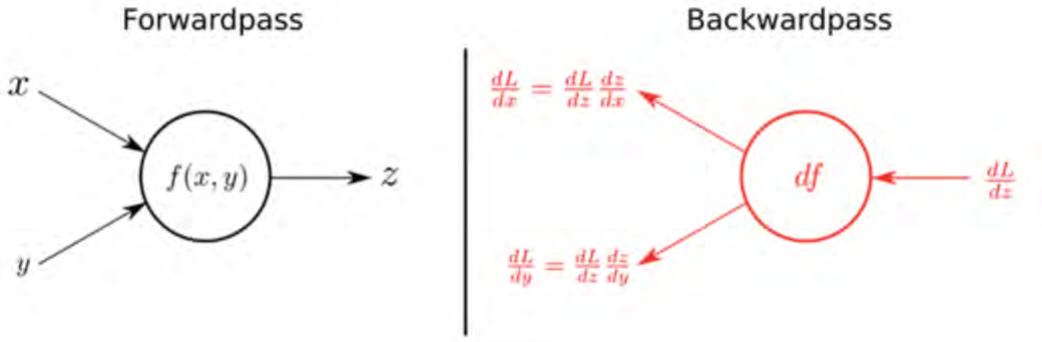

**รูปที่ 2.11 โมเดล Multilayer Perceptron**

ที่มา: https://bigdata.go.th/big-data-101/neural-network/

รูปที่ 2.11 เมื่อนำค่าคุณลักษณะเข้าในโมเดล Multilayer Perceptron จะถูกคำนวณผ่าน hidden layer ที่กำหนดจำนวน layer ตามความต้องการให้โมเดลมีการเรียนรู้ที่ซับซ้อน เพื่อให้โมเดลสามารถทำนายผลลัพธ์ที่ต้องการได้

### 2.12.1 อัลกอริทึม Backpropagation

โมเดล Artificial Neural Networks ที่มีโครงสร้างพื้นฐานมาจากโมเดล Multilayer Perceptron ในการสร้างโมเดล Artificial Neural Networks วิธีการฝึกสอนการเรียนรู้จากอัลกอริทึม Backpropagation เป็นอัลกอริทึมที่คำนวณค่าการทำนายที่ผิดพลาดในจากในชุดข้อมูลฝึกสอนไปปรับค่าน้ำหนักในแต่ละจุดข้อมูล เพื่อลดข้อผิดพลาดในการทำนายกับเซตคำตอบจริง




(ก) (ข)

**รูปที่ 2.12 ขั้นตอนการทำงานของอัลกอริทึม Backpropagation**

ที่มา: https://bdi.or.th/big-data-101/neural-network/

รูปที่ 2.12 แสดงขั้นตอนการทำงานของอัลกอริทึม Backpropagation รูป (ก) แสดงการนำข้อมูลเข้า Artificial Neural Networks รูป (ข) แสดงการส่งข้อมูลกลับไปตาม Artificial Neural Networks เพื่อปรับน้ำหนักให้โมเดลทำนายได้ถูกต้อง

ขั้นตอนการทำงานของอัลกอริทึม Backpropagation มีขั้นตอนดังนี้

1. Forward Pass ข้อมูลที่ถูกส่งเข้าไป Artificial Neural Networks ผ่านการกำหนดให้ค่าน้ำหนักแต่ละจุดข้อมูลส่งผ่านฟังก์ชันการกระตุ้นเพื่อให้ได้ผลลัพธ์การทำนาย
2. คำนวณข้อผิดพลาดจากผลลัพธ์การทำนายของ Artificial Neural Networks ถูกเปรียบเทียบกับค่าคำตอบจริง ความแตกต่างระหว่างผลลัพธ์ที่ทำนายได้กับคำตอบจริง คำนวณได้จากสมการ 2.10

$$\text{Error} = \frac{\sum (\text{Actual} - \text{Predicted})^2}{2} \qquad (2.10)$$

3. Backward Pass อัลกอริทึมทำงานย้อนกลับทางโครงข่าย เพื่อคำนวณค่าความผิดพลาดต่อน้ำหนักและปรับน้ำหนักในแต่ละจุดข้อมูลฝึกสอน น้ำหนักถูกปรับเพื่อลดค่าความผิดพลาด โดยการทำ Backward Pass ต้องการหาค่าน้ำหนักที่เหมาะสมที่สุดที่ลดความแตกต่างระหว่างผลลัพธ์ที่ทำนายได้กับค่าคำตอบจริง
4. ขั้นตอนที่ 1-4 จนกว่าโมเดล Artificial Neural Networks จบการฝึกสอนเรียนรู้และสามารถทำนายจำแนกประเภทในชุดข้อมูลที่ไม่เคยเห็นได้อย่างมีประสิทธิภาพ

## 2.13 การวัดประสิทธิภาพโมเดล

เทคนิคการวัดประสิทธิภาพโมเดล คือ เทคนิคที่ใช้ในการประเมินประสิทธิภาพและคุณภาพของโมเดลได้ตรงตามที่กำหนด ซึ่งจะยกตัวอย่างเทคนิคการประเมินประสิทธิภาพโมเดลดังนี้

### 2.13.1 ตาราง Confusion Matrix

ตารางที่ 2.1 ตาราง Confusion Matrix

| | **Actual Positive** | **Actual Negative** |
|---|---|---|
| **Predicted Positive** | True Positive (TP) | False Positive (FP) |
| **Predicted Negative** | False Negative (FN) | True Negative (TN) |

ตาราง confusion matrix เป็นตารางที่ใช้ในการอธิบายประสิทธิภาพโมเดลแบบจำแนกประเภท (Classification) การประเมินประสิทธิภาพโมเดล สามารถตรวจสอบได้จากตาราง โดยตารางกำหนดให้ ผลลัพธ์ที่ควรจะได้ (Actual) ด้านบนตาราง และด้านซ้ายมือถูกกำหนดให้ ผลลัพธ์การทำนาย (Predicted)

โดยที่ True Positive (TP) คือ จำนวนตัวอย่างที่โมเดลทำนายถูกต้องตรงกับคำตอบข้อมูลจริงในประเภท Positive class

True Negative (TN) คือ จำนวนตัวอย่างที่โมเดลทำนายถูกต้องตรงกับคำตอบข้อมูลจริงในประเภท Negative class

False Positive (FP) คือ จำนวนตัวอย่างที่โมเดลทำนายผิดจากคำตอบข้อมูลจริงในประเภท Positive class

False Negative (FN) คือ จำนวนตัวอย่างที่โมเดลทำนายผิดจากคำตอบข้อมูลจริงในประเภท Negative class

### 2.13.2 ค่า Accuracy

Accuracy เป็นการประเมินวัดความถูกต้องของโมเดล โดยพิจารณาทุกประเภทคลาส โดยสามารถคำนวณได้จากสมการ 2.11

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.11}$$

### 2.13.3 ค่า Precision

Precision เป็นการประเมินวัดความแม่นยำของโมเดลในการทำนายข้อมูลในประเภท Positive class โดยสามารถคำนวณได้จากสมการ 2.12

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2.12}$$

**2.13.4 ค่า Recall**

Recall เป็นการประเมินวัดความสามารถในการระบุประเภท Positive class ได้ถูกต้อง โดยคำนวณได้จากสมการ 2.13

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2.13}$$

**2.13.5 ค่า F1-Score**

F1-Score เป็นการหาค่าเฉลี่ยการประเมินวัดประสิทธิภาพระหว่างค่า Precision และค่า Recall ที่วัดความสามารถของโมเดลสามารถคำนวณได้จากสมการ 2.14

$$\text{F1 Score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.14}$$

## 2.14 งานวิจัยที่เกี่ยวข้อง

Yoga Pristyanto, Anggit Ferdita Nugraha, Rifda Faticha Alfa Aziza, Ibnu Hadi Purwanto, Mulia Sulistiyono และ Akhmad Dahlan (2023) ได้ศึกษาเกี่ยวกับความสามารถในการจัดการความไม่สมดุลของคลาสโดยการสร้างโมเดลโดยเลือกใช้อัลกอริทึมสามอัลกอริทึมที่แตกต่างกัน ได้แก่ XGBoost, Stacking และ Bagging แล้วพบว่าอัลกอริทึม XGBoost เป็นวิธีแก้ไขปัญหาที่มีประสิทธิภาพ จากการทดลองในชุดข้อมูลที่แตกต่างกันอัลกอริทึม XGBoost แสดงให้ถึงประสิทธิภาพที่ดีกว่าในทดลองปัญหาที่ไม่สมดุลแบบ Multiclass Classification

Shengnan Shi, Jie Li, Dan Zhu, Fang Yang และ Yong Xu (2023) ได้นำเสนองานวิจัยที่กล่าวถึงการจัดการข้อมูลที่ไม่สมดุลที่เกิดขึ้นในที่ใช้งานจริง โดยใช้วิธี hybrid คือ การผสมผสานวิธีการจัดการทั้งสองวิธีเข้าด้วยกัน ระหว่างการจัดการในระดับข้อมูล และระดับอัลกอริทึม แนวคิดเริ่มจากการจัดการในระดับข้อมูล (Data Level) การสุ่มตัวอย่างตามความหนาแน่นการกระจายตัวของข้อมูล ไปจนถึงการจัดการในการเลือกอัลกอริทึมที่เหมาะสมในการสุ่มตัวอย่าง ระดับอัลกอริทึม (Algorithm Level) โมเดลที่จะถูกสร้างขึ้นจะสร้างโมเดลตามการกระจายตัวของข้อมูลโดยใช้อัลกอริทึมที่แตกต่างกัน ชุดข้อมูลจะถูกเข้าไปฝึกสอนเรียนรู้ในแต่ละโมเดลจะได้โมเดลที่มีอัลกอริทึมที่เหมาะสมกับการกระจายตัวของชุดข้อมูลนั้น จึงเรียกโมเดลที่ใช้ทดลองนี้ว่า Hybrid Imbalanced Classification Model หรือเรียกว่า HICD โดยงานวิจัยนี้จะแสดงให้เห็นถึงประสิทธิภาพในการทดลองโดยใช้โมเดลแบบจำลอง HICD มีประสิทธิภาพที่ดีกว่าการจัดการโดยเน้นแต่ระดับอัลกอริทึมหรือระดับข้อมูล เพราะเนื่องด้วยข้อจำกัดของโมเดลที่มีอยู่ที่มีการนำเสนอการจำแนกคลาสข้อมูลไม่สมดุล แต่มักจะมองข้ามปัจจัยสำคัญ เช่น ความหนาแน่นของข้อมูลที่มีนัยสำคัญ ซึ่งอาจจะส่งผลต่อการประเมินวัดประสิทธิภาพโมเดลในการจำแนกประเภทคลาส

Rishabh Rustogi, Ayush Prasad (2019) ได้ศึกษาและวิจัยเกี่ยวกับการจัดการความไม่สมดุลของชุดข้อมูลที่ยังคงเป็นปัญหา จากการศึกษาพบเขาการจัดการกับข้อมูลที่ไม่สมดุล โดยการสุ่มตัวอย่างโดยใช้เทคนิค Undersampling ทำให้ชุดข้อมูลสมดุล และนอกจากนี้มีหลายเทคนิคมากที่จัดการกับปัญหาได้ดี แต่ในงานนี้วิจัยนี้เสนอการจัดการในรูปแบบ Hybrid สำหรับการจำแนกคลาสแบบ binary classification โดยการใช้เทคนิคสุ่มตัวอย่างในกลุ่มคลาสส่วนน้อยเป็นตัวอย่างสังเคราะห์ (SMOTE) และใช้อัลกอริทึม Extreme Learning Machine (ELM) โดยกำหนดให้มีอัตราการเรียนรู้ที่รวดเร็ว ซึ่งจะมีส่วนช่วยเพิ่มประสิทธิภาพในการเรียนรู้ของโมเดลได้ทำนายผลได้อย่างมีประสิทธิภาพตามที่ต้องการ โดยสรุปงานวิจัยนี้เน้นที่การกำหนดให้ค่าอัตราการเรียนรู้ที่รวดเร็วของโมเดล และวัดประสิทธิภาพจากทดลองโดยวัดจากการประเมินผล เช่น F-measure, G-mean และ ROC

Sima Mayabadi และ Hamid Saadatfar ได้ทำการศึกษาและวิจัยชุดข้อมูลที่ไม่สมดุล โดยงานวิจัยนี้มุ่งเน้นไปที่การจัดการข้อมูลที่พิจารณาจากความหนาแน่นของข้อมูลเพื่อกำจัดความทับซ้อนระหว่างคลาสสองคลาส และจัดการกระจายตัวข้อมูลในแต่ละคลาส โดยใช้เทคนิค Undersampling และ oversampling เพื่อลดจำนวนตัวอย่างจากคลาสส่วนใหญ่ โดยจะทดสอบกับชุดข้อมูลที่ไม่สมดุล 16 ชุดข้อมูลโดยใช้อัลกอริทึม Random Forest และ SVM ในการจำแนกประเภทข้อมูล โดยผลการทดลองได้ให้ข้อสรุปว่าโมเดลที่ใช้อัลกอริทึมสองอัลกอริทึมที่กล่าวมาในข้างต้นมีประสิทธิภาพมากกว่าอัลกอริทึมอื่น และโมเดลเหล่านี้ยังรักษาความสมดุลและโครงสร้างรูปร่างของข้อมูลคลาส

Xinmin Tao, Xinyue Guo, Yujia Zheng, Xiaohan Zhang และ Zhiyu Chen ทำการศึกษาและวิจัยการจัดการปัญหาชุดข้อมูลที่ไม่สมดุลที่มีปัญหาอย่างเพิ่มเข้ามาจากความไม่สมดุลของชุดข้อมูล เช่น การทับซ้อนของข้อมูลในแต่ละคลาส ค่าผิดปกติ เป็นต้น งานวิจัยนี้ได้นำเสนออัลกอริทึมใหม่ที่ปรับให้เข้ากับความซับซ้อนของข้อมูลคลาสกลุ่มส่วนน้อยเพื่อจัดการปัญหาการจำแนกที่ไม่สมดุล โดยใช้หลักการ Hyperspheres เป็นแนวคิดในการสร้างจุดข้อมูลสังเคราะห์สำหรับการสุ่มตัวอย่างที่มากเกินไปในชุดข้อมูลที่ไม่สมดุล เพื่อแก้ไขปัญหาที่ความไม่สมดุลของคลาสและการทับซ้อนกันในการจำแนกประเภทข้อมูล จากผลการทดลองในการวิจัยนี้พบว่าอัลกอริทึมที่นำเสนอมีประสิทธิเหนือกว่าวิธีการสุ่มตัวอย่างแบบ oversampling แบบอื่นในแง่ของประสิทธิภาพในการจำแนกประเภทคลาส

Behzad Mirzaei, Bahareh Nikpour และ Hossein Nezamabadi-pour (2020) ได้นำเสนองานวิจัยที่กล่าวถึงการใช้เทคนิคการสุ่มตัวอย่างใหม่โดยใช้อัลกอริทึม DSCAN. เพื่อเลือกตัวอย่างที่เหมาะสมในคลาสส่วนใหญ่ ในขณะเดียวกันก็จะใช้เทคนิคอื่นในการสร้างความสมดุลอีก 6 วิธีให้กับข้อมูลสำหรับการเรียนรู้ในโมเดล เพื่อเปรียบเทียบ และแสดงให้เห็นผ่านการทดลองว่าการใช้ DBSCAN มีประสิทธิภาพที่ดีกว่าอย่างไรในการจัดการกับชุดข้อมูลที่ไม่สมดุลเมื่อเปรียบเทียบกับวิธีอื่นที่ใช้ในการทดลอง

ตารางที่ 2.2 ตารางสรุปงานวิจัยที่เกี่ยวข้อง

| งานวิจัยที่เกี่ยวข้อง | อัลกอริทึมที่ใช้ | ผลการทดลอง | Classification | จำนวนชุดข้อมูล |
|---|---|---|---|---|
| Comparison of Ensemble Models as Solutions for Imbalanced Class Classification of Dataset | Algorithm Level Ensemble Model | อัลกอริทึม XGBoost แสดงให้ถึงประสิทธิภาพที่ดีกว่าในทดลองปัญหาชุดข้อมูลที่ไม่สมดุลแบบMulticlass Classification | Multiclass Classification | 5 |
| A hybrid imbalanced classification model based on data density | Data Level and Algorithm Level | HICD มีประสิทธิที่ดีกว่าการจัดการโดยเน้นแต่ระดับอัลกอริทึม หรือระดับข้อมูล | Binary Classification | 18 |
| Swift Imbalance Data Classification using SMOTE and Extreme Learning Machin | Data Level and Algorithm Level | โมเดลมีประสิทธิภาพการเรียนรู้ได้ดียิ่งขึ้นจากการใช้เทคนิค SMOTE กับอัลกอริทึม ELM ในการเรียนรู้ของโมเดล | Binary Classification | 5 |
| Two density-based sampling approaches for imbalanced and overlapping data | Data Level | อัลกอริทึม Random Forest และ SVM ในการจำแนกประเภทข้อมูลมีประสิทธิภาพที่ดี และรักษาความสมดุลและโครงสร้างรูปร่างของข้อมูลคลาส | Binary Classification | 16 |
| Self-adaptive oversampling method based on the complexity of minority data in imbalanced datasets classification | Data Level | อัลกอริทึมที่นำเสนอมีประสิทธิภาพที่เหนือกว่าวิธีการสุ่มตัวอย่างแบบ oversampling สุ่มตัวอย่างแบบเพิ่มขนาดข้อมูลในแง่ของการวัดประเมินประสิทธิภาพในการจำแนกประเภทคลาส | Binary Classification | 28 |
| An under-sampling technique for imbalanced data classification based on DBSCAN algorithm | Data Level | อัลกอริทึม DBSCAN มีประสิทธิภาพที่ดีกว่าอย่างไรในการจัดการกับชุดข้อมูลที่ไม่สมดุล เมื่อเทียบกับ SMOTE-Tomek, RUS และเทคนิคอื่น | Binary Classification | 15 |

# บทที่ 3
# วิธีการดำเนินงานวิจัย

ในบทนี้จะอธิบายขั้นตอนการดำเนินงาน โดยใช้เทคนิคดังต่อไปนี้ ในการจัดการชุดข้อมูลที่ไม่สมดุลกัน เทคนิคเพิ่มจำนวนกลุ่มตัวอย่างคลาสส่วนน้อย โดยใช้เทคนิคการสุ่มตัวอย่างแบบ SMOTE เทคนิคการสุ่มตัวอย่างแบบลดขนาดกลุ่มคลาสส่วนใหญ่อย่างเทคนิค Tomek Links และเทคนิคการเพิ่มประสิทธิภาพโมเดลแบบ Boosting ที่ใช้การผสมผสานระหว่างเทคนิคการสุ่มตัวอย่างแบบ Random Under Sampling และเทคนิค AdaBoost โดยการเรียนรู้ของโมเดลจะถูกฝึกสอนการเรียนรู้ด้วยอัลกอริทึม Decision Trees, Naïve Bayes, k-Nearest Neighbors, Support Vector Machines และ Artificial Neural Networks โดยชุดข้อมูลที่จะนำมาใช้ในการทดสอบประสิทธิภาพโมเดลจะถูกแบ่งชุดข้อมูลแบบ Cross Validation โดยใช้วิธีการสุ่มแบ่งข้อมูลเป็นสัดส่วนตามจำนวนค่าคงที่ K (K-Fold Cross Validation) เพื่อให้ได้ชุดข้อมูลที่มีการกระจายตัวของข้อมูล สำหรับชุดข้อมูลฝึกสอน และส่วนหนึ่งจะถูกแบ่งออกเป็นชุดข้อมูลสำหรับทดสอบและจะทำวนซ้ำจนครบทุกส่วน และเพื่อเปรียบเทียบประสิทธิภาพโมเดลทำนายการจำแนกประเภทบนชุดข้อมูลที่ไม่สมดุลจำนวน 10 ชุดข้อมูลที่มีการใช้งานจริงจากเว็บไซต์ Kaggle โดยดำเนินการผ่านการเรียกใช้ Library จากภาษาโปรแกรม Python ในการสร้างโมเดลสำหรับทดสอบและประเมินวัดประสิทธิภาพเทคนิคที่ใช้แตกต่างกันในแต่ละอัลกอริทึมการเรียนรู้ของเครื่อง

## 3.1 ชุดข้อมูลที่ไม่สมดุล

ชุดข้อมูลที่ไม่สมดุลที่ใช้ในการดำเนินงานวิจัยใช้ชุดข้อมูลทั้งหมดเป็นจำนวน 10 ชุดข้อมูล โดยทุกชุดข้อมูลที่นำมาใช้ในการดำเนินการทดลองเป็นชุดข้อมูลที่จำแนกประเภทคลาสออกเป็น Binary classification จะแสดงชุดข้อมูลที่ไม่สมดุลออกเป็นตารางข้อมูล โดยจะมีชื่อของชุดข้อมูลที่ไม่สมดุล จำนวนข้อมูล Majority class หรือ Negative class จำนวนข้อมูล Minority class หรือ Positive class จำนวนกลุ่มตัวอย่างคลาสทั้งหมด จำนวนคุณลักษณะ หรือปัจจัยในการจำแนกคลาส อัตราส่วน Imbalanced Ratio ระหว่างทั้ง 2 คลาส ประเภทคุณสมบัติ หรือปัจจัยข้อมูล และประเภทหมวดหมู่ชุดข้อมูล แสดงดังตารางต่อไปนี้

ตารางที่ 3.1 ตารางชุดข้อมูลที่ไม่สมดุล

| NO. | Dataset | Positive | Negative | Instance | Attributes | IR | Attribute types | Domain |
|---|---|---|---|---|---|---|---|---|
| 1 | Stroke Prediction Dataset | 249 | 4,861 | 5,110 | 11 | 19.52 | Nominal, Numeric, Binary | medical |
| 2 | COVID - 19 Dataset | 8981 | 116,171 | 125,152 | 21 | 12.94 | Nominal, Numeric | medical |
| 3 | Diabetes prediction dataset | 8,500 | 91,500 | 100,000 | 811 | 10.76 | Nominal, Numeric, Binary | medical |
| 4 | Water Quality | 912 | 7,084 | 7,999 | 20 | 7.77 | Numeric | quality |
| 5 | Credit Card Fraud | 87,403 | 912,597 | 100,000 | 7 | 10.44 | Nominal, Numeric | fraud |
| 6 | Bank Marketing Dataset | 5,289 | 39,922 | 45,211 | 16 | 7.55 | Nominal, Numeric | marketing |
| 7 | Heart Disease Dataset | 10,332 | 109,463 | 119,795 | 17 | 10.59 | Nominal, Numeric | medical |
| 8 | Lumpy Skin Disease | 3,039 | 21,764 | 24,803 | 19 | 7.16 | Numeric | medical |
| 9 | Microcalcification classification | 260 | 10,923 | 11,183 | 5 | 42.01 | Numeric | medical |
| 10 | Bank Marketing task | 521 | 4,000 | 4,521 | 16 | 7.68 | Nominal, Numeric | marketing |

## 3.2 ขั้นตอนการคัดเลือกคุณลักษณะ

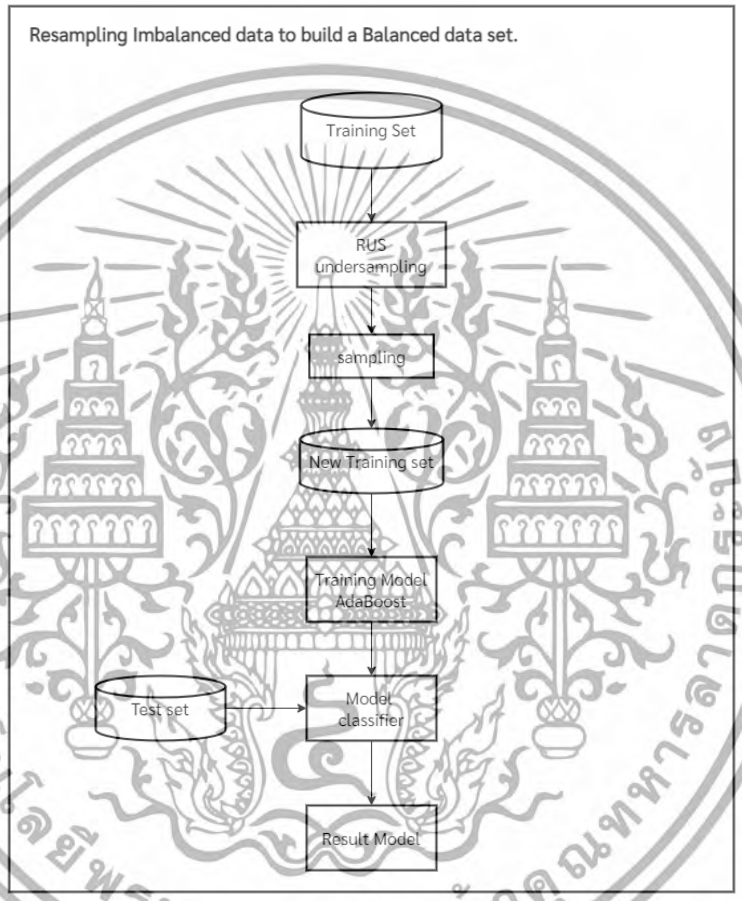
การคัดเลือกคุณลักษณะ คือ ขั้นตอนที่จะทำการคัดเลือกคุณลักษณะที่เกี่ยวข้องและสำคัญที่สุดผ่านวิธีการคัดเลือกแบบ ANOVA ที่พิจารณาจากค่า F-statistic และ p-value โดยจะคัดเลือกคุณลักษณะจากเปอร์เซ็นต์คุณลักษณะที่เรียงลำดับความเกี่ยวข้องนัยสำคัญที่สุดเป็นเปอร์เซ็นต์ ดังนี้ 10 เปอร์เซ็นต์ 25 เปอร์เซ็นต์ 50 เปอร์เซ็นต์ และ 75 เปอร์เซ็นต์ ในการเปรียบเทียบจำนวนคุณลักษณะที่มีผลต่อจำแนกประเภทของโมเดล โดยทำการคัดเลือกหลังจากที่จัดการคุณลักษณะที่มีข้อมูลเป็น null หรือข้อมูลที่หายไปออกจากชุดข้อมูล

ตารางที่ 3.2 ตารางการคัดเลือกคุณลักษณะ

| no. | dataset | Original features | After remove missing value | Features Selection | | | |
|---|---|---|---|---|---|---|---|
| | | | | top 10% | top 25% | top 50% | top 75% |
| 1 | Stroke Prediction | 11 | 11 | 2 | 3 | 6 | 8 |
| 2 | COVID - 19 Dataset | 21 | 17 | 2 | 4 | 8 | 12 |
| 3 | Diabetes Prediction | 8 | 8 | 1 | 2 | 4 | 6 |
| 4 | Water Quality | 20 | 16 | 2 | 4 | 8 | 12 |
| 5 | Credit Card Fraud | 7 | 7 | 1 | 2 | 4 | 5 |
| 6 | Bank Marketing | 16 | 16 | 2 | 4 | 8 | 12 |
| 7 | Heart Disease | 17 | 17 | 2 | 5 | 9 | 13 |
| 8 | Lumpy Skin Disease | 19 | 16 | 2 | 4 | 8 | 12 |
| 9 | Microcalcification classification | 5 | 5 | 1 | 2 | 3 | 4 |
| 10 | Bank Marketing Task | 16 | 16 | 2 | 4 | 8 | 12 |

## 3.3 ขั้นตอนวิธีการ RUSBoostClassifier

RUSBoostClassifier คือ ขั้นตอนวิธีการหรือเทคนิคอย่างหนึ่งในการเพิ่มประสิทธิโมเดลแบบ Boosting โดยการนำเทคนิคการสุ่มตัวอย่างแบบ Random Under sampling ซึ่งเป็นหนึ่งในวิธีการแก้ปัญหาชุดข้อมูลไม่สมดุล ผสมผสานกับเทคนิค AdaBoost เป็นเทคนิควิธีการเรียนรู้ของเครื่องสำหรับงานการทำนายจำแนกประเภทคลาส เพื่อเพิ่มความแม่นยำ โดยการรวบรวมโมเดลที่มีการทำนายประสิทธิภาพต่ำ เพื่อสร้างโมเดลที่มีความสามารถการทำนายประสิทธิภาพสูง




**รูปที่ 3.1 แผนภาพแสดงการงานของโมเดลที่ใช้อัลกอริทึม RUSBoostClassifier**

รูปที่ 3.1 แผนภาพแสดงการงานของโมเดลที่ใช้อัลกอริทึม RUSBoostClassifier โดยนำข้อมูลที่ไม่สมดุลสำหรับฝึกสอนโมเดล เข้าสู่การทำชุดข้อมูลให้สมดุลผ่านเทคนิคการสุ่มตัวอย่างแบบ Random Under sampling หลังจากที่ชุดข้อมูลสมดุลกัน ชุดข้อมูลจะถูกนำไปสู่การฝึกสอนการเรียนรู้ โดยการเรียนรู้ของโมเดลจะถูกฝึกสอนการเรียนรู้แบบ Boosting คือการถ่วงน้ำหนักชุดข้อมูล เมื่อโมเดลทำนายผิดจุดข้อมูลในชุดข้อมูลจะถูกถ่วงน้ำหนักที่เพิ่มมากขึ้นจนกว่าโมเดลจะทำนายถูกหรือมีประสิทธิภาพตามที่ต้องการ

**ขั้นตอนที่ 1:** เตรียมชุดข้อมูลที่ไม่สมดุลจากชุดข้อมูลที่มีการใช้งานจริงจาก Kaggle จำนวนชุดความข้อมูลที่ไม่สมดุล 10 ชุดข้อมูล ดำเนินการแบ่งชุดข้อมูลสำหรับฝึกสอนและทดสอบโมเดล

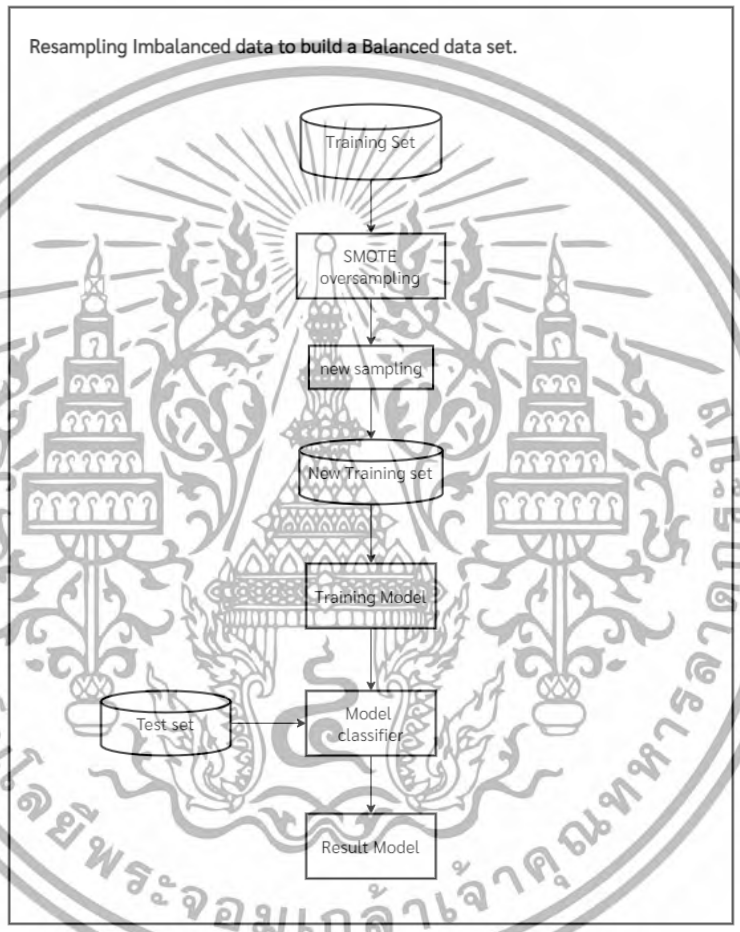
**ขั้นตอนที่ 2:** ชุดข้อมูลที่ไม่สมดุลสำหรับฝึกสอนโมเดลที่ผ่านการคัดเลือกคุณลักษณะเมื่อเข้าสู่กระบวนการในอัลกอริทึม RUSBoostClassifier โดยเรียกใช้ Imbalanced Learn Library จากโมดูล imblearn.ensemble โดยเทคนิค RUSBoostClassifier จะทำการสุ่มตัวอย่างสำหรับแบบ RUS เป็นขั้นตอนที่ลดจำนวนกลุ่มตัวอย่าง Majority class เพื่อให้มีจำนวนเท่ากับจำนวนกลุ่มตัวอย่าง Minority class เพื่อสร้างความสมดุลในชุดข้อมูลก่อนที่จะนำไปสู่การเรียนรู้ฝึกสอนให้กับโมเดล

**ขั้นตอนที่ 3:** เมื่อชุดข้อมูลสมดุลกันชุดข้อมูลจะถูกนำไปใช้ฝึกสอนในโมเดล อัลกอริทึมที่ใช้ในการจำแนกประเภทและการทำนายถูกเรียกใช้จาก scikit-learn Library ในภาษา Python โดยใช้อัลกอริทึมที่ใช้ความน่าจะเป็นในการจำแนก ได้แก่ Decision Trees และ Naïve Bayes ที่กำหนดค่าพารามิเตอร์เป็นค่าเริ่มต้น และอัลกอริทึมที่ใช้การคำนวณเชิงคณิตศาสตร์ ได้แก่ Support vector Machines ที่กำหนดค่าพารามิเตอร์ kernel เป็น linear ที่ใช้ในการคำนวณเป็นค่าเริ่มต้น k-Nearest Neighbors ที่กำหนดค่าพารามิเตอร์จำนวนเพื่อนบ้านที่ใกล้เคียงเท่ากับ 5 และ Artificial Neural Networks จะถูกสร้างโมเดลจากการเรียกใช้ Keras Library โดยกำหนดค่า hyperparameter ดังนี้ activation ใน input layer และ hidden layer คือ ReLu ส่วน output layer กำหนด activation เป็น sigmoid ในการเรียนรู้สำหรับ Binary Classification กำหนดค่า loss ใช้เป็นฟังก์ชัน binary cross entropy จากโมดูล losses เพื่อให้โมเดลสามารถจำแนกได้อย่างแม่นยำเพิ่มขึ้น และกำหนดให้ optimizer เป็น Adam เพื่อเพิ่มประสิทธิภาพการทำงานของโมเดล โดยในโปรแกรม Python จะนำอัลกอริทึมเหล่านี้ไปกำหนดพารามิเตอร์เป็นอัลกอริทึมที่ใช้ในการเรียนรู้ของเครื่องเมื่อผ่านเทคนิค RUSBoostClassifier โมเดลจะถูกฝึกสอนให้มีการเรียนรู้ AdaBoost เทคนิคนี้จะกำหนดค่าน้ำหนักให้กับทุกจุดชุดข้อมูลมีน้ำหนักเท่ากัน ในการฝึกสอนให้โมเดลที่มีการเรียนรู้ในประสิทธิภาพที่ต่ำรวบรวมโมเดลการทำนายประสิทธิภาพต่ำ เพื่อสร้างโมเดลที่มีการเรียนรู้และการทำนายในประสิทธิภาพที่สูง

**ขั้นตอนที่ 4:** กระบวนการในขั้นตอนนี้ คือการทดสอบประสิทธิภาพของโมเดล โดยจะวัดประสิทธิภาพโมเดลด้วยวิธี Cross Validation แบ่งข้อมูลเป็นจำนวน 10 ส่วนเท่ากัน 10-Fold Cross Validation โดยจะใช้ข้อมูลส่วนหนึ่งสำหรับทดสอบประสิทธิภาพและส่วนที่เหลือในการเรียนรู้ฝึกสอนเพื่อสร้างโมเดล ตัวอย่างเช่น ข้อมูลตั้งแต่ส่วนที่ 2-10 จะถูกนำมาฝึกสอนและสร้างโมเดลขึ้นมา และข้อมูลส่วนที่ 1 จะถูกนำมาใช้สำหรับทดสอบประสิทธิภาพโมเดล และจะทำวนซ้ำจนกว่าครบทุกส่วน หรือกล่าวได้ว่าจะทดสอบประสิทธิภาพโมเดลตามรอบจำนวนสัดส่วนที่แบ่งข้อมูลค่า K หากมีเปอร์เซ็นต์ Recall จะถือว่าโมเดลมีประสิทธิภาพการทำนายและการเรียนรู้ที่สูง และเปรียบเทียบจำนวนคุณลักษณะที่ส่งผลต่อการจำแนกประเภท

## 3.4 ขั้นตอนวิธีการ SMOTE

SMOTE (Synthetic Minority Over-sampling Technique) คือ ขั้นตอนวิธีการอย่างหนึ่งที่นิยมที่ใช้ในการจัดการปัญหาชุดข้อมูลที่ไม่สมดุล โดยวิธีการของ SMOTE จะแก้ปัญหาให้ชุดข้อมูลสมดุลกันได้โดยการสร้างตัวอย่างสังเคราะห์ใหม่จะถูกเพิ่มเข้าไปในชุดข้อมูลสำหรับฝึกสอนการเรียนรู้ของโมเดล เพื่อไม่ให้โมเดลทำนายผลโอนเอียงไปในทางคลาสใดคลาสหนึ่ง และเพิ่มความแม่นยำในทำนายในคลาสส่วนน้อยได้อย่างมีประสิทธิภาพ




**รูปที่ 3.2 แผนภาพแสดงการทำงานของโมเดลที่ใช้อัลกอริทึม SMOTE**

รูปที่ 3.2 แผนภาพแสดงการทำงานของโมเดลที่ใช้อัลกอริทึม SMOTE โดยการนำชุดข้อมูลที่ไม่สมดุลนำเข้าสู่กระบวนการจัดการความสมดุลข้อมูลโดยผ่านขั้นตอนวิธีการ SMOTE เพื่อสร้างตัวอย่างสังเคราะห์ใหม่เพิ่มเข้าไปในชุดข้อมูล เพื่อให้ชุดข้อมูลมีความสมดุลกันก่อนที่จะนำมาฝึกสอนในโมเดล

**ขั้นตอนที่ 1:** ชุดข้อมูลที่ไม่สมดุลที่ถูกเตรียมไว้สำหรับฝึกสอนโมเดลจากชุดข้อมูล 10 ชุด ข้อมูลที่ผ่านการคัดเลือกคุณลักษณะของข้อมูลแต่ละเปอร์เซ็นต์ที่เกี่ยวข้องมากที่สุด

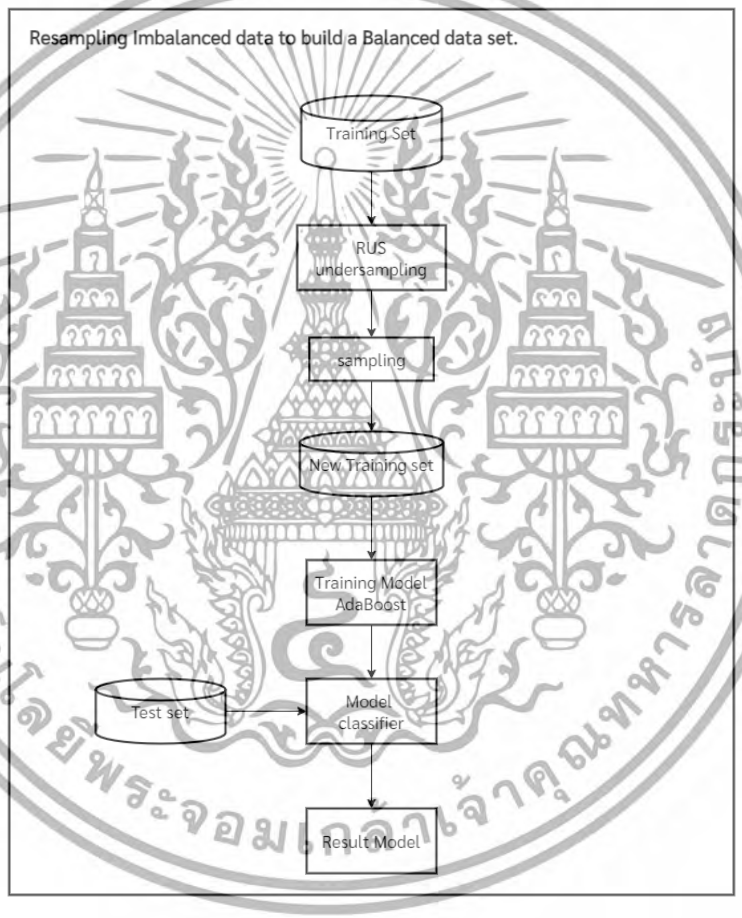
**ขั้นตอนที่ 2:** อัลกอริทึมการสุ่มตัวอย่างสร้างตัวอย่างสังเคราะห์ SMOTE โดยเรียกใช้จาก Imbalanced Learn Library โมดูลที่ชื่อว่า oversampling กำหนดพารามิเตอร์ของอัลกอริทึมนี้เป็นค่าเริ่มต้น โดยเทคนิคนี้จะทำการสุ่มจุดข้อมูลเริ่มต้นในการหาระยะห่างระหว่างจุดข้อมูลในคลาสส่วนน้อย โดยใช้วิธีการหาระยะห่างแบบ Euclidean ในการคำนวณหาจุดข้อมูลเพื่อนบ้านที่อยู่ใกล้เคียงมากที่สุดกับจุดข้อมูลเริ่มต้นที่ทำการสุ่มขึ้นมาในตอนต้น ตัวอย่างใหม่ที่ถูกสังเคราะห์ขึ้นมาจะถูกสร้างจากการสุ่มค่า 0-1 และคูณเข้ากับระยะห่างจุดข้อมูลเริ่มต้นกับจุดข้อมูลเพื่อนบ้านที่ใกล้เคียง เมื่อตัวอย่างใหม่สังเคราะห์ถูกเพิ่มเข้าจนจำนวนข้อมูลกลุ่มตัวอย่างคลาสส่วนน้อยมีขนาดเท่ากันกับจำนวนกลุ่มตัวอย่างคลาสส่วนใหญ่ หรือกล่าวได้ว่าตัวอย่างสังเคราะห์ใหม่จะถูกเพิ่มเข้าไปในชุดข้อมูลจนกว่าชุดข้อมูลจะสมดุลกัน

**ขั้นตอนที่ 3:** เมื่อชุดข้อมูลสมดุลกันผ่านเทคนิคการสุ่มตัวอย่างแบบการสร้างตัวอย่างสังเคราะห์ใหม่ SMOTE ชุดข้อมูลจะถูกนำมาฝึกสอนการเรียนรู้ให้กับเครื่อง (Machine Learning) โมเดลจะถูกฝึกสอนให้มีการเรียนรู้และจำแนกประเภทข้อมูล โดยพื้นฐานหลักในการฝึกสอนการจำแนกประเภท โมเดลจะถูกฝึกสอนจากการเรียนรู้ด้วยอัลกอริทึมที่ใช้จะถูกเรียกจาก scikit-learn Library ในภาษา Python โดยใช้อัลกอริทึมที่ใช้ความน่าจะเป็นในการจำแนก ได้แก่ Decision Trees และ Naïve Bayes ที่กำหนดค่าพารามิเตอร์เป็นค่าเริ่มต้น และอัลกอริทึมที่ใช้การคำนวณเชิงคณิตศาสตร์ ได้แก่ Support vector Machines ที่กำหนดค่าพารามิเตอร์ kernel เป็น linear ที่ใช้ในการคำนวณเป็นค่าเริ่มต้น k-Nearest Neighbors ที่กำหนดค่าพารามิเตอร์จำนวนเพื่อนบ้านที่ใกล้เคียงเท่ากับ 5 และ Artificial Neural Networks จะถูกสร้างโมเดลจากการเรียกใช้ Keras Library โดยกำหนดค่า hyperparameter ดังนี้ activation ใน input layer และ hidden layer คือ ReLu ส่วน output layer กำหนด activation เป็น sigmoid ในการเรียนรู้สำหรับ Binary Classification กำหนดค่า loss ใช้เป็นฟังก์ชัน binary cross entropy จากโมดูล losses เพื่อให้โมเดลสามารถจำแนกได้อย่างแม่นยำเพิ่มขึ้น และกำหนดให้ optimizer เป็น Adam เพื่อเพิ่มประสิทธิภาพการทำงานของโมเดล โมเดลเหล่านี้จะถูกนำไปเข้าสู่ขั้นตอนถัดไป คือการทดสอบประสิทธิภาพของโมเดล

**ขั้นตอนที่ 4:** ชุดข้อมูลที่ไม่สมดุลสำหรับการทดสอบประสิทธิภาพโมเดล Cross Validation แบบ K-Fold Cross Validation ในการนำมาทดสอบประสิทธิภาพโมเดล แบ่งชุดข้อมูลออกเป็น 10 ส่วน การทำงานของ K-Fold จะแบ่งส่วนหนึ่งไว้สำหรับในการทดสอบโมเดล และส่วนที่เหลือจะถูกนำไปฝึกสอนให้กับโมเดล เมื่อได้ผลการทดสอบประสิทธิภาพโมเดลจะพิจารณาไปเปอร์เซ็นต์ Recall คือความสามารถในการระบุ Positive class ได้อย่างถูกต้อง และเปรียบเทียบจำนวนคุณลักษณะที่ส่งผลต่อการจำแนกประเภทข้อมูล ในกรณีที่มีเปอร์เซ็นต์ที่สูงกว่าโมเดลอื่นจะถือว่าโมเดลนั้นมีประสิทธิภาพที่ดีเหมาะสมกับชุดข้อมูลที่นำมาทดสอบ

## 3.5 ขั้นตอนวิธีการ Tomek Links

Tomek Links คือขั้นตอนวิธีการที่ใช้ในการแก้ไขปัญหาชุดข้อมูลที่ไม่สมดุล โดยวิธีการแก้ปัญหาของ Tomek Links แก้ปัญหาโดยการจับคู่คลาสที่อยู่ตรงข้ามกันภายใต้เงื่อนไขที่จุดข้อมูลทั้ง 2 จุดข้อมูลมีระยะห่างเป็นเพื่อนบ้านที่ใกล้เคียงกันมากที่สุด จึงสามารถสร้างคู่ Tomek Links ได้ สร้างความสมดุลของชุดข้อมูลจากการลบจุดข้อมูลคลาสส่วนใหญ่ในคู่ Tomek Links เพื่อเหลือจุดข้อมูลเพียงจุดเดียว เพื่อให้จำนวนคลาสส่วนใหญ่และจำนวนคลาสส่วนน้อยมีขนาดเท่ากันและชุดข้อมูลสมดุลก่อนที่จะนำเข้าสู่กระบวนการฝึกสอนโมเดล การใช้เทคนิคการสุ่มตัวอย่างแบบการลบจำนวนกลุ่มตัวอย่าง Tomek Links ช่วยปรับปรุงการจำแนกประเภทคลาสให้มีประสิทธิภาพที่ดีขึ้น




**รูปที่ 3.3 แผนภาพแสดงการทำงานของโมเดลที่ใช้อัลกอริทึม Tomek Links**

รูปที่ 3.3 แผนภาพแสดงการทำงานของโมเดลที่ใช้อัลกอริทึม Tomek Links โดยการนำชุดข้อมูลที่ไม่สมดุลถูกจัดการความสมดุลข้อมูลโดยผ่านขั้นตอนวิธีการจับคู่จุดข้อมูลคลาสตรงข้าม โดยการลบจุดข้อมูลคลาสส่วนส่วนใหญ่ที่จับคู่ Tomek Links เป็นการลดจำนวนกลุ่มตัวอย่างในแต่ละคลาสให้มีความสมดุล โดยวิธีการของ Tomek Links นอกจากจะทำให้ชุดข้อมูลสมดุลกันจะช่วยปรับปรุงชุดข้อมูลให้โมเดลสามารถจำแนกประเภทคลาสได้อย่างประสิทธิภาพ

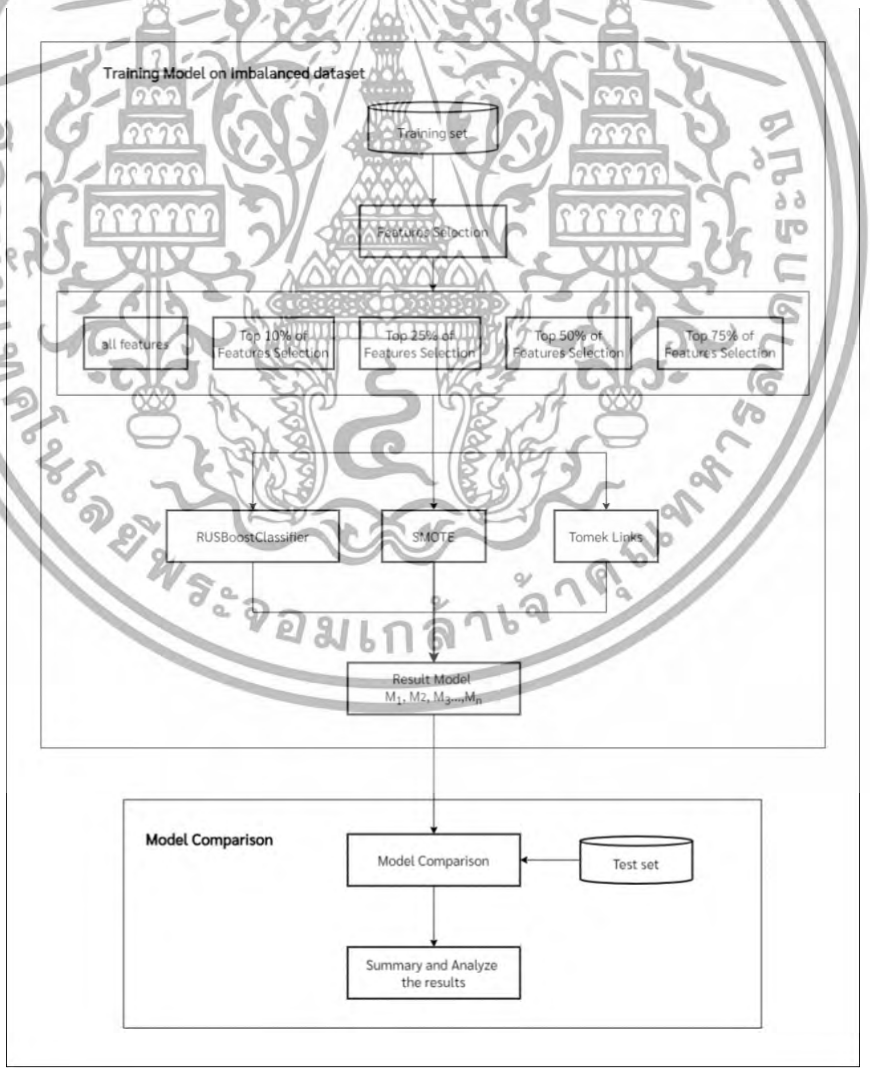
**ขั้นตอนที่ 1:** ชุดข้อมูลที่ไม่สมดุลที่ถูกเตรียมไว้สำหรับฝึกสอนโมเดลจากชุดข้อมูล 10 ชุดที่มีการคัดเลือกคุณลักษณะในแต่ละเปอร์เซ็นต์ที่เกี่ยวข้องนัยสำคัญที่สุดถูกจัดการผ่านเทคนิคการสุ่มตัวอย่างโดยใช้วิธีการ Tomek Links ในภาษา Python จะผ่านการเรียกใช้จาก Imbalanced Learn Library โมดูลที่ชื่อว่า undersampling และกำหนดค่าพารามิเตอร์เป็นค่าเริ่มต้น เมื่อชุดข้อมูลที่ไม่สมดุลเข้าสู่กระบวนการ Tomek Links จุดข้อมูลระหว่างกลุ่มตัวอย่างคลาสส่วนใหญ่และกลุ่มตัวอย่างคลาส่วนน้อยจะจับคู่หาระยะห่างเพื่อนบ้านที่ใกล้เคียงมากที่สุด โดยใช้วิธีการคำนวณหาระยะห่างแบบ Euclidean เมื่อได้จุดข้อมูลที่เป็นเพื่อนบ้านที่ใกล้เคียงของคลาสที่ตรงข้ามกัน จึงจะถือว่าเป็นการจับคู่ของ Tomek Links จุดข้อมูลที่จับคู่จะถูกลบจุดข้อมูลให้เหลือเพียงจุดข้อมูลเดียวภายในคู่ การลบจุดข้อมูลถูกตัดสินใจลบ หรือเก็บตัวอย่างไว้จากขอบเขตการตัดสินใจที่กำหนดไว้ โดยจะกำหนดให้ลบจุดข้อมูลที่มาจากกลุ่มตัวอย่างคลาสส่วนใหญ่จนกว่าจำนวนตัวอย่างกลุ่มคลาสส่วนใหญ่มีขนาดเท่ากับจำนวนตัวอย่างกลุ่มคลาสส่วนน้อย และทำวนซ้ำจนกว่าจะไม่มีคู่ Tomek Links เหลืออยู่ในชุดข้อมูล จะได้ชุดข้อมูลสำหรับฝึกสอนใหม่ที่มีความสมดุลผ่านกระบวนการ Tomek Links

**ขั้นตอนที่ 2:** เมื่อได้ชุดข้อมูลที่สมดุล ชุดข้อมูลจะถูกนำมาฝึกสอนการเรียนรู้โมเดล ผ่านอัลกอริทึมการเรียนรู้ของเครื่อง โดยเรียกใช้อัลกอริทึมจาก scikit-learn Library ในภาษา Python โดยใช้อัลกอริทึมที่ใช้ความน่าจะเป็นในการจำแนก ได้แก่ Decision Trees และ Naïve Bayes ที่กำหนดค่าพารามิเตอร์เป็นค่าเริ่มต้น และอัลกอริทึมที่ใช้การคำนวณเชิงคณิตศาสตร์ ได้แก่ Support vector Machines ที่กำหนดค่าพารามิเตอร์ kernel เป็น linear ที่ใช้ในการคำนวณเป็นค่าเริ่มต้น k-Nearest Neighbors ที่กำหนดค่าพารามิเตอร์จำนวนเพื่อนบ้านที่ใกล้เคียงเท่ากับ 5 และ Artificial Neural Networks จะถูกสร้างโมเดลจากการเรียกใช้ Keras Library โดยกำหนดค่า hyperparameter ดังนี้ activation ใน input layer และ hidden layer คือ ReLu ส่วน output layer กำหนด activation เป็น sigmoid ในการเรียนรู้สำหรับ Binary Classification กำหนดค่า loss ใช้เป็นฟังก์ชัน binary cross entropy จากโมดูล losses เพื่อให้โมเดลสามารถจำแนกได้อย่างแม่นยำเพิ่มขึ้น และกำหนดให้ optimizer เป็น Adam เพื่อเพิ่มประสิทธิภาพการทำงานของโมเดล โมเดลที่ได้รับการฝึกสอนแล้วจะถูกทดสอบโดยใช้ชุดข้อมูลที่ไม่สมดุลในการทดสอบให้โมเดลจำแนกประเภท โดยจะประเมินวัดประสิทธิภาพโมเดลเพื่อให้ได้ประสิทธิภาพในระดับที่ต้องการ หรือว่ามีประสิทธิภาพที่สูง

**ขั้นตอนที่ 3:** ขั้นตอนการวัดประสิทธิภาพโมเดล ใช้วิธีการทดสอบวัดประสิทธิภาพโมเดลแบบ K-Fold Cross Validation โดยจะแบ่งข้อมูลออกเป็น 10 ส่วน ในการทดสอบจะแบ่งข้อมูลส่วนหนึ่งสำหรับทดสอบโมเดล และส่วนที่เหลือนำมาฝึกสอนการเรียนรู้ให้กับโมเดล และเปรียบเทียบจำนวนคุณลักษณะที่ส่งผลต่อการจำแนกประเภทข้อมูลจากเปอร์เซ็นต์ความสามารถในการระบุ Positive class ได้อย่างถูกต้อง หรือค่า Recall มีเปอร์เซ็นต์ที่สูงจึงจะสรุปผลได้ว่าโมเดลมีประสิทธิภาพเหมาะสมในการจัดการชุดข้อมูลที่ไม่สมดุล

## 3.6 ขั้นตอนวิธีการ Model Comparison

Model Comparison คือขั้นตอนการเปรียบเทียบโมเดลบนชุดข้อมูลที่ไม่สมดุล 10 ชุดข้อมูลที่มีการใช้งานจริงจากเว็บไซต์ Kaggle ผ่านการคัดเลือกคุณลักษณะที่เกี่ยวข้องที่สุดเป็นเปอร์เซ็นต์ความเกี่ยวข้อง ดังนี้ 10 เปอร์เซ็นต์ 25 เปอร์เซ็นต์ 50 เปอร์เซ็นต์ และ 75 เปอร์เซ็นต์ เพื่อทำการทดลองว่าจำนวนคุณลักษณะมีผลต่อการจำแนกประเภทหรือไม่ และชุดข้อมูลแบบใดเหมาะสมโมเดลที่ใช้เทคนิค 3 เทคนิคที่กล่าวมาข้างต้น ได้แก่ เทคนิคการเพิ่มประสิทธิภาพโมเดลแบบ Boosting โดยใช้อัลกอริทึม RUSBoostClassifier เทคนิคการสุ่มตัวอย่างโดยการเพิ่มจำนวนตัวอย่างแบบเทคนิค SMOTE และเทคนิคการสุ่มตัวอย่างการลดจำนวนตัวอย่างแบบ Tomek Links โดยโมเดลผ่านการเรียนรู้โดยใช้อัลกอริทึม Decision Trees, Naïve Bayes, k-Nearest Neighbors, Support Vector Machines และ Artificial Neural Networks ในการวัดตัดสินใจโมเดลจะพิจารณาจากประสิทธิภาพความสามารถในระบุ Positive class ได้อย่างถูกต้อง หรือค่า Recall ในแต่ละชุดข้อมูลที่ไม่สมดุล




**รูปที่ 3.4 แผนภาพแสดงกระบวนการ model comparison**

รูปที่ 3.4 กระบวนการขั้นตอนของการทำ model comparison พิจารณาประสิทธิภาพโมเดลที่ได้รับจากการประเมินวัดประสิทธิภาพ โดยให้ความสนใจไปที่การประเมินวัดประสิทธิภาพเปอร์เซ็นต์ความสามารถในการระบุ Positive class ได้อย่างถูกต้อง หรือค่า Recall จากการโมเดลจากการทดสอบชุดข้อมูลที่ไม่สมดุลจำนวน 10 ชุดข้อมูล

หลังจากที่ชุดข้อมูลที่ไม่สมดุลผ่านการคัดเลือกคุณลักษณะโดยแบ่งการคัดเลือกคุณลักษณะออกเป็นเปอร์เซ็นต์คุณลักษณะที่มีความเกี่ยวข้องกับประเภทคลาสในการจำแนกมากที่สุด ดังนี้ 10 เปอร์เซ็นต์ 25 เปอร์เซ็นต์ 50 เปอร์เซ็นต์ และ 75 เปอร์เซ็นต์ ชุดข้อมูลที่ผ่านการคัดเลือกคุณลักษณะถูกจัดการให้สมดุลผ่านการใช้เทคนิคการเพิ่มประสิทธิภาพโมเดลแบบเทคนิค RUSBoostClassifier เทคนิคการสุ่มตัวอย่างเพิ่มจำนวนตัวอย่างแบบเทคนิค SMOTE และเทคนิคการสุ่มตัวอย่างแบบลดจำนวนตัวอย่างแบบเทคนิค Tomek Links โดยโมเดลผ่านการฝึกสอนการเรียนรู้ผ่าน 5 อัลกอริทึม ดังนี้ Decision Trees, Naïve Bayes, k-Nearest Neighbors, Support Vector Machines และ Artificial Neural Networks โดยนำโมเดลเหล่านี้มาทดสอบบนชุดข้อมูลไม่สมดุลจำนวน 10 ชุดข้อมูล ทดสอบโมเดลและประเมินวัดประสิทธิภาพผ่านวิธีการ Cross Validation เป็นวิธีที่นิยมใช้ในงานวิจัยและมีความน่าเชื่อถือ ผลการทดสอบประสิทธิภาพโมเดล พิจารณาจากค่า Recall โดยจะสรุปผลและวิเคราะห์ผลการเปรียบเทียบประสิทธิภาพโมเดล โดยการแสดงค่า Recall เปรียบเทียบในรูปแบบตารางและกราฟเส้น เพื่อให้ง่ายต่อการนำไปวิเคราะห์ผลการทดลอง

# บทที่ 4
# ผลการวิจัยและการอภิปรายผล

## 4.1 ผลการทดลอง

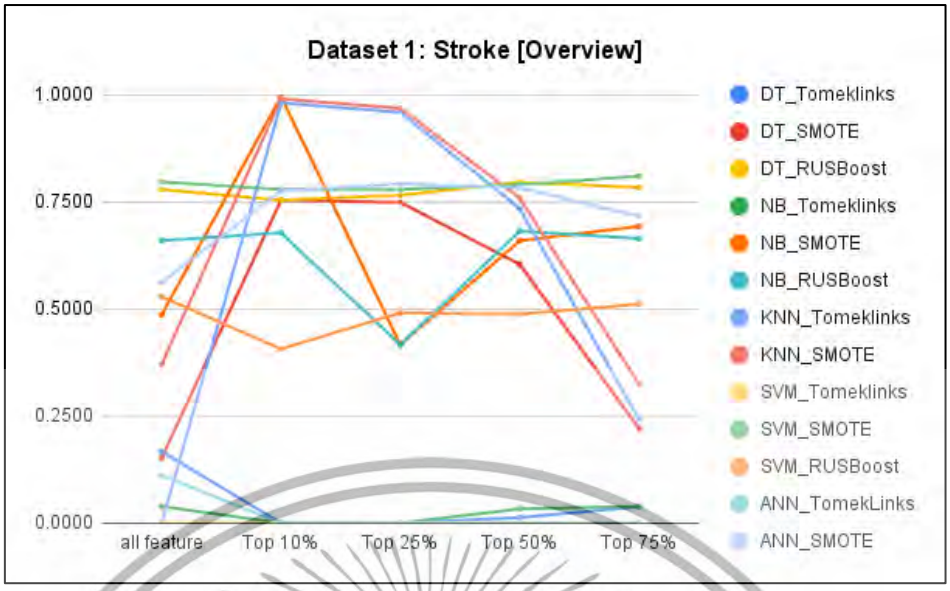
การแสดงผลการทดลองของชุดข้อมูลที่ไม่สมดุล 10 ชุดข้อมูลจากเว็บไซต์ Kaggle ผ่านการคัดเลือกคุณลักษณะที่เกี่ยวข้อง และปรับความสมดุลโดยใช้เทคนิคการสุ่มตัวอย่างด้วยเทคนิค SMOTE เทคนิค Tomek Links และเทคนิค RUSBoostClassifier โดยสร้างโมเดลให้เครื่องเรียนรู้ผ่านอัลกอริทึมประกอบด้วยอัลกอริทึม Decision Trees, Naïve Bayes, k-Nearest Neighbors, Support Vector Machines และ Artificial Neural Networks มีผลการทดลอง ดังนี้

### 4.1.1 ชุดข้อมูล Stroke Prediction

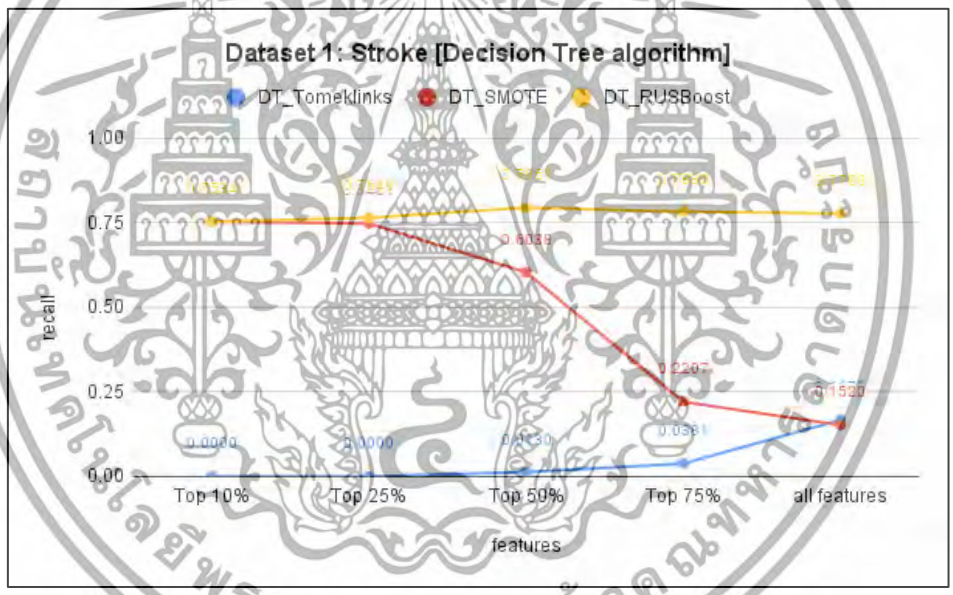
ตารางที่ 4.1 ค่า Recall ของชุดข้อมูลที่ไม่สมดุลชุดที่ 1 Stroke Prediction

| Model | all features | Top 10% | Top 25% | Top 50% | Top 75% |
|---|---|---|---|---|---|
| DT_Tomeklinks | **0.167±0.0946** | 0±0 | 0±0 | 0.013±0.0279 | 0.0381±0.0522 |
| DT_SMOTE | 0.152±0.1014 | **0.7534±0.072** | 0.7481±0.086 | 0.6038±0.0956 | 0.2207±0.0716 |
| DT_RUSBoost | 0.778±0.1073 | 0.7534±0.072 | 0.7649±0.0699 | **0.7951±0.1069** | 0.783±0.1191 |
| NB_Tomeklinks | 0.039±0.0387 | 0±0 | 0±0 | 0.0332±0.0404 | **0.0391±0.0393** |
| NB_SMOTE | 0.486±0.1312 | **0.9952±0.0151** | 0.4158±0.0891 | 0.6585±0.041 | 0.6921±0.1575 |
| NB_RUSBoost | 0.659±0.1069 | 0.6772±0.4201 | 0.4158±0.0891 | **0.6809±0.0543** | 0.6636±0.1103 |
| KNN_Tomeklinks | 0±0 | **0.9802±0** | 0.9581±0.0605 | 0.7337±0.0939 | 0.2415±0.0654 |
| KNN_SMOTE | 0.371±0.1217 | **0.9897±0** | 0.9676±0.0377 | 0.758±0.0938 | 0.3245±0.1144 |
| SVM_Tomeklinks | 0±0 | 0±0 | 0±0 | 0±0 | 0±0 |
| SVM_SMOTE | 0.796±0.1325 | 0.7782±0.1776 | 0.7782±0.1776 | 0.7893±0.175 | **0.8095±0.1551** |
| SVM_RUSBoost | **0.528±0.188** | 0.4061±0.2857 | 0.4907±0.3254 | 0.4876±0.2781 | 0.5107±0.2137 |
| ANN_TomekLinks | **0.1111±0.3143** | 0±0 | 0±0 | 0±0 | 0±0 |
| ANN_SMOTE | 0.5617±0.4721 | 0.775±0.127 | **0.791±0.1127** | 0.7819±0.1216 | 0.7161±0.117 |

จากตารางที่ 4.1 แสดงการเปรียบเทียบค่า Recall ที่แสดงจำนวนคุณลักษณะส่งผลต่อประสิทธิภาพความสามารถในการจดจำ Positive class อย่างค่า Recall การจำแนกประเภทของโมเดลที่ใช้เทคนิคการสุ่มตัวอย่าง Tomek Links เทคนิค SMOTE และเทคนิค RUSBoostClassifier ในการจัดความสมดุลของชุดข้อมูลผ่านการเรียนรู้ของเครื่องโดยใช้อัลกอริทึม Decision Trees, Naïve Bayes, k-Nearest Neighbors, Support Vector Machines และ Artificial Neural Networks

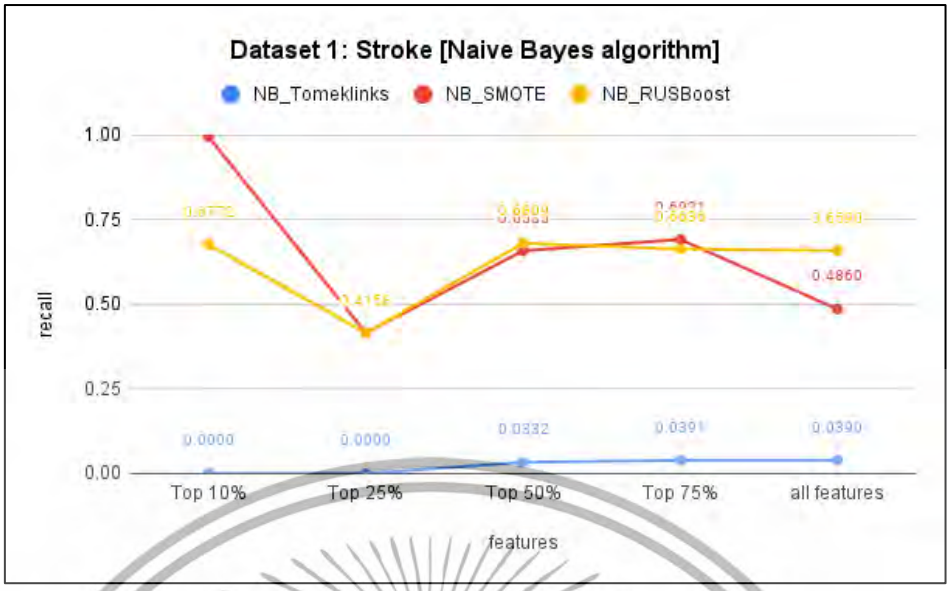



**รูปที่ 4.1 กราฟเส้นแสดงค่า Recall ในชุดข้อมูล Stroke Prediction**



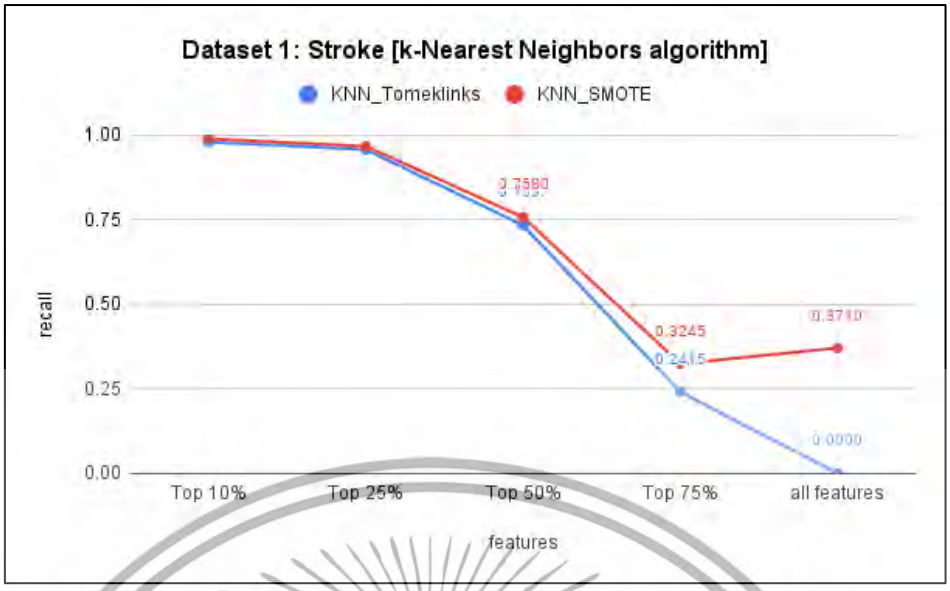

**รูปที่ 4.2 กราฟเส้นแสดงค่า Recall ในชุดข้อมูล Stroke Prediction ของอัลกอริทึม Decision Trees**

จากรูปที่ 4.2 ชุดข้อมูล Stroke Prediction ทดลองบนอัลกอริทึม Decision Trees พบว่า เทคนิค Tomek Links จำนวนคุณลักษณะมีผลต่อประสิทธิภาพในการจำแนกของโมเดล จำนวนคุณลักษณะที่ลดลงประสิทธิภาพในความสามารถการจดจำ Positive class ค่า Recall ของโมเดลมีประสิทธิภาพลดลงตามลำดับ ส่วนเทคนิค SMOTE ตรงกันข้ามกับเทคนิค Tomek Links จำนวนคุณลักษณะที่ลดลง ส่งผลให้โมเดลมีประสิทธิภาพการจำแนกค่า Recall ที่เพิ่มขึ้น และเทคนิค RUSBoostClassifier จำนวนคุณลักษณะไม่ส่งผลต่อค่า Recall ที่แตกต่างอย่างเห็นได้ชัดเจน



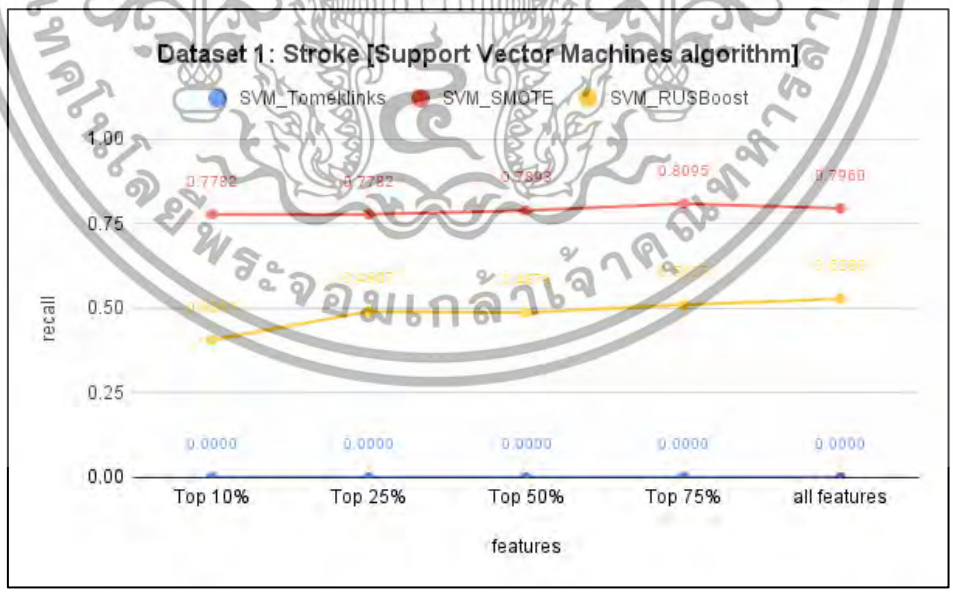

**รูปที่ 4.3 กราฟเส้นแสดงค่า Recall ในชุดข้อมูล Stroke Prediction ของอัลกอริทึม Naïve Bayes**

จากรูปที่ 4.3 ชุดข้อมูล Stroke Prediction ทดลองบนอัลกอริทึม Naïve Bayes พบว่าเทคนิค Tomek Links จำนวนคุณลักษณะมีผลต่อการจำแนกข้อมูล จำนวนคุณลักษณะที่ลดลง ส่งผลต่อประสิทธิภาพค่า Recall ลดลง ส่วนเทคนิค SMOTE ตรงกันข้ามกับ Tomek Links จำนวนคุณลักษณะที่ลดลง ส่งผลให้โมเดลมีประสิทธิภาพการจำแนกค่า Recall ที่เพิ่มขึ้น และเทคนิค RUSBoostClassifier จำนวนคุณลักษณะไม่ส่งผลต่อค่า Recall ที่แตกต่างอย่างเห็นได้ชัด และในอัลกอริทึม Naïve Bayes ที่ใช้เทคนิคเพิ่มประสิทธิภาพโมเดลทำงานร่วมกันจำนวนคุณลักษณะที่เกี่ยวข้องกันในระดับ 25 เปอร์เซ็นต์มีประสิทธิภาพต่ำกว่าจำนวนคุณลักษณะในเปอร์เซ็นต์อื่น เนื่องจากความไม่สัมพันธ์กันในคุณลักษณะ



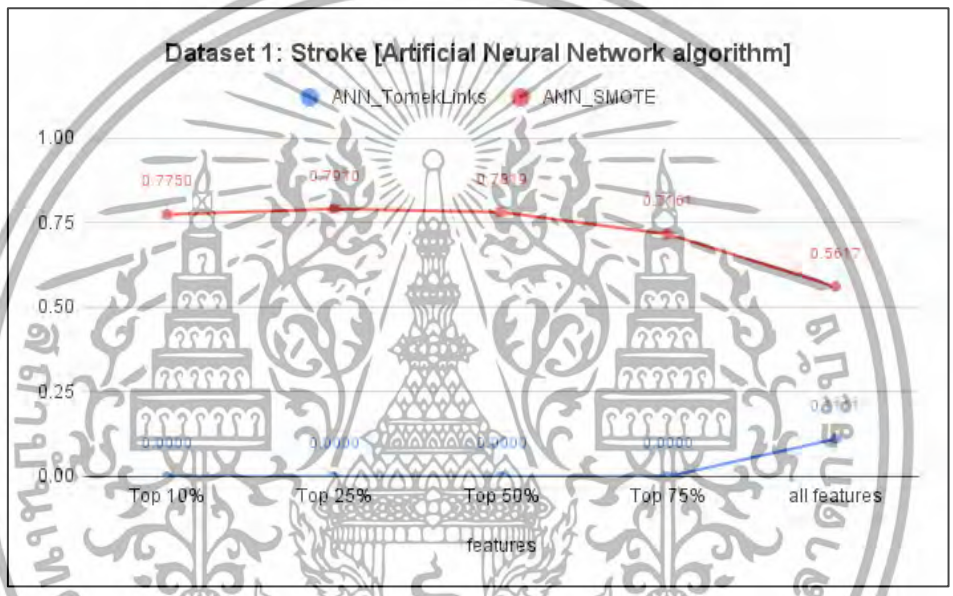

**รูปที่ 4.4 กราฟเส้นแสดงค่า Recall ในชุดข้อมูล Stroke Prediction ของอัลกอริทึม k-Nearest Neighbors**

จากรูปที่ 4.4 ชุดข้อมูล Stroke Prediction ทดลองบนอัลกอริทึม k-Nearest Neighbors พบว่าเทคนิค Tomek Links จำนวนคุณลักษณะที่ลดลงกลับทำให้โมเดลการจำแนกมีประสิทธิภาพที่ดีมากขึ้นแต่เมื่อเปรียบเทียบเทคนิค SMOTE กลับมีประสิทธิภาพค่า Recall ที่มากกว่าจากการสังเกตกราฟเส้น




**รูปที่ 4.5 กราฟเส้นแสดงค่า Recall ในชุดข้อมูล Stroke Prediction ของอัลกอริทึม Support Vector Machines**

จากรูปที่ 4.5 ชุดข้อมูล Stroke Prediction ทดลองบนอัลกอริทึม Support Vector Machines พบว่าเทคนิค Tomek Links ทำงานร่วมกับอัลกอริทึม Support Vector Machines ประเมินวัดประสิทธิภาพค่า Recall อยู่ในจุดตำแหน่งทศนิยมมากกว่าตำแหน่งที่ 6 เข้าใกล้ค่า 0 แสดงว่าเทคนิคนี้อาจจะไม่เหมาะสมกับการนำมาใช้งานร่วมกับอัลกอริทึม Support Vector Machines ในทางตรงกันข้ามเทคนิค SMOTE สามารถทำงานร่วมกับอัลกอริทึม Support Vector Machines ได้อย่างมีประสิทธิภาพ และจำนวนคุณลักษณะที่ลดลงส่งผลต่อโมเดลมีประสิทธิภาพค่า Recall ที่เพิ่มขึ้นตามลำดับ และเทคนิค RUSBoostClassifier จำนวนคุณลักษณะที่ลดลง ส่งผลต่อค่า Recall ประสิทธิภาพในการจำแนกของโมเดลมีประสิทธิภาพลดลงตาม




**รูปที่ 4.6 กราฟเส้นแสดงค่า Recall ในชุดข้อมูล Stroke Prediction ของอัลกอริทึม Artificial Neural Networks**

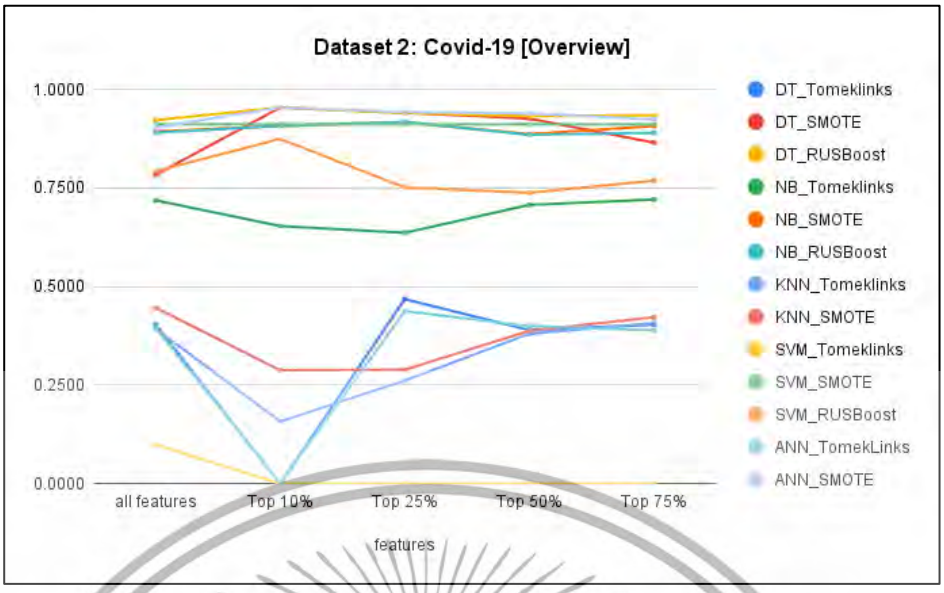
จากรูปที่ 4.6 ชุดข้อมูล Stroke Prediction ทดลองบนอัลกอริทึม Artificial Neural Networks เทคนิค Tomek Links จำนวนคุณลักษณะที่ลดลงตามลำดับ ส่งผลต่อประสิทธิภาพการจำแนกประเภทของโมเดล Artificial Neural Networks ลดลง ส่วนเทคนิค SMOTE สามารถทำงานร่วมกับอัลกอริทึม Artificial Neural Networks ได้อย่างมีประสิทธิภาพ

### 4.1.2 ชุดข้อมูล Covid-19

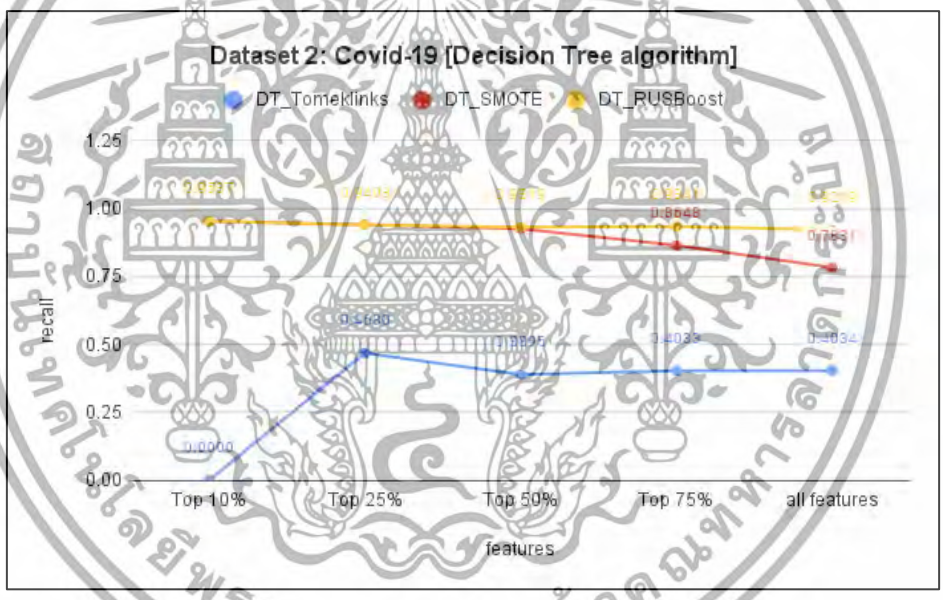
ตารางที่ 4.2 ค่า Recall ของชุดข้อมูลที่ไม่สมดุลชุดที่ 2 Covid-19

| Model | all features | Top 10% | Top 25% | Top 50% | Top 75% |
|---|---|---|---|---|---|
| DT_Tomeklinks | 0.4034±0.0135 | 0±0 | **0.468±0.0152** | 0.3895±0.019 | 0.4033±0.0181 |
| DT_SMOTE | 0.7831±0.0151 | **0.9537±0.006** | 0.9407±0.0081 | 0.9258±0.0082 | 0.8648±0.0149 |
| DT_RUSBoost | 0.9219±0.0118 | **0.9537±0.006** | 0.9403±0.0074 | 0.9319±0.0224 | 0.9341±0.0113 |
| NB_Tomeklinks | 0.7185±0.0162 | 0.6533±0.02 | 0.636±0.0217 | 0.7069±0.0157 | **0.7206±0.0151** |
| NB_SMOTE | 0.8929±0.0078 | 0.9072±0.0083 | **0.9165±0.0062** | 0.8864±0.0095 | 0.9066±0.0094 |
| NB_RUSBoost | 0.8899±0.0074 | 0.9072±0.0083 | **0.9174±0.0062** | 0.8847±0.0133 | 0.8898±0.0082 |
| KNN_Tomeklinks | 0.3929±0.0175 | 0.1578±0.3382 | 0.2612±0.1431 | 0.3801±0.0307 | 0.4069±0.0289 |
| KNN_SMOTE | **0.4462±0.0202** | 0.2875±0.3781 | 0.2887±0.0767 | 0.3877±0.0272 | **0.4221±0.0328** |
| SVM_Tomeklinks | **0.099±0.1282** | 0±0 | 0±0 | 0±0 | 0±0 |
| SVM_SMOTE | **0.9112±0.0119** | **0.9112±0.0119** | **0.9112±0.0119** | **0.9112±0.0119** | **0.9112±0.0119** |
| SVM_RUSBoost | 0.7923±0.1015 | **0.8742±0.0799** | 0.7512±0.1011 | 0.7376±0.0453 | 0.7684±0.1403 |
| ANN_TomekLinks | 0.3948±0.0412 | 0±0 | **0.4379±0.0881** | 0.3993±0.0224 | 0.3889±0.0468 |
| ANN_SMOTE | 0.9032±0.0099 | **0.9537±0.006** | 0.9422±0.007 | 0.9389±0.0064 | 0.9236±0.0123 |

จากตารางที่ 4.2 แสดงการเปรียบเทียบค่า Recall ที่แสดงจำนวนคุณลักษณะส่งผลต่อประสิทธิภาพความสามารถในการจดจำ Positive class อย่างค่า Recall การจำแนกประเภทของโมเดลที่ใช้เทคนิคการสุ่มตัวอย่าง Tomek Links เทคนิค SMOTE และเทคนิค RUSBoostClassifier ในการจัดความสมดุลของชุดข้อมูลผ่านการเรียนรู้ของเครื่องโดยใช้อัลกอริทึม Decision Trees, Naïve Bayes, k-Nearest Neighbors, Support Vector Machines และ Artificial Neural Networks

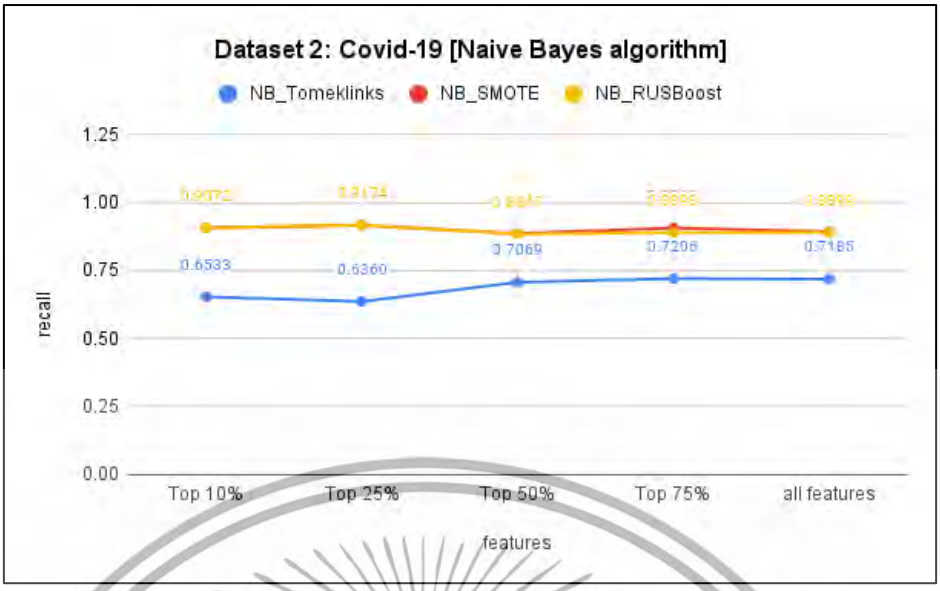



รูปที่ 4.7 กราฟเส้นแสดงค่า Recall ในชุดข้อมูล Covid-19



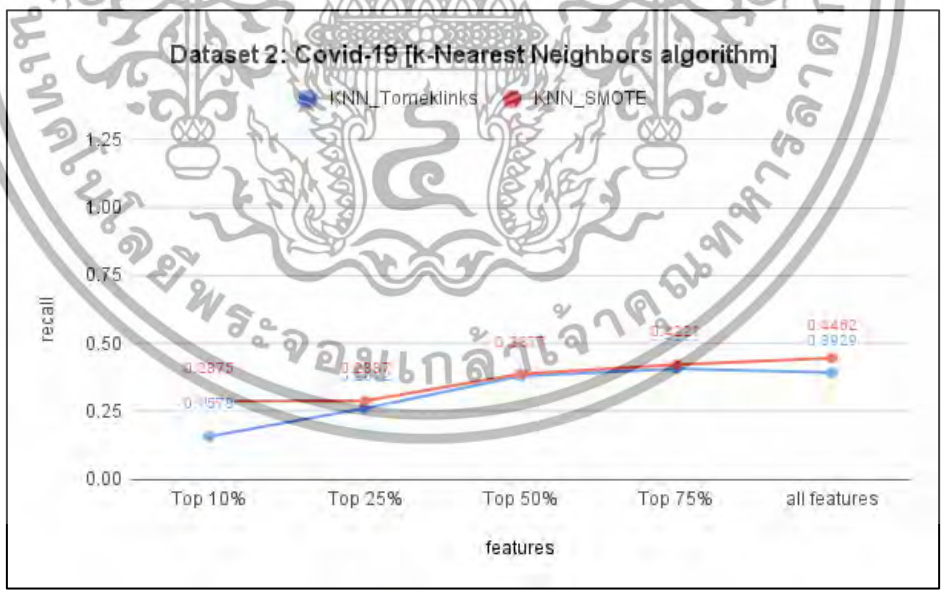

รูปที่ 4.8 กราฟเส้นแสดงค่า Recall ในชุดข้อมูล Covid-19 ของอัลกอริทึม Decision Trees

จากรูปที่ 4.8 ชุดข้อมูล Covid-19 ทดลองบนอัลกอริทึม Decision Trees พบว่าเทคนิค Tomek Links จำนวนคุณลักษณะที่ลดลงส่งผลต่อประสิทธิภาพการจำแนกของโมเดลมีค่า Recall ลดลง ในทางตรงกันข้ามเทคนิค SMOTE จำนวนคุณลักษณะที่ลดลง กลับส่งผลให้ประสิทธิภาพในการจำแนกของโมเดลมีค่า Recall เพิ่มขึ้น และเทคนิค RUSBoostClassifier จำนวนคุณลักษณะที่ลดลง ไม่ส่งผลต่อประสิทธิในการจำแนกประเภทค่า Recall ที่แตกต่างกันมากอย่างเห็นได้ชัด และสามารถทำงานกับอัลกอริทึม Decision Trees ได้อย่างมีประสิทธิภาพดีกว่าเทคนิคอื่น



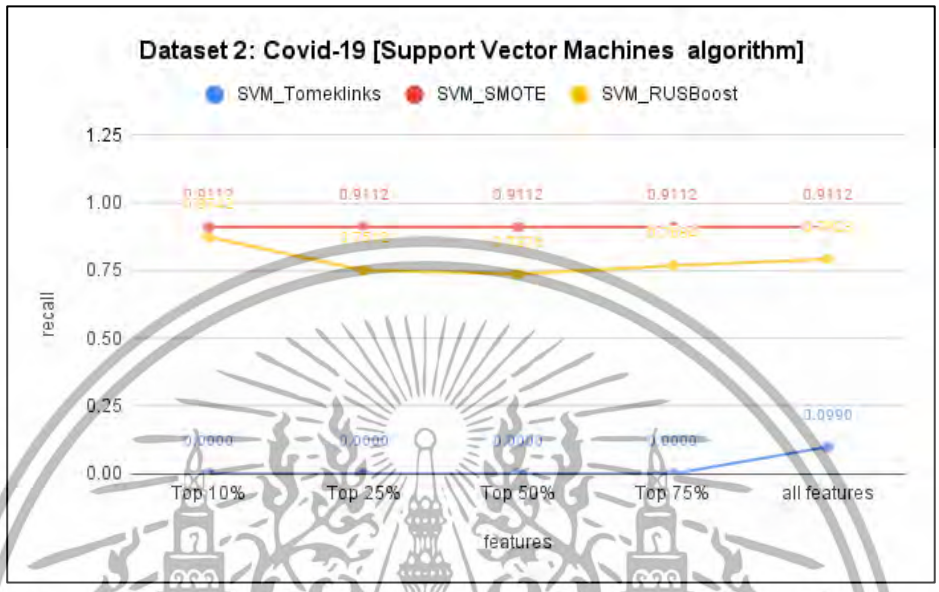

**รูปที่ 4.9 กราฟเส้นแสดงค่า Recall ในชุดข้อมูล Covid-19 ของอัลกอริทึม Naïve Bayes**

จากรูปที่ 4.9 ชุดข้อมูล Covid-19 ทดลองบนอัลกอริทึม Naïve Bayes พบว่าเทคนิค Tomek Links จำนวนคุณลักษณะที่ลดลงส่งผลต่อประสิทธิภาพการจำแนกของโมเดลลดลง ส่วนเทคนิค SMOTE และเทคนิค RUSBoostClassifier จำนวนคุณลักษณะที่ลดลง ไม่ส่งผลต่อประสิทธิภาพค่า Recall ที่แตกต่างกันอย่างเห็นได้ชัดเจน



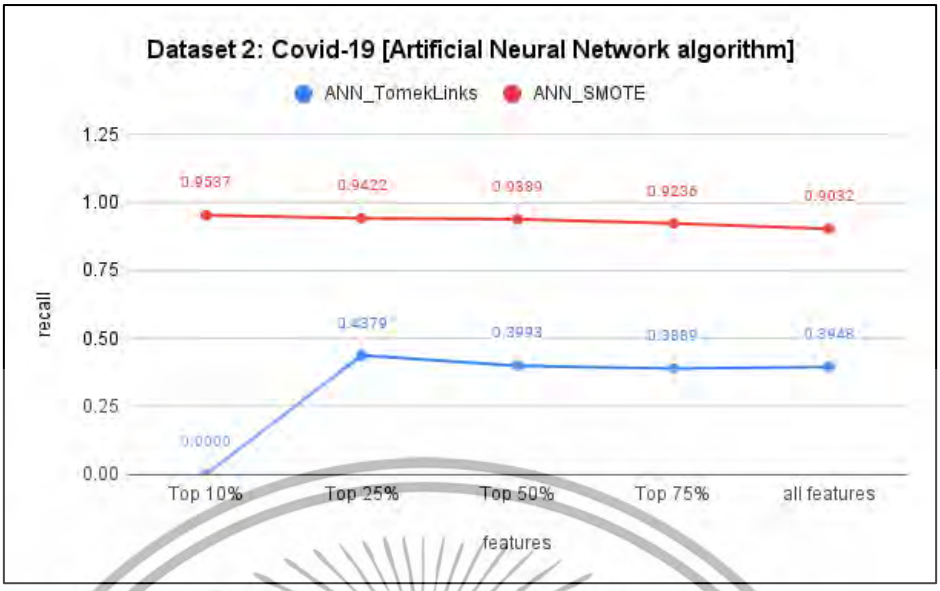

**รูปที่ 4.10 กราฟเส้นแสดงค่า Recall ในชุดข้อมูล Covid-19 ของอัลกอริทึม k-Nearest Neighbors**

จากรูปที่ 4.10 ชุดข้อมูล Covid-19 ทดลองบนอัลกอริทึม k-Nearest Neighbors พบว่าเทคนิค Tomek Links และเทคนิค SMOTE จำนวนคุณลักษณะที่ลดลงส่งผลต่อประสิทธิภาพการจำแนกของโมเดลมีค่า Recall ลดลงตามลำดับ




**รูปที่ 4.11 กราฟเส้นแสดงค่า Recall ในชุดข้อมูล Covid-19 ของอัลกอริทึม Support Vector Machines**

จากรูปที่ 4.11 ชุดข้อมูล Covid-19 ทดลองบนอัลกอริทึม Support Vector Machines พบว่าเทคนิค Tomek Links จำนวนคุณลักษณะที่ลดลงส่งผลต่อประสิทธิภาพในการจำแนกประเภทของโมเดลมีค่า Recall ที่ลดลง แต่ในทางกลับกันจำนวนคุณลักษณะไม่ผลส่งประสิทธิภาพในการจำแนกประเภทของโมเดลที่ใช้เทคนิคการสุ่มตัวอย่างแบบ SMOTE และในส่วนของเทคนิค RUSBoostClassifier จำนวนคุณลักษณะมีผลต่อค่า Recall ประสิทธิภาพในการจำแนกประเภทของโมเดล




**รูปที่ 4.12 กราฟเส้นแสดงค่า Recall ในชุดข้อมูล Covid-19 ของอัลกอริทึม Artificial Neural Networks**

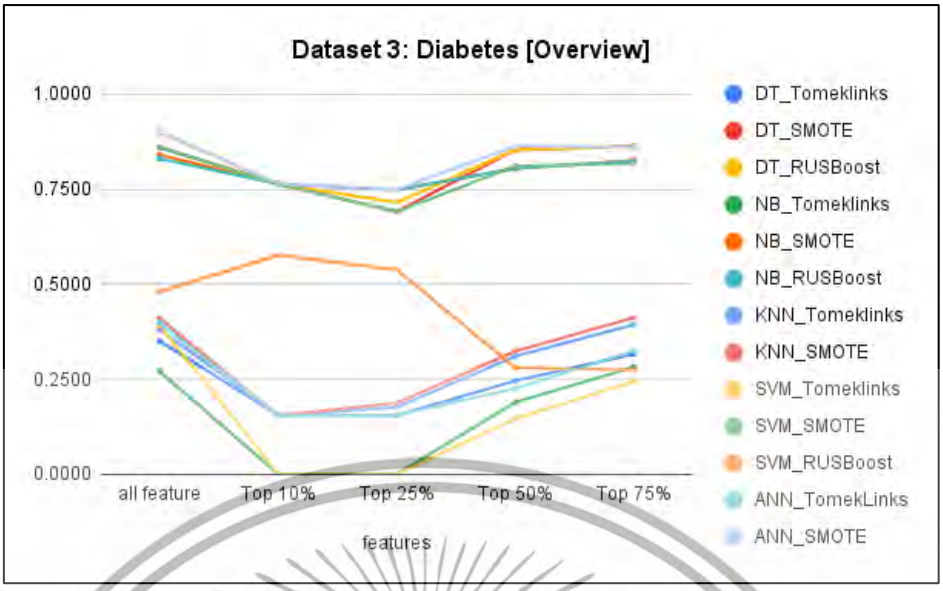
จากรูปที่ 4.12 ชุดข้อมูล Covid-19 ทดลองบนอัลกอริทึม Artificial Neural Networks พบว่าเทคนิค Tomek Links จำนวนคุณลักษณะที่ลดลงส่งผลต่อประสิทธิภาพการจำแนกของโมเดลมีค่า Recall ลดลง ในทางตรงกันข้ามเทคนิค SMOTE จำนวนคุณลักษณะที่ลดลง กลับไม่ส่งผลต่อประสิทธิภาพในการจำแนกของโมเดลมีค่า Recall ที่ไม่แตกต่างกันมาก

### 4.1.3 ชุดข้อมูล Diabetes Prediction

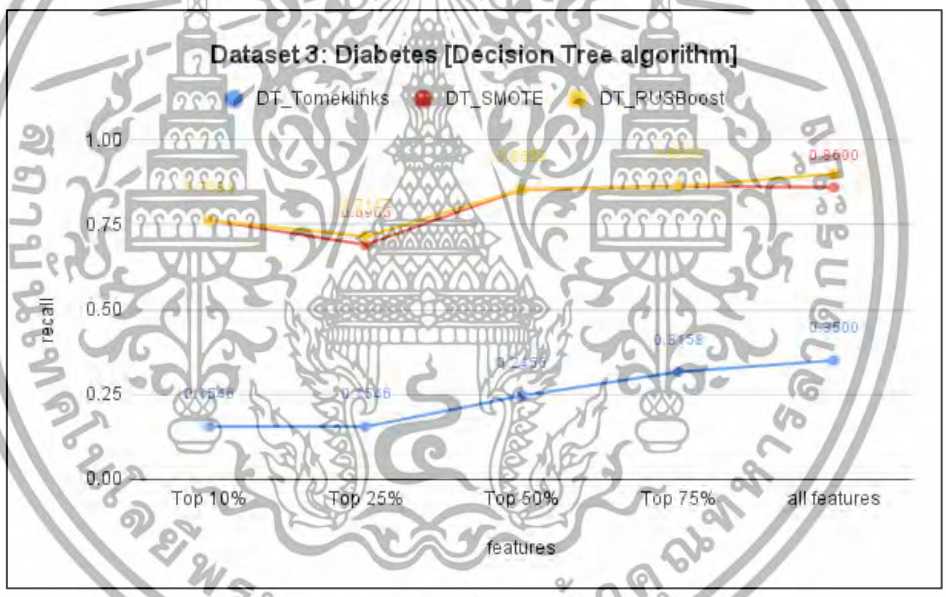
ตารางที่ 4.3 ค่า Recall ของชุดข้อมูลที่ไม่สมดุลชุดที่ 3 Diabetes Prediction

| Model | all features | Top 10% | Top 25% | Top 50% | Top 75% |
|---|---|---|---|---|---|
| DT_Tomeklinks | **0.35±0.0125** | 0.1546±0.0079 | 0.1546±0.0079 | 0.2456±0.0089 | 0.3158±0.0408 |
| DT_SMOTE | 0.86±0.0158 | 0.7634±0.0166 | 0.6905±0.0156 | 0.8539±0.0113 | **0.8631±0.0103** |
| DT_RUSBoost | **0.9±0.0577** | 0.7634±0.0166 | 0.7147±0.0566 | 0.8539±0.0113 | 0.862±0.0127 |
| NB_Tomeklinks | 0.27±0.0106 | 0±0 | 0±0 | 0.189±0.0046 | **0.2814±0.0097** |
| NB_SMOTE | **0.84±0.0166** | 0.7634±0.0166 | 0.7478±0.015 | 0.8046±0.0104 | 0.8254±0.0113 |
| NB_RUSBoost | **0.83±0.0106** | 0.7634±0.0166 | 0.7478±0.015 | 0.8046±0.0104 | 0.8236±0.0157 |
| KNN_Tomeklinks | 0.38±0.0107 | 0.1546±0.0079 | 0.176±0.067 | 0.3102±0.1167 | **0.3932±0.0842** |
| KNN_SMOTE | 0.41±0.0105 | 0.1546±0.0079 | 0.1873±0.1045 | 0.3234±0.1278 | **0.4118±0.0551** |
| SVM_Tomeklinks | **0.39±0.1451** | 0±0 | 0±0 | 0.1459±0.1344 | 0.2438±0.1453 |
| SVM_SMOTE | **0.86±0.0088** | 0.7634±0.0166 | 0.6905±0.0156 | 0.8104±0.0776 | 0.8177±0.0085 |
| SVM_RUSBoost | 0.48±0.0166 | **0.5764±0.2929** | 0.5377±0.3486 | 0.2797±0.0563 | 0.2739±0.1447 |
| ANN_TomekLinks | **0.3981±0.0226** | 0.1546±0.0079 | 0.1546±0.0079 | 0.2255±0.0566 | 0.3238±0.063 |
| ANN_SMOTE | **0.9013±0.0199** | 0.7634±0.0166 | 0.7485±0.0688 | 0.8631±0.0223 | 0.8602±0.0211 |

จากตารางที่ 4.3 แสดงการเปรียบเทียบค่า Recall ที่แสดงจำนวนคุณลักษณะส่งผลต่อประสิทธิภาพความสามารถในการจดจำ Positive class อย่างค่า Recall การจำแนกประเภทของโมเดลที่ใช้เทคนิคการสุ่มตัวอย่าง Tomek Links เทคนิค SMOTE และเทคนิค RUSBoostClassifier ในการจัดความสมดุลของชุดข้อมูลผ่านการเรียนรู้ของเครื่องโดยใช้อัลกอริทึม Decision Trees, Naïve Bayes, k-Nearest Neighbors, Support Vector Machines และ Artificial Neural Networks

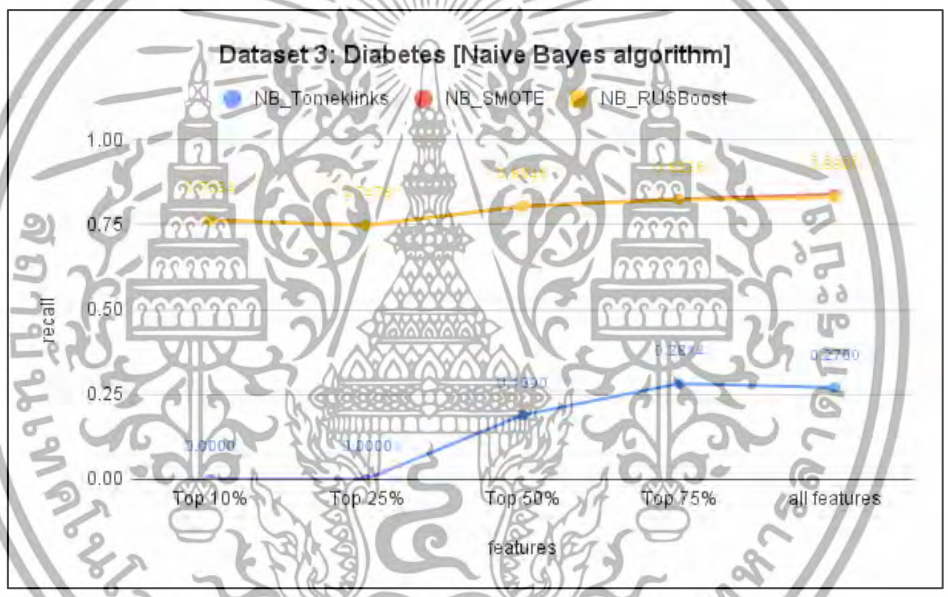



**รูปที่ 4.13 กราฟเส้นแสดงค่า Recall ในชุดข้อมูล Diabetes Prediction**



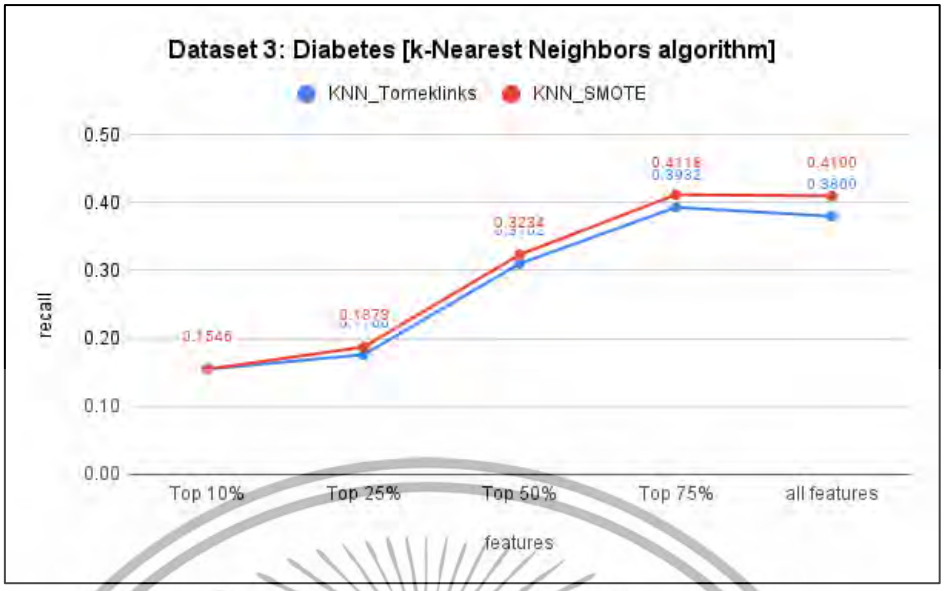

**รูปที่ 4.14 กราฟเส้นแสดงค่า Recall ในชุดข้อมูล Diabetes Prediction ของอัลกอริทึม Decision Trees**

จากรูปที่ 4.14 ชุดข้อมูล Diabetes Prediction ทดลองบนอัลกอริทึม Decision Trees พบว่าจำนวนคุณลักษณะของชุดข้อมูลที่ไม่สมดุลนี้ จำนวนคุณลักษณะทั้งหมดมีจำนวนที่น้อยมาก และมีความสัมพันธ์กัน ส่งผลต่อจำนวนคุณลักษณะที่ผ่านการคัดเลือกคุณลักษณะให้มีคุณลักษณะที่สัมพันธ์กันมาก การวัดประสิทธิภาพผ่านการเรียนรู้โมเดล มีประสิทธิภาพค่า Recall ที่ลดลงตามลำดับ และเทคนิค Tomek Links มีประสิทธิภาพต่ำที่สุด เมื่อเปรียบเทียบระหว่าง 3 เทคนิค จำนวนคุณลักษณะมีผลต่อโมเดลในการจำแนกประเภทข้อมูล โดยจำนวนคุณลักษณะลดลง ค่า Recall มีประสิทธิภาพลดลงตาม เทคนิค SMOTE ตรงกันข้ามกับเทคนิค Tomek Links จำนวนคุณลักษณะที่ลดลง ส่งผลต่อโมเดลมีประสิทธิภาพการจำแนกค่า Recall ที่เพิ่มขึ้น และเทคนิค RUSBoostClassifier จำนวนคุณลักษณะไม่ส่งผลต่อประสิทธิภาพการจำแนกของโมเดลที่ค่า Recall แสดงผลความแตกต่างกันมาก



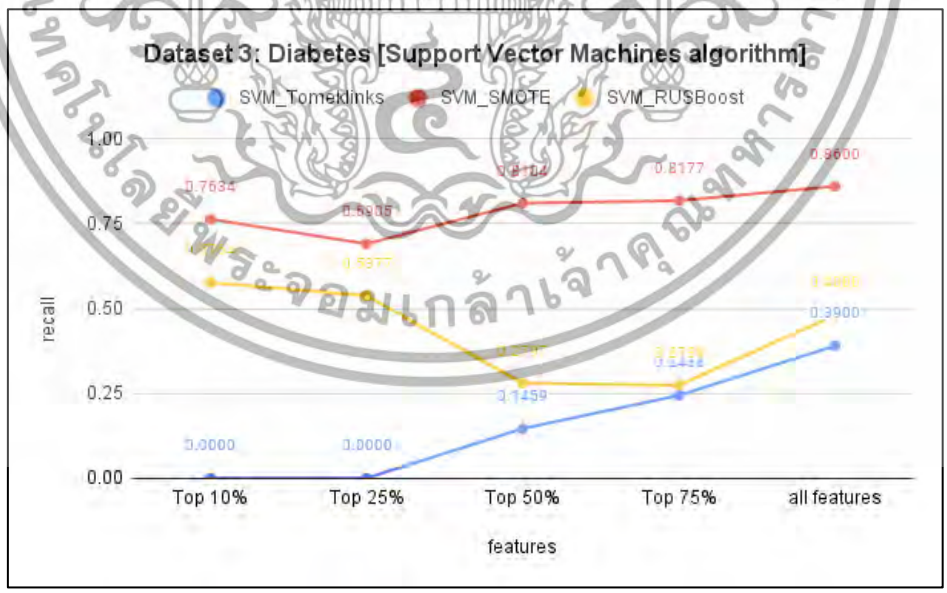

**รูปที่ 4.15 กราฟเส้นแสดงค่า Recall ในชุดข้อมูล Diabetes Prediction ของอัลกอริทึม Naïve Bayes**

จากรูปที่ 4.15 ชุดข้อมูล Diabetes Prediction ทดลองบนอัลกอริทึม Naïve Bayes พบว่าจำนวนคุณลักษณะของชุดข้อมูลที่ไม่สมดุลนี้ จำนวนคุณลักษณะทั้งหมดมีจำนวนที่น้อยมากและมีความสัมพันธ์กัน ส่งผลให้จำนวน คุณลักษณะที่ผ่านการคัดเลือกคุณลักษณะให้มีคุณลักษณะที่สัมพันธ์กันมากที่สุด ทดสอบวัดประสิทธิผ่านการเรียนรู้โมเดล มีประสิทธิภาพการจำแนกที่ค่า Recall ลดลงตามลำดับ ส่วนเทคนิค Tomek Links มีประสิทธิภาพต่ำที่สุด เมื่อเทียบกับเทคนิค SMOTE และเทคนิค RUSBoostClassifier ที่มีประสิทธิภาพในการทำงานร่วมกับอัลกอริทึม Naïve Bayes ได้อย่างมีประสิทธิภาพ



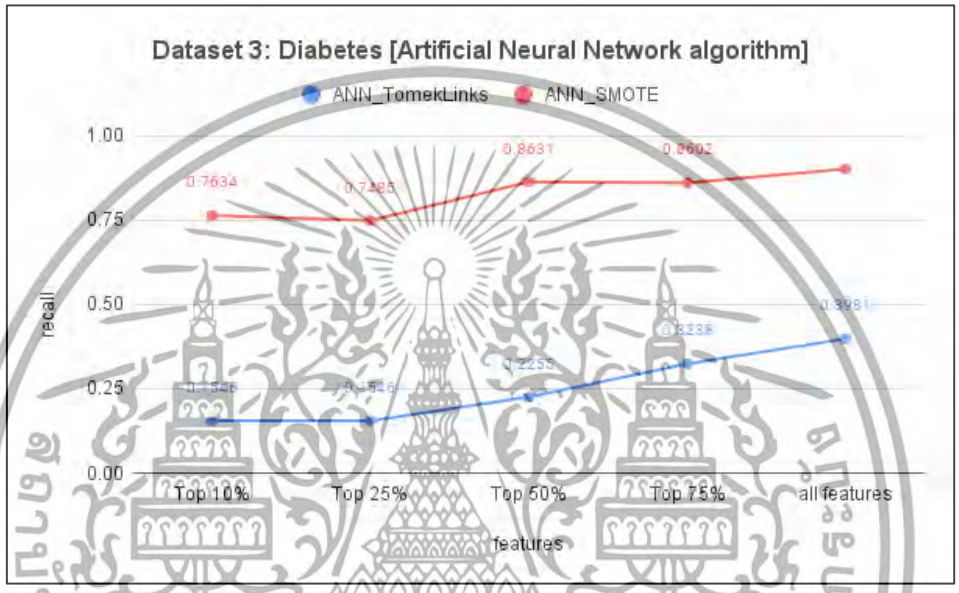

**รูปที่ 4.16 กราฟเส้นแสดงค่า Recall ในชุดข้อมูล Diabetes Prediction ของอัลกอริทึม k-Nearest Neighbors**

จากรูปที่ 4.16 ชุดข้อมูล Diabetes Prediction ทดลองบนอัลกอริทึม k-Nearest Neighbors พบว่าเทคนิค Tomek Links และเทคนิค SMOTE มีประสิทธิภาพการจำแนกที่ดี แต่เปอร์เซ็นต์ความความสามารถในการระบุ Positive class ได้ถูกต้อง หรือค่า Recall อยู่เกณฑ์ที่ไม่ดี และมีค่าไม่ถึง 50 เปอร์เซ็นต์




**รูปที่ 4.17 กราฟเส้นแสดงค่า Recall ในชุดข้อมูล Diabetes Prediction ของอัลกอริทึม Support Vector Machines**

จากรูปที่ 4.17 ชุดข้อมูล Diabetes Prediction ทดลองบนอัลกอริทึม Support Vector Machines พบว่าเทคนิค Tomek Links จำนวนคุณลักษณะที่ลดลง ส่งผลต่อประสิทธิภาพการจำแนกของโมเดลมีค่า Recall มีประสิทธิภาพที่ลดลง เมื่อเปรียบเทียบกับเทคนิคอื่น ส่วนเทคนิค SMOTE ทำงานร่วมกันได้ดีกับอัลกอริทึม Support Vector Machines อัลกอริทึมที่ใช้การคำนวณระยะทางระหว่างจุดข้อมูล และเทคนิค RUSBoostClassifier กลับให้ผลลัพธ์ว่าจำนวนคุณลักษณะที่ลดลง ส่งผลต่อประสิทธิภาพการจำแนกของโมเดลมีค่า Recall ที่เพิ่มขึ้น




**รูปที่ 4.18 กราฟเส้นแสดงค่า Recall ในชุดข้อมูล Diabetes Prediction ของอัลกอริทึม Artificial Neural Networks**

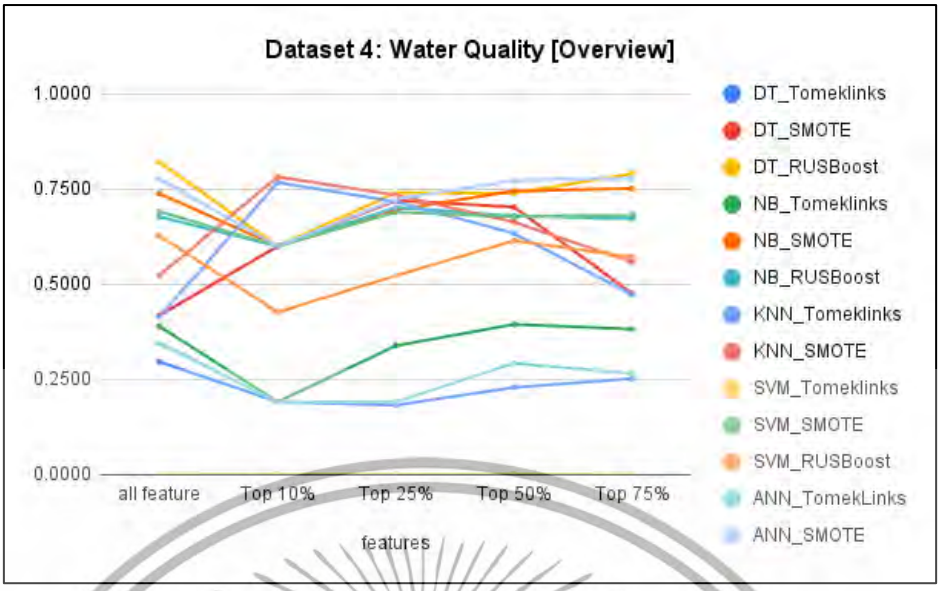
จากรูปที่ 4.18 ชุดข้อมูล Diabetes Prediction ทดลองบนอัลกอริทึม Artificial Neural Networks พบว่าจำนวนคุณลักษณะส่งผลต่อการจำแนกประเภทข้อมูลทั้ง 2 เทคนิคที่เครื่องใช้การเรียนรู้ผ่านอัลกอริทึม Artificial Neural Networks เทคนิค Tomek Links มีค่า Recall ที่ต่ำเมื่อเทียบกับเทคนิค SMOTE และเทคนิค SMOTE สามารถทำงานร่วมกับอัลกอริทึม Artificial Neural Networks ได้อย่างมีประสิทธิภาพ

### 4.1.4 ชุดข้อมูล Water Quality

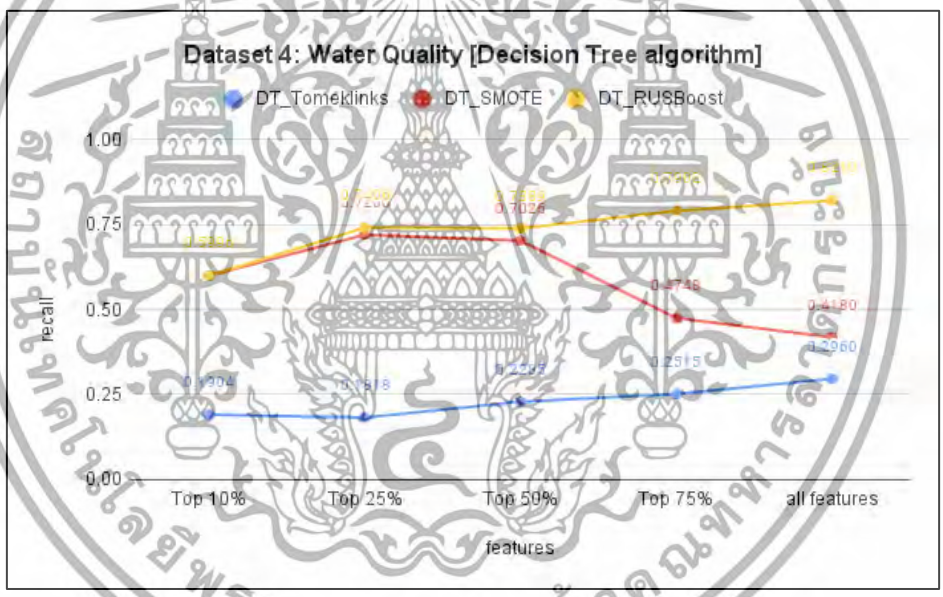
ตารางที่ 4.4 ค่า Recall ของชุดข้อมูลที่ไม่สมดุลชุดที่ 4 Water Quality

| Model | all features | Top 10% | Top 25% | Top 50% | Top 75% |
|---|---|---|---|---|---|
| DT_Tomeklinks | **0.296±0.0378** | 0.1904±0.0431 | 0.1818±0.0442 | 0.2285±0.0441 | 0.2515±0.0256 |
| DT_SMOTE | 0.418±0.0322 | 0.5996±0.0542 | **0.72±0.1044** | 0.7026±0.0418 | 0.4748±0.0498 |
| DT_RUSBoost | **0.821±0.0651** | 0.5996±0.0542 | 0.7406±0.0504 | 0.7389±0.0818 | 0.7902±0.0594 |
| NB_Tomeklinks | 0.389±0.0384 | 0.1904±0.0431 | 0.3383±0.049 | **0.3943±0.054** | 0.3819±0.0351 |
| NB_SMOTE | 0.738±0.0565 | 0.5996±0.0542 | 0.6951±0.0746 | 0.7451±0.0451 | **0.7515±0.0416** |
| NB_RUSBoost | 0.677±0.0529 | 0.5996±0.0542 | **0.7017±0.073** | 0.679±0.0672 | 0.6724±0.0529 |
| KNN_Tomeklinks | 0.415±0.0425 | **0.7671±0.233** | 0.7168±0.1092 | 0.6335±0.0595 | 0.4723±0.0564 |
| KNN_SMOTE | 0.523±0.0365 | **0.7823±0.1065** | 0.7342±0.0802 | 0.6633±0.0529 | 0.5578±0.0456 |
| SVM_Tomeklinks | 0±0 | 0±0 | 0±0 | 0±0 | 0±0 |
| SVM_SMOTE | **0.691±0.0543** | 0.5996±0.0542 | 0.6893±0.1327 | 0.6772±0.0656 | 0.6826±0.063 |
| SVM_RUSBoost | **0.627±0.102** | 0.4264±0.1601 | 0.5226±0.0809 | 0.6143±0.0403 | 0.5714±0.127 |
| ANN_TomekLinks | **0.3445±0.0636** | 0.1904±0.0431 | 0.1904±0.0431 | 0.292±0.0507 | 0.2645±0.0564 |
| ANN_SMOTE | 0.7765±0.0623 | 0.5996±0.0542 | 0.7232±0.1051 | 0.7727±0.0462 | **0.7785±0.0492** |

จากตารางที่ 4.4 แสดงการเปรียบเทียบค่า Recall ที่แสดงจำนวนคุณลักษณะส่งผลต่อประสิทธิภาพความสามารถในการจดจำ Positive class อย่างค่า Recall การจำแนกประเภทของโมเดลที่ใช้เทคนิคการสุ่มตัวอย่าง Tomek Links เทคนิค SMOTE และเทคนิค RUSBoostClassifier ในการจัดความสมดุลของชุดข้อมูลผ่านการเรียนรู้ของเครื่องโดยใช้อัลกอริทึม Decision Trees, Naïve Bayes, k-Nearest Neighbors, Support Vector Machines และ Artificial Neural Networks

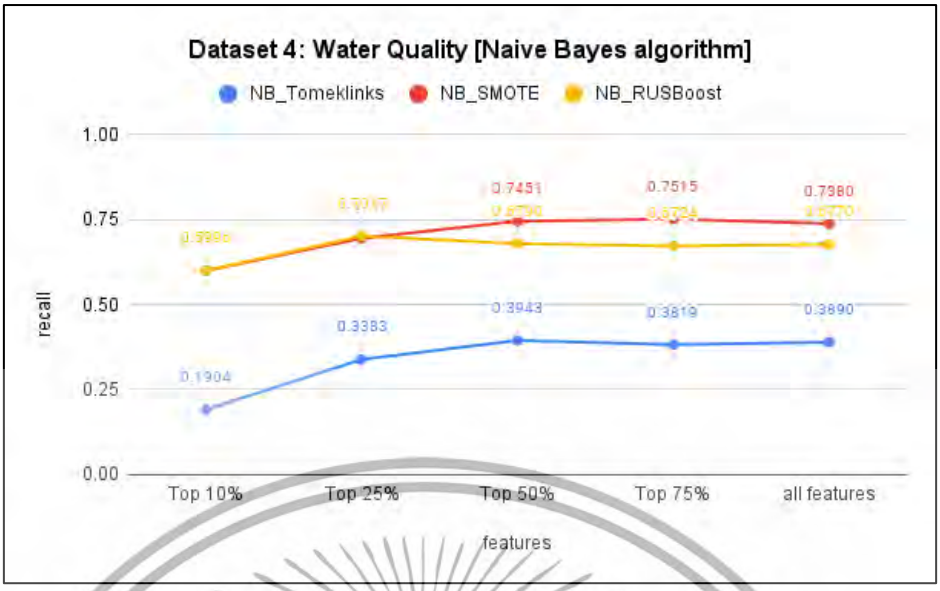



**รูปที่ 4.19 กราฟเส้นแสดงค่า Recall ในชุดข้อมูล Water Quality**



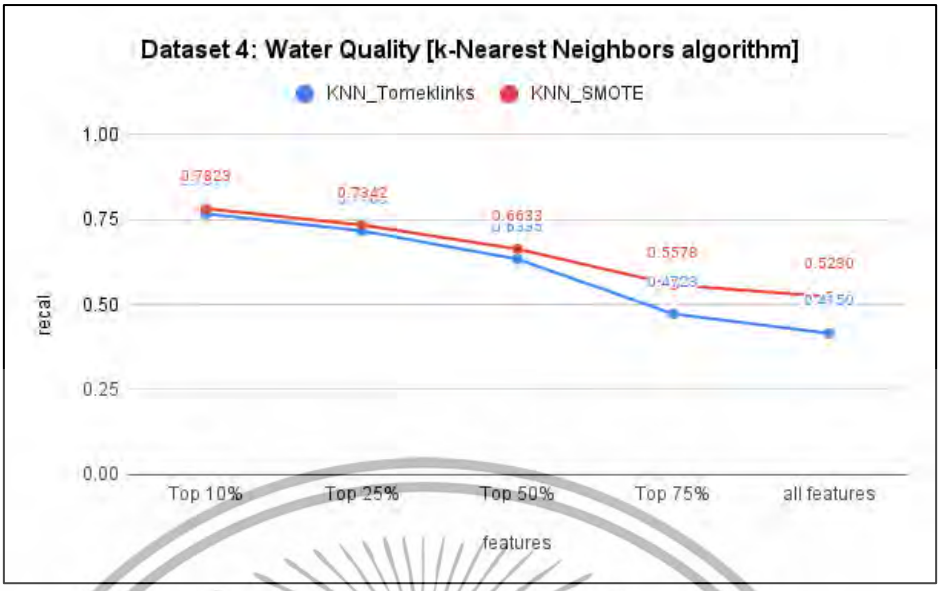

**รูปที่ 4.20 กราฟเส้นแสดงค่า Recall ในชุดข้อมูล Water Quality ของอัลกอริทึม Decision Trees**

จากรูปที่ 4.20 ชุดข้อมูล Water Quality ทดลองบนอัลกอริทึม Decision Trees พบว่า เทคนิค Tomek Links จำนวนคุณลักษณะที่ลดลง ส่งผลต่อมีประสิทธิภาพการจำแนกค่า Recall ที่ลดลงเป็นอย่างมากเมื่อเทียบกับเทคนิคอื่น ในส่วนของเทคนิค SMOTE จำนวนคุณลักษณะที่ลดลง ส่งผลโมเดลมีประสิทธิภาพการจำแนกค่า Recall ได้ดีเพิ่มขึ้น และเทคนิค RUSBoostClassifier จำนวนคุณลักษณะที่ลดลง ส่งผลต่อโมเดลประสิทธิภาพในการจำแนกประเภทค่า Recall ลดลงตามลำดับ



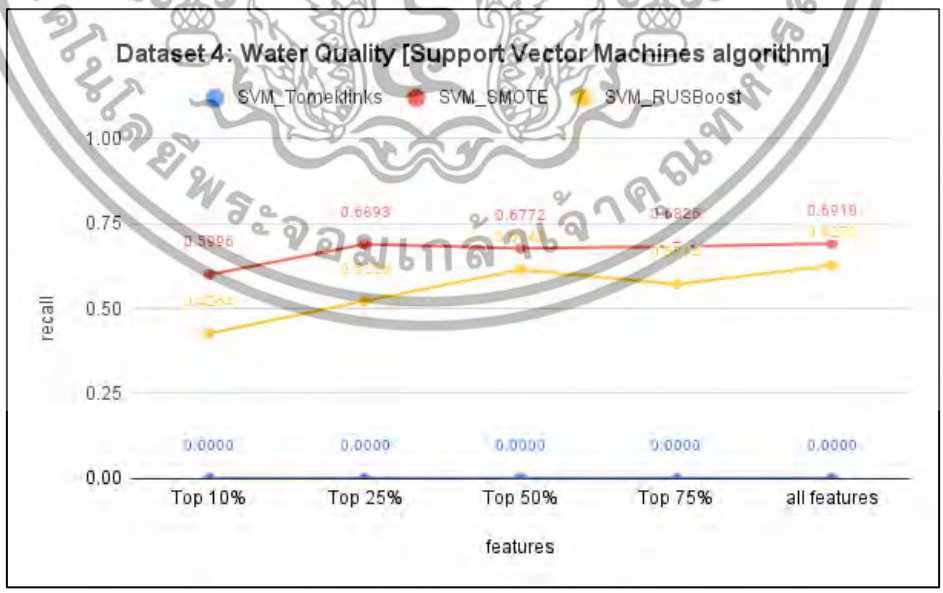

**รูปที่ 4.21 กราฟเส้นแสดงค่า Recall ในชุดข้อมูล Water Quality ของอัลกอริทึม Naïve Bayes**

จากรูปที่ 4.21 ชุดข้อมูล Water Quality ทดลองบนอัลกอริทึม Naïve Bayes พบว่าเทคนิค Tomek Links จำนวน คุณลักษณะที่ลดลงส่งผลต่อเทคนิคการสุ่มตัวอย่างนี้ ทำให้มีประสิทธิภาพการจำแนกค่า Recall ที่ลดลงเป็นอย่างมาก เมื่อเปรียบเทียบกับเทคนิคอื่น เทคนิค SMOTE เมื่อจำนวนคุณลักษณะที่ลดลง ส่งผลให้โมเดลมีประสิทธิภาพการจำแนกค่า Recall ที่ลดลง และเทคนิค RUSBoostClassifier จำนวนคุณลักษณะที่ลดลงส่งผลต่อประสิทธิภาพในการจำแนกค่า Recall ที่ลดลง



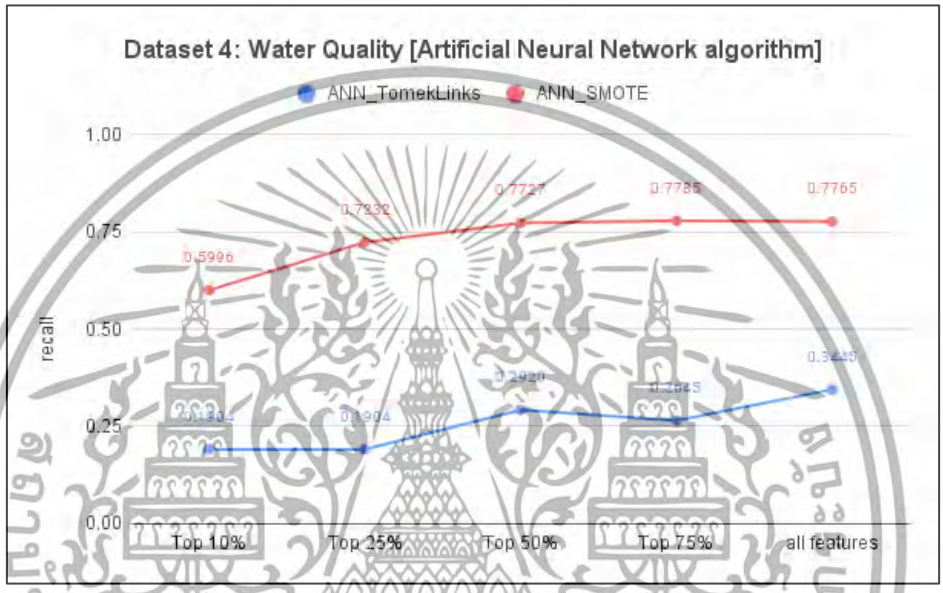

**รูปที่ 4.22 กราฟเส้นแสดงค่า Recall ในชุดข้อมูล Water Quality ของอัลกอริทึม k-Nearest Neighbors**

จากรูปที่ 4.22 ชุดข้อมูล Water Quality ทดลองบนอัลกอริทึม k-Nearest Neighbors พบว่าเทคนิค Tomek Links จำนวนคุณลักษณะที่ลดลง ส่งผลต่อประสิทธิภาพการจำแนกค่า Recall ที่เพิ่มขึ้น เมื่อทำงานร่วมกับอัลกอริทึมเพื่อนบ้านที่ใกล้เคียงกันจำนวน k ตัว และเทคนิค SMOTE จำนวนคุณลักษณะที่ลดลง ส่งผลให้โมเดลมีประสิทธิภาพการจำแนกค่า Recall ได้ดีอย่างมีประสิทธิภาพ




**รูปที่ 4.23 กราฟเส้นแสดงค่า Recall ในชุดข้อมูล Water Quality ของอัลกอริทึม Support Vector Machines**

จากรูปที่ 4.23 ชุดข้อมูล Water Quality ทดลองบนอัลกอริทึม Support Vector Machines พบว่าเทคนิค Tomek Links มีประสิทธิภาพในการจำแนกประสิทธิภาพค่า Recall ที่ต่ำเมื่อทำงานร่วมกับอัลกอริทึม Support Vector Machines ส่วนเทคนิค SMOTE จำนวนคุณลักษณะที่ลดลง ส่งผลต่อโมเดลมีประสิทธิภาพการจำแนกประเภทของโมเดลมีค่า Recall ลดลงตามลำดับ และเทคนิค RUSBoostClassifier จำนวนคุณลักษณะส่งผลต่อโมเดล โดยประสิทธิภาพในการจำแนกจะแปรผันตรงกับจำนวนคุณลักษณะ




**รูปที่ 4.24 กราฟเส้นแสดงค่า Recall ในชุดข้อมูล Water Quality ของอัลกอริทึม Artificial Neural Networks**

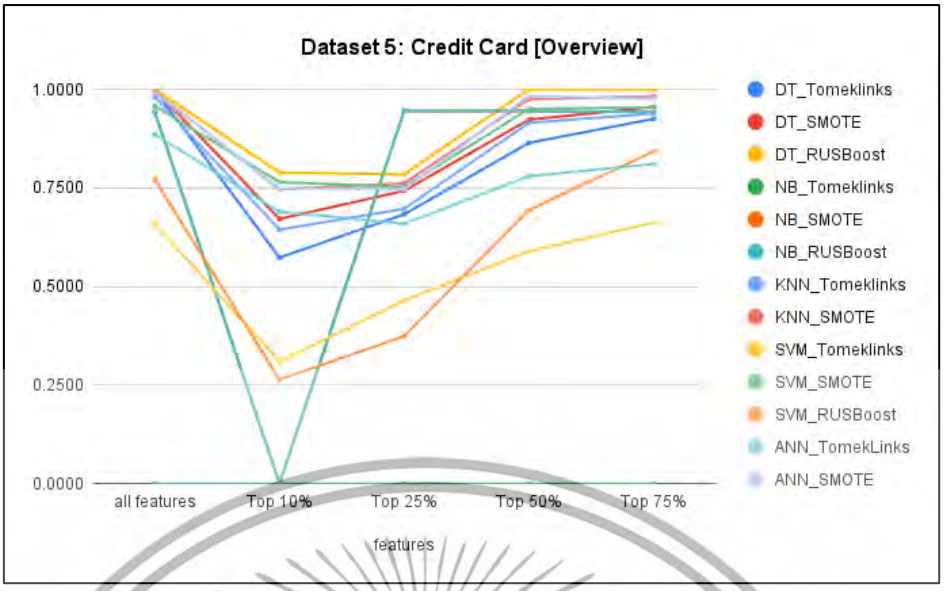
จากรูปที่ 4.24 ชุดข้อมูล Water Quality ทดลองบนอัลกอริทึม Artificial Neural Networks พบว่าเทคนิค Tomek Links มีประสิทธิภาพในการจำแนกประสิทธิภาพค่า Recall ที่ต่ำเมื่อเทียบกับเทคนิค SMOTE และเทคนิค SMOTE จำนวนคุณลักษณะที่ลดลง ส่งผลต่อโมเดลมีประสิทธิภาพการจำแนกค่า Recall ลดลงแปรผันตรงกับจำนวนคุณลักษณะ

### 4.1.5 ชุดข้อมูล Credit Card Fraud

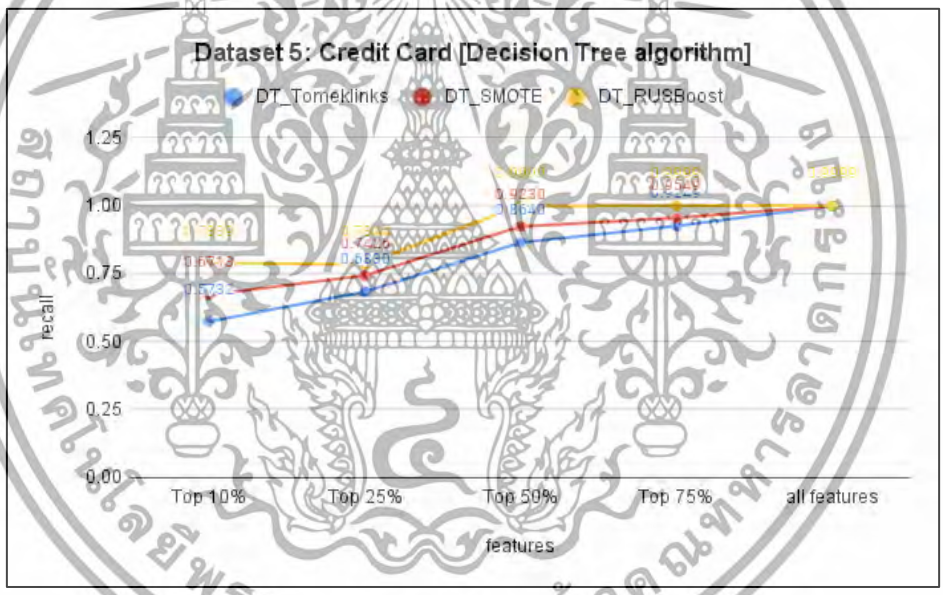
ตารางที่ 4.5 ค่า Recall ของชุดข้อมูลที่ไม่สมดุลชุดที่ 5 Credit Card Fraud

| Model | all features | Top 10% | Top 25% | Top 50% | Top 75% |
|---|---|---|---|---|---|
| DT_Tomeklinks | **0.9991±0.0012** | 0.5732±0.0177 | 0.683±0.0184 | 0.864±0.0114 | 0.9249±0.011 |
| DT_SMOTE | **0.9993±0.0008** | 0.6713±0.0132 | 0.7426±0.0171 | 0.923±0.0131 | 0.9549±0.0071 |
| DT_RUSBoost | **0.9999±0.0004** | 0.7889±0.0099 | 0.7844±0.0114 | **0.9999±0.0004** | **0.9999±0.0004** |
| NB_Tomeklinks | 0±0 | 0±0 | 0±0 | 0±0 | 0±0 |
| NB_SMOTE | 0.9423±0.0102 | 0±0 | **0.945±0.0098** | **0.945±0.0098** | 0.9423±0.0102 |
| NB_RUSBoost | 0.9423±0.0102 | 0±0 | **0.945±0.0098** | **0.945±0.0098** | 0.9423±0.0102 |
| KNN_Tomeklinks | **0.9802±0.0049** | 0.6442±0.0136 | 0.6971±0.0161 | 0.9149±0.0108 | 0.9385±0.0093 |
| KNN_SMOTE | **0.9941±0.0044** | 0.7465±0.0123 | 0.7615±0.016 | 0.9748±0.0067 | 0.9822±0.005 |
| SVM_Tomeklinks | 0.6584±0.0775 | 0.31±0.0581 | 0.4643±0.0764 | 0.5893±0.0788 | **0.6619±0.0777** |
| SVM_SMOTE | **0.9575±0.0282** | 0.7647±0.0103 | 0.7523±0.0134 | 0.9503±0.0287 | 0.9542±0.0277 |
| SVM_RUSBoost | 0.7706±0.1215 | 0.2639±0.2681 | 0.3741±0.2508 | 0.6935±0.2721 | **0.8426±0.1081** |
| ANN_TomekLinks | **0.8856±0.0377** | 0.6896±0.0337 | 0.6591±0.0411 | 0.7795±0.045 | 0.8105±0.0513 |
| ANN_SMOTE | **0.9908±0.0073** | 0.7489±0.0102 | 0.7466±0.0096 | 0.9825±0.007 | 0.9764±0.0112 |

จากตารางที่ 4.5 แสดงการเปรียบเทียบค่า Recall ที่แสดงจำนวนคุณลักษณะส่งผลต่อประสิทธิภาพความสามารถในการจดจำ Positive class อย่างค่า Recall การจำแนกประเภทของโมเดลที่ใช้เทคนิคการสุ่มตัวอย่าง Tomek Links เทคนิค SMOTE และเทคนิค RUSBoostClassifier ในการจัดความสมดุลของชุดข้อมูลผ่านการเรียนรู้ของเครื่องโดยใช้อัลกอริทึม Decision Trees, Naïve Bayes, k-Nearest Neighbors, Support Vector Machines และ Artificial Neural Networks

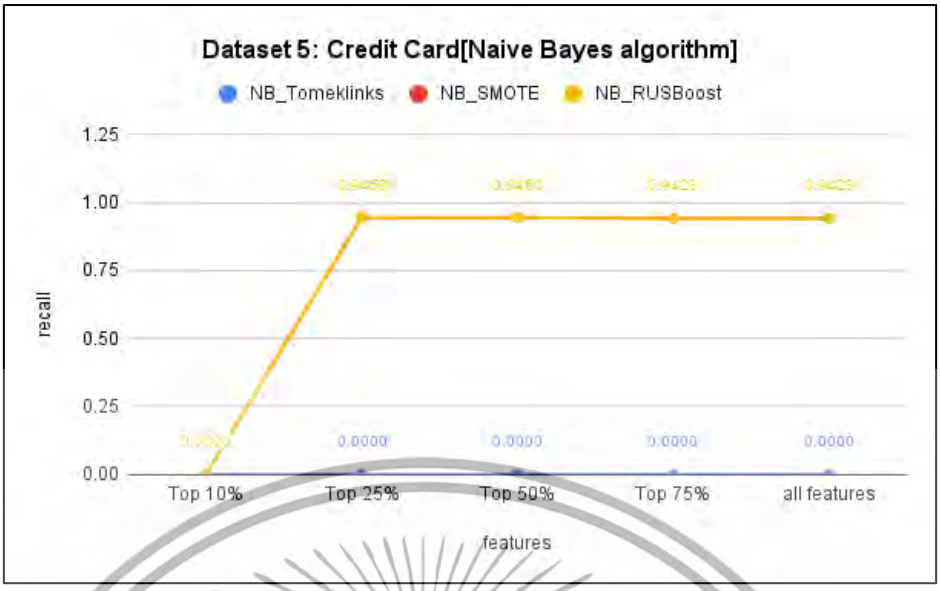



รูปที่ 4.25 กราฟเส้นแสดงค่า Recall ในชุดข้อมูล Credit Card



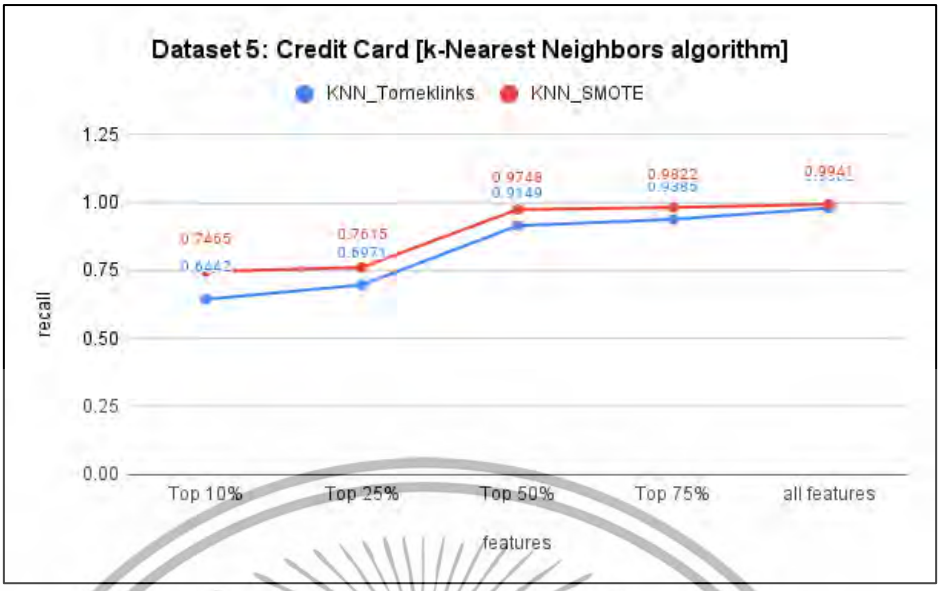

รูปที่ 4.26 กราฟเส้นแสดงค่า Recall ในชุดข้อมูล Credit Card ของอัลกอริทึม Decision Trees

จากรูปที่ 4.26 ชุดข้อมูล Credit Card ทดลองบนอัลกอริทึม Decision Trees พบว่าเทคนิค Tomek Links เทคนิค SMOTE และเทคนิค RUSBoostClassifier จำนวนคุณลักษณะที่ลดลงส่งผลต่อประสิทธิภาพการจำแนกของโมเดลมีค่า Recall ลดลงตามลำดับ เทคนิค RUSBoostClassifier ทำงานกับอัลกอริทึม Decision Trees ได้อย่างมีประสิทธิภาพดีกว่าเมื่อเทียบกับเทคนิคอื่น



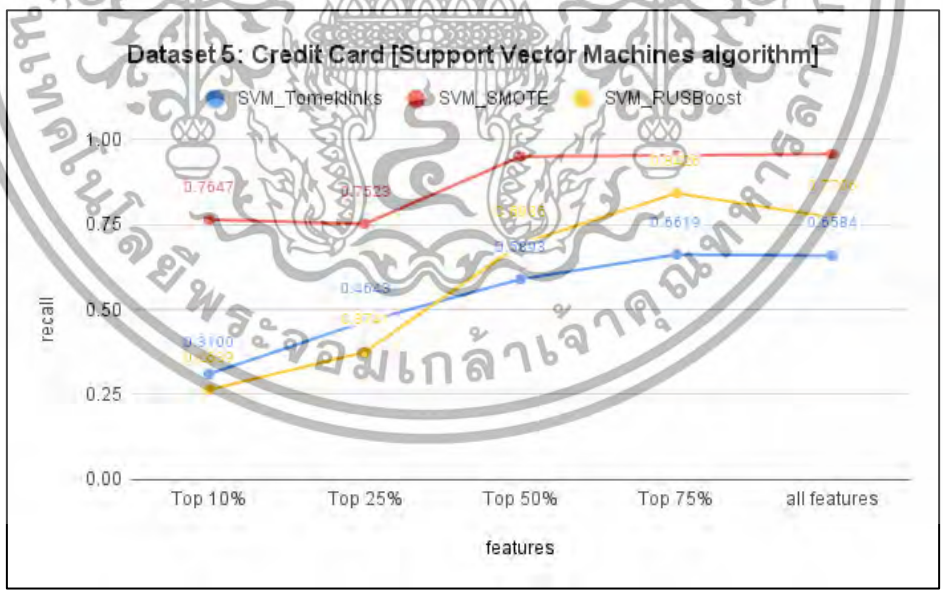

**รูปที่ 4.27 กราฟเส้นแสดงค่า Recall ในชุดข้อมูล Credit Card ของอัลกอริทึม Naïve Bayes**

จากรูปที่ 4.27 ชุดข้อมูล Credit Card ทดลองบนอัลกอริทึม Naïve Bayes พบว่าเทคนิค Tomek Links จำนวนคุณลักษณะ ไม่ส่งผลต่อการจำแนกข้อมูลของโมเดลนี้เนื่องจากเปอร์เซ็นต์ Recall ในการจำแนก Positive class หรือกลุ่ม Minority class มีประสิทธิภาพการที่ต่ำมาก ในทางตรงกันข้ามเทคนิค SMOTE และเทคนิค RUSBoostClassifier จำนวนคุณลักษณะที่เกี่ยวข้องที่สุดในระดับ 10 เปอร์เซ็นต์ หรือจำนวนคุณลักษณะที่น้อยเกินไป ส่งผลต่อประสิทธิภาพในการจำแนกประเภทของโมเดล ในเปอร์เซ็นต์ระดับอื่นกลับไม่ส่งผลต่อประสิทธิภาพในการจำแนกที่แตกต่างกันมาก



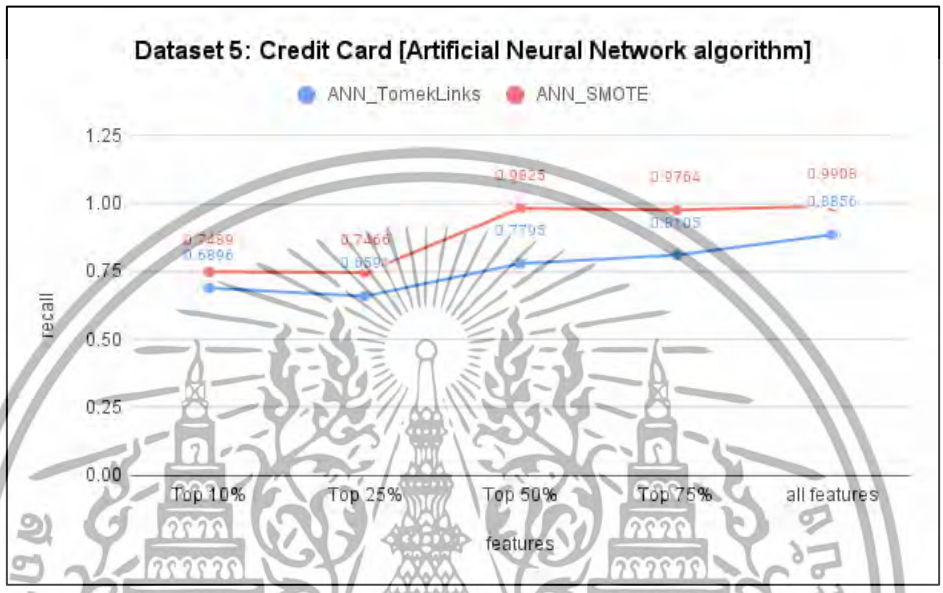

**รูปที่ 4.28 กราฟเส้นแสดงค่า Recall ในชุดข้อมูล Credit Card ของอัลกอริทึม k-Nearest Neighbors**

จากรูปที่ 4.28 ชุดข้อมูล Credit Card ทดลองบนอัลกอริทึม k-Nearest Neighbors พบว่า เทคนิค Tomek Links และเทคนิค SMOTE จำนวนคุณลักษณะที่ลดลงส่งผลต่อประสิทธิภาพการจำแนกของโมเดลมีค่า Recall ลดลงตามลำดับ




**รูปที่ 4.29 กราฟเส้นแสดงค่า Recall ในชุดข้อมูล Credit Card ของอัลกอริทึม Support Vector Machines**

จากรูปที่ 4.29 ชุดข้อมูล Credit Card ทดลองบนอัลกอริทึม Support Vector Machines พบว่าเทคนิค Tomek Links เทคนิค SMOTE และเทคนิค RUSBoostClassifier ทั้ง 3 เทคนิคจำนวนคุณลักษณะส่งผลต่อประสิทธิภาพในการจำแนกประเภทของโมเดล จำนวนคุณลักษณะที่ลดลงส่งผลให้ประสิทธิภาพในการการจำแนกประเภทของโมเดลมีค่า Recall ลดลงตามลำดับ




**รูปที่ 4.30 กราฟเส้นแสดงค่า Recall ในชุดข้อมูล Credit Card ของอัลกอริทึม Artificial Neural Networks**

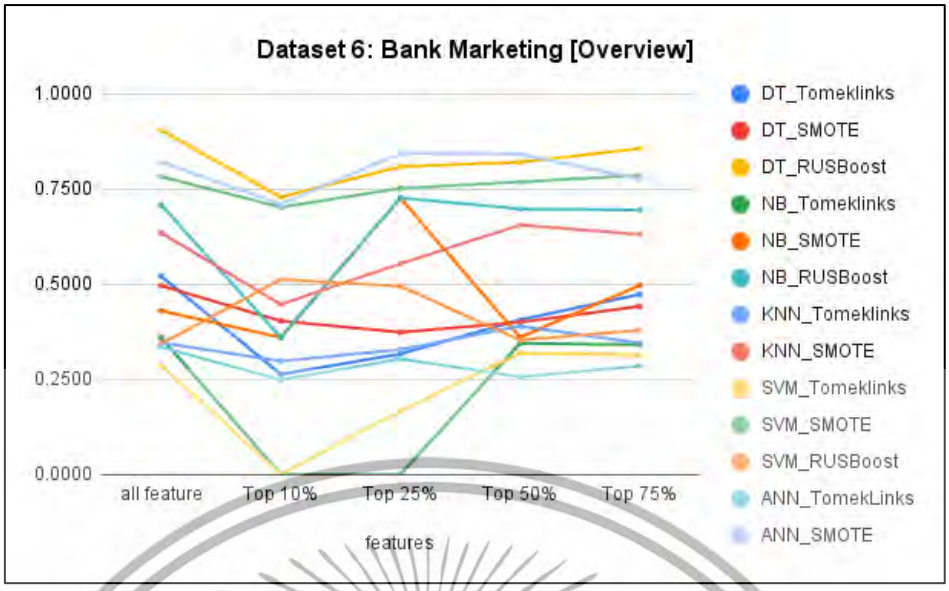
จากรูปที่ 4.30 ชุดข้อมูล Credit Card ทดลองบนอัลกอริทึม Artificial Neural Networks พบว่าเทคนิค Tomek Links และเทคนิค SMOTE จำนวนคุณลักษณะที่ลดลงส่งผลต่อประสิทธิภาพการจำแนกของโมเดลมีค่า Recall ลดลงตามลำดับ เทคนิค SMOTE ทำงานได้อย่างมีประสิทธิภาพในการทำงานร่วมกับอัลกอริทึม Artificial Neural Networks

### 4.1.6 ชุดข้อมูล Bank Marketing

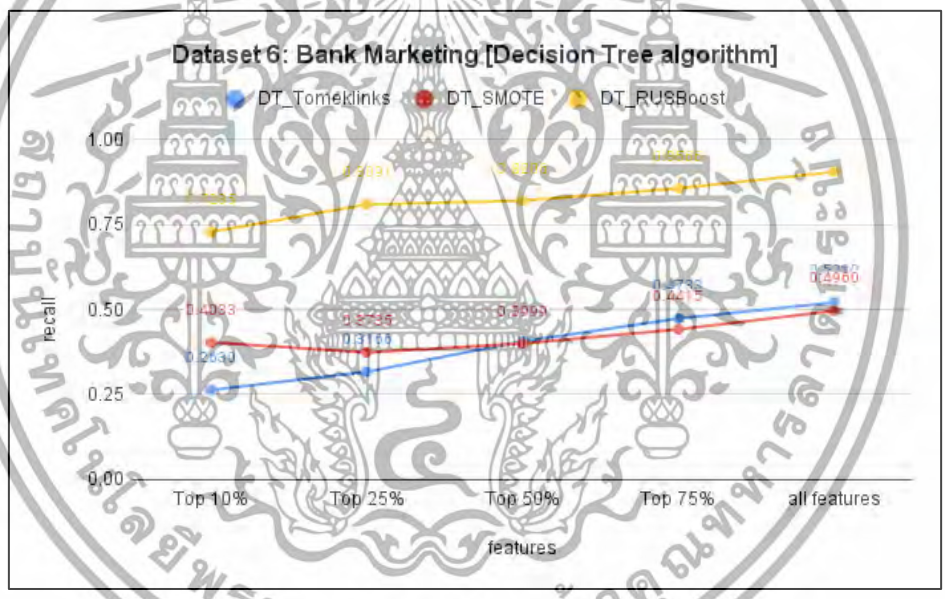
ตารางที่ 4.6 ค่า Recall ของชุดข้อมูลที่ไม่สมดุลชุดที่ 6 Bank Marketing

| Model | all features | Top 10% | Top 25% | Top 50% | Top 75% |
|---|---|---|---|---|---|
| DT_Tomeklinks | **0.521±0.0288** | 0.263±0.0099 | 0.3166±0.0136 | 0.4063±0.0179 | 0.4733±0.0216 |
| DT_SMOTE | **0.496±0.0212** | 0.4033±0.0176 | 0.3735±0.0144 | 0.3999±0.0219 | 0.4415±0.025 |
| DT_RUSBoost | **0.905±0.0158** | 0.7285±0.0295 | 0.8091±0.0253 | 0.8206±0.0152 | 0.8566±0.0143 |
| NB_Tomeklinks | **0.359±0.0866** | 0±0 | 0±0 | 0.3444±0.0174 | 0.341±0.0159 |
| NB_SMOTE | 0.43±0.0231 | 0.3601±0.0186 | **0.726±0.0238** | 0.3603±0.0186 | 0.4973±0.0274 |
| NB_RUSBoost | 0.708±0.0244 | 0.3601±0.0186 | **0.7267±0.0233** | 0.6978±0.0244 | 0.6945±0.0253 |
| KNN_Tomeklinks | 0.346±0.0135 | 0.2976±0.0109 | 0.3279±0.0128 | **0.3896±0.0162** | 0.3452±0.014 |
| KNN_SMOTE | 0.635±0.0184 | 0.4467±0.0173 | 0.5538±0.0198 | **0.6558±0.0153** | 0.6307±0.0216 |
| SVM_Tomeklinks | 0.287±0.033 | 0±0 | 0.1664±0.0148 | **0.3185±0.0105** | 0.3132±0.0121 |
| SVM_SMOTE | 0.783±0.0353 | 0.7022±0.0127 | 0.752±0.0218 | 0.7688±0.0195 | **0.7861±0.0226** |
| SVM_RUSBoost | 0.343±0.1437 | **0.5124±0.1157** | 0.4947±0.1035 | 0.353±0.0968 | 0.379±0.1318 |
| ANN_TomekLinks | 0.3345±0.1719 | 0.2485±0.1293 | 0.3034±0.2168 | 0.2561±0.1415 | 0.2844±0.1441 |
| ANN_SMOTE | 0.8209±0.0693 | 0.7102±0.0527 | **0.8445±0.0351** | 0.8417±0.0609 | 0.7765±0.0926 |

จากตารางที่ 4.6 แสดงการเปรียบเทียบค่า Recall ที่แสดงจำนวนคุณลักษณะส่งผลต่อประสิทธิภาพความสามารถในการจดจำ Positive class อย่างค่า Recall การจำแนกประเภทของโมเดลที่ใช้เทคนิคการสุ่มตัวอย่าง Tomek Links เทคนิค SMOTE และเทคนิค RUSBoostClassifier ในการจัดความสมดุลของชุดข้อมูลผ่านการเรียนรู้ของเครื่องโดยใช้อัลกอริทึม Decision Trees, Naïve Bayes, k-Nearest Neighbors, Support Vector Machines และ Artificial Neural Networks

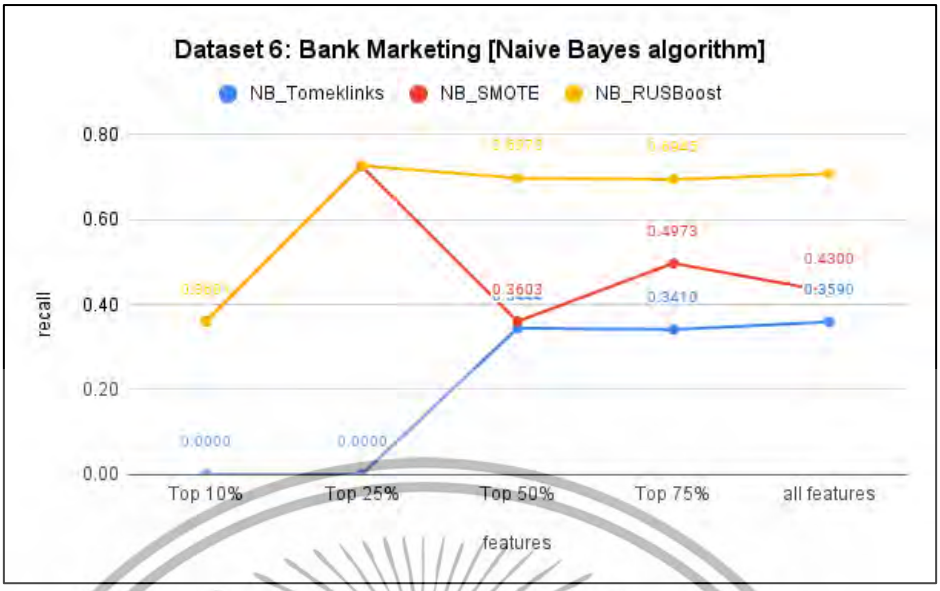



**รูปที่ 4.31 กราฟเส้นแสดงค่า Recall ในชุดข้อมูล Bank Marketing**



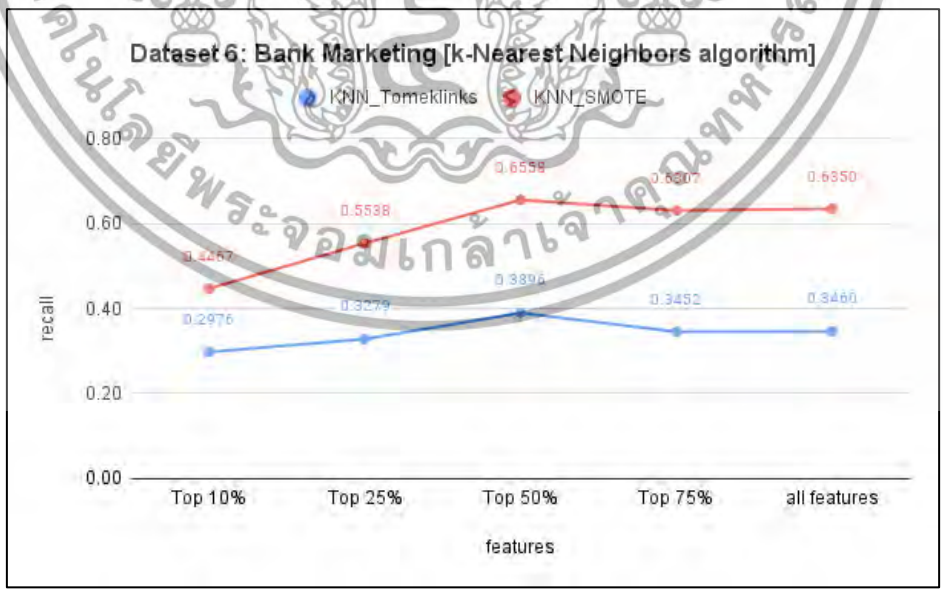

**รูปที่ 4.32 กราฟเส้นแสดงค่า Recall ในชุดข้อมูล Bank Marketing ของอัลกอริทึม Decision Trees**

จากรูปที่ 4.32 ชุดข้อมูล Bank Marketing ทดลองบนอัลกอริทึม Decision Trees พบว่า เทคนิค Tomek Links จำนวน คุณลักษณะที่ลดลง ส่งผลต่อโมเดลมีประสิทธิภาพในการจำแนกประสิทธิภาพค่า Recall ที่ต่ำแปรผันตรงตามกัน ส่วนเทคนิค SMOTE จำนวนคุณลักษณะที่ลดลงส่งผลต่อโมเดลมีประสิทธิภาพการจำแนกประเภทของโมเดลมีค่า Recall ลดลงตามลำดับและเทคนิค RUSBoostClassifier ทำงานร่วมกับอัลกอริทึม Decision Trees ได้อย่างมีประสิทธิภาพ จำนวนคุณลักษณะแปรผันตรงต่อประสิทธิภาพความสามารถในการจดจำ Positive class หรือเรียกว่าค่า Recall



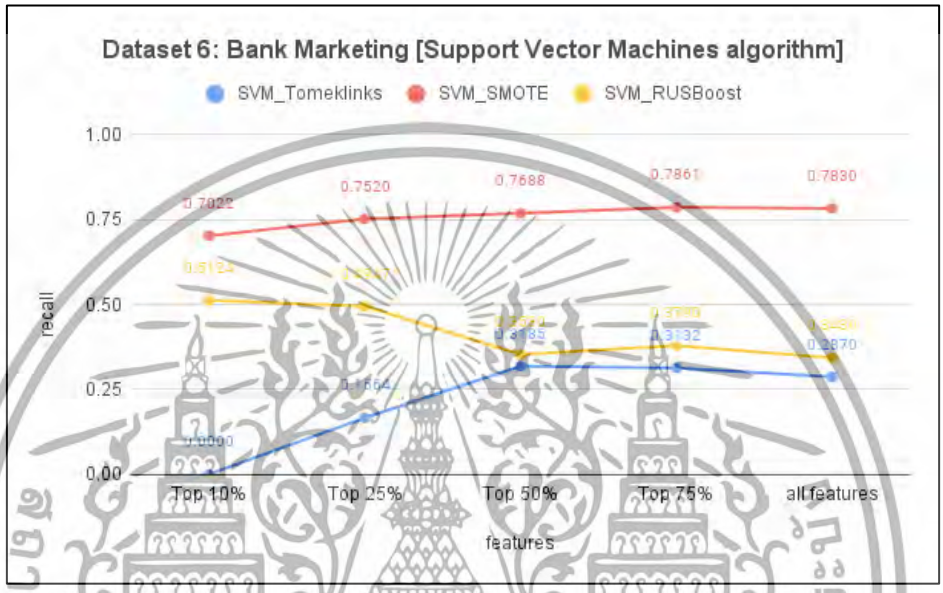

**รูปที่ 4.33 กราฟเส้นแสดงค่า Recall ในชุดข้อมูล Bank Marketing ของอัลกอริทึม Naïve Bayes**

จากรูปที่ 4.33 ชุดข้อมูล Bank Marketing ทดลองบนอัลกอริทึม Naïve Bayes พบว่า เทคนิค Tomek Links จำนวนคุณลักษณะที่ลดลง ส่งผลต่อโมเดลมีประสิทธิภาพในการจำแนกประสิทธิภาพค่า Recall ที่ต่ำ ส่วนเทคนิค SMOTE ความสัมพันธ์ระหว่างคุณลักษณะมีผลต่อการจำแนกข้อมูล และประสิทธิภาพในการจำแนกประเภทของโมเดล และเทคนิค RUSBoostClassifier โมเดลมีประสิทธิภาพค่า Recall ดีที่สุดเมื่อเทียบกันระหว่าง 3 เทคนิค



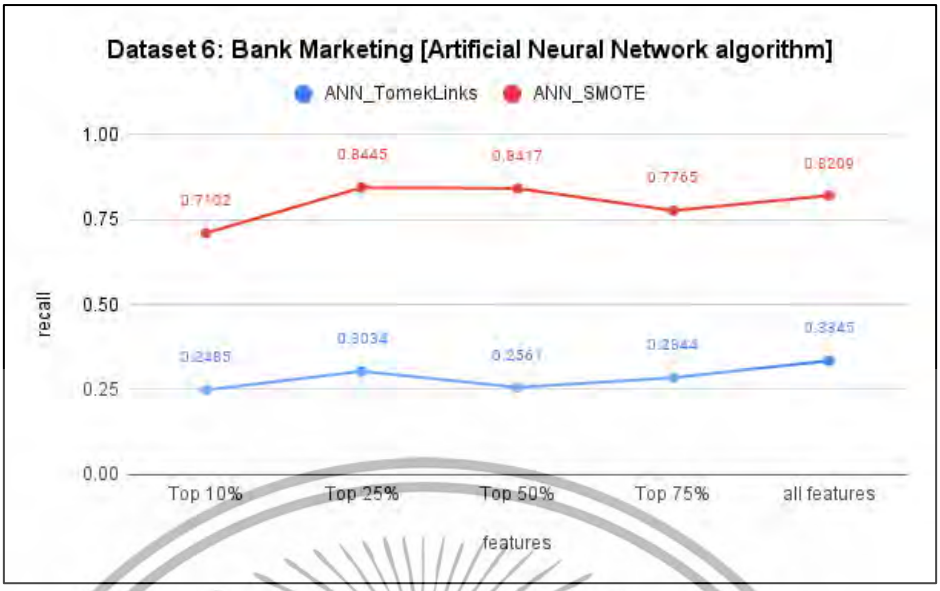

**รูปที่ 4.34 กราฟเส้นแสดงค่า Recall ในชุดข้อมูล Bank Marketing ของอัลกอริทึม k-Nearest Neighbors**

จากรูปที่ 4.34 ชุดข้อมูล Bank Marketing ทดลองบนอัลกอริทึม k-Nearest Neighbors พบว่าเทคนิค Tomek Links จำนวนคุณลักษณะที่ลดลง ส่งผลต่อโมเดลมีประสิทธิภาพในการจำแนกประสิทธิภาพค่า Recall ที่ต่ำ ส่วนเทคนิค SMOTE มีประสิทธิภาพการจำแนกค่า Recall ดีกว่าเมื่อเทียบกับเทคนิคอื่น และจำนวน คุณลักษณะที่ลดลง ส่งผลต่อประสิทธิภาพการจำแนกของโมเดลที่ลดลงแปรผันตรงตามจำนวนคุณลักษณะ




**รูปที่ 4.35 กราฟเส้นแสดงค่า Recall ในชุดข้อมูล Bank Marketing ของอัลกอริทึม Support Vector Machines**

จากรูปที่ 4.35 ชุดข้อมูล Bank Marketing ทดลองบนอัลกอริทึม Support Vector Machines พบว่าเทคนิค Tomek Links จำนวนคุณลักษณะที่ลดลง ส่งผลต่อโมเดลมีประสิทธิภาพในการจำแนกประสิทธิภาพค่า Recall ที่ลดลงตามลำดับ ส่วนเทคนิค SMOTE มีประสิทธิภาพการจำแนกค่า Recall ที่ดีกว่า เมื่อเทียบกับเทคนิคอื่น และจำนวนคุณลักษณะที่ลดลง ส่งผลต่อประสิทธิภาพการจำแนกของโมเดลที่ลดลงแปรผันตรงตามจำนวนคุณลักษณะ และเทคนิค RUSBoostClassifier จำนวนคุณลักษณะที่ลดลงส่งผลให้มีประสิทธิภาพการจำแนกค่า Recall ได้ดีกว่าในการทำงานร่วมกับอัลกอริทึม Support Vector Machines




**รูปที่ 4.36 กราฟเส้นแสดงค่า Recall ในชุดข้อมูล Bank Marketing ของอัลกอริทึม Artificial Neural Networks**

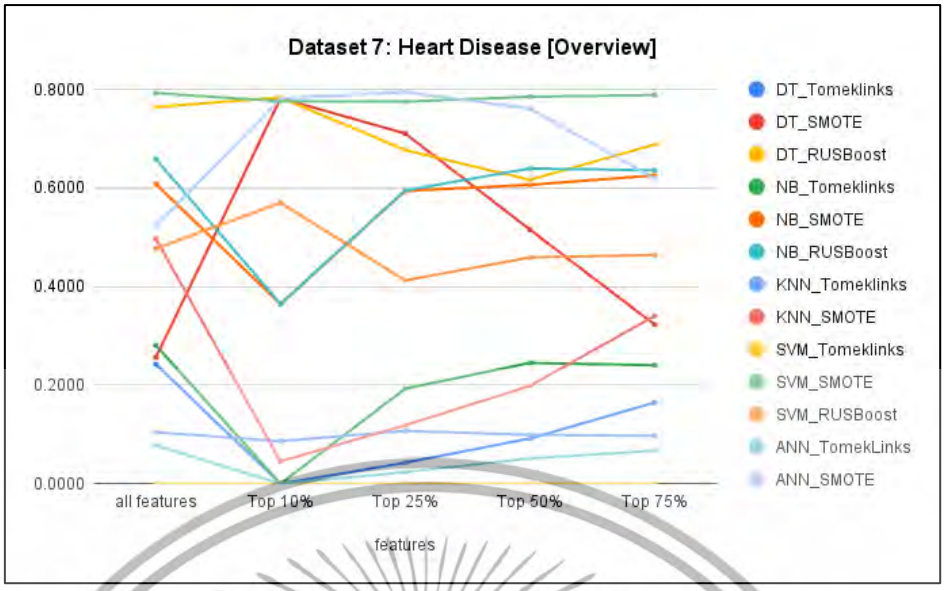
จากรูปที่ 4.36 ชุดข้อมูล Bank Marketing ทดลองบนอัลกอริทึม Artificial Neural Networks พบว่าเทคนิค Tomek Links จำนวนคุณลักษณะที่ลดลง ส่งผลต่อโมเดลมีประสิทธิภาพในการจำแนกประสิทธิภาพค่า Recall ที่ลดลง และเทคนิค SMOTE มีประสิทธิภาพการจำแนกค่า Recall ที่ดีกว่า เมื่อเทียบกับ Tomek Links และในกรณีที่ความสัมพันธ์ระหว่างคุณลักษณะมีความสัมพันธ์กันน้อยส่งผลต่อประสิทธิภาพการจำแนกค่า Recall ที่ลดลง

### 4.1.7 ชุดข้อมูล Heart Disease

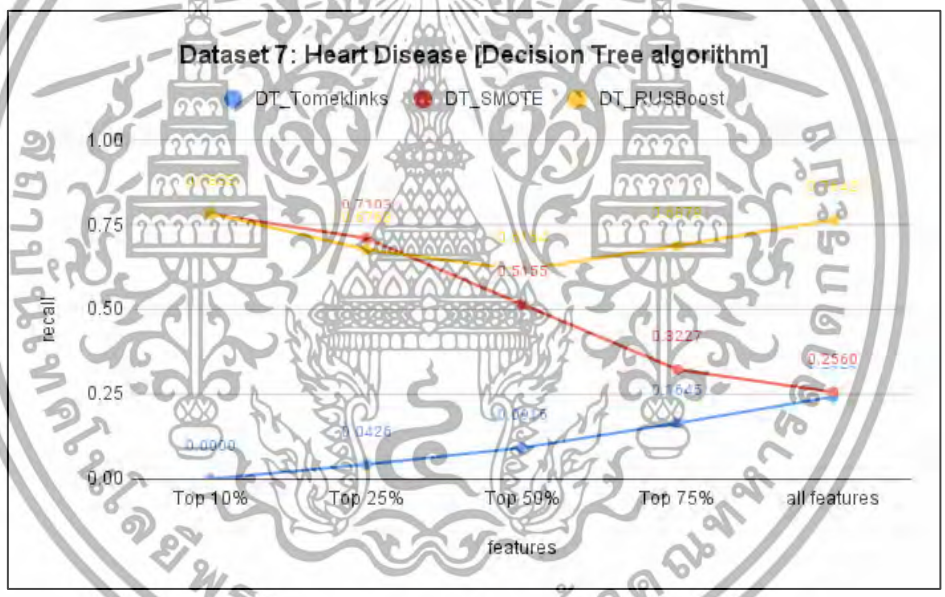
ตารางที่ 4.7 ค่า Recall ของชุดข้อมูลที่ไม่สมดุลชุดที่ 7 Heart Disease

| Model | all features | Top 10% | Top 25% | Top 50% | Top 75% |
|---|---|---|---|---|---|
| DT_Tomeklinks | **0.2424±0.018** | 0±0 | 0.0426±0.0092 | 0.0916±0.0068 | 0.1645±0.0124 |
| DT_SMOTE | 0.256±0.0209 | **0.7819±0.0117** | 0.7103±0.0121 | 0.5155±0.013 | 0.3227±0.0153 |
| DT_RUSBoost | 0.7642±0.0172 | **0.7833±0.0114** | 0.6768±0.1003 | 0.6164±0.0966 | 0.6879±0.0477 |
| NB_Tomeklinks | **0.2807±0.0113** | 0±0 | 0.1928±0.0103 | 0.245±0.0089 | 0.2402±0.0116 |
| NB_SMOTE | 0.6082±0.0135 | 0.3643±0.0115 | 0.5939±0.0157 | 0.6062±0.0117 | **0.6254±0.0136** |
| NB_RUSBoost | **0.6585±0.0171** | 0.3647±0.0113 | 0.5948±0.0161 | 0.6395±0.0245 | 0.6356±0.0159 |
| KNN_Tomeklinks | 0.1043±0.0096 | 0.0864±0.0841 | **0.1071±0.0201** | 0.0991±0.0107 | 0.0973±0.0101 |
| KNN_SMOTE | **0.497±0.0153** | 0.0454±0.0694 | 0.1185±0.0186 | 0.1984±0.0102 | 0.3401±0.0155 |
| SVM_Tomeklinks | 0±0 | 0±0 | 0±0 | 0±0 | 0±0 |
| SVM_SMOTE | **0.7921±0.0116** | 0.7758±0.0107 | 0.775±0.0112 | 0.7847±0.0174 | 0.7885±0.0109 |
| SVM_RUSBoost | 0.4772±0.0796 | **0.5696±0.2676** | 0.4123±0.3114 | 0.459±0.0253 | 0.464±0.0762 |
| ANN_TomekLinks | **0.0777±0.0326** | 0±0 | 0.0235±0.02 | 0.0514±0.013 | 0.0669±0.0135 |
| ANN_SMOTE | 0.5224±0.088 | 0.7812±0.0441 | **0.7945±0.0263** | 0.7609±0.0342 | 0.6191±0.0535 |

จากตารางที่ 4.7 แสดงการเปรียบเทียบค่า Recall ที่แสดงจำนวนคุณลักษณะส่งผลต่อประสิทธิภาพความสามารถในการจดจำ Positive class อย่างค่า Recall การจำแนกประเภทของโมเดลที่ใช้เทคนิคการสุ่มตัวอย่าง Tomek Links เทคนิค SMOTE และเทคนิค RUSBoostClassifier ในการจัดความสมดุลของชุดข้อมูลผ่านการเรียนรู้ของเครื่องโดยใช้อัลกอริทึม Decision Trees, Naïve Bayes, k-Nearest Neighbors, Support Vector Machines และ Artificial Neural Networks

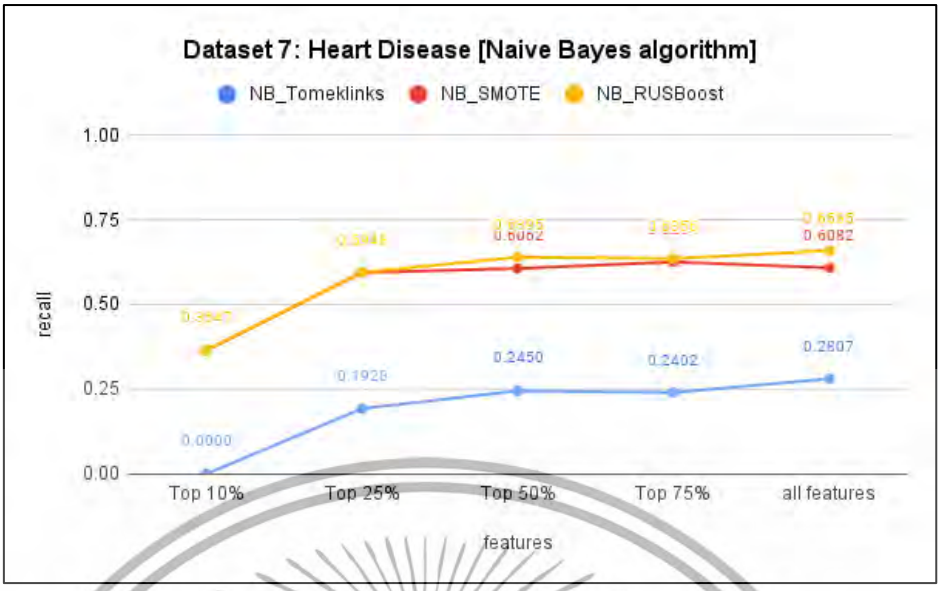



**รูปที่ 4.37 กราฟเส้นแสดงค่า Recall ในชุดข้อมูล Heart Disease**



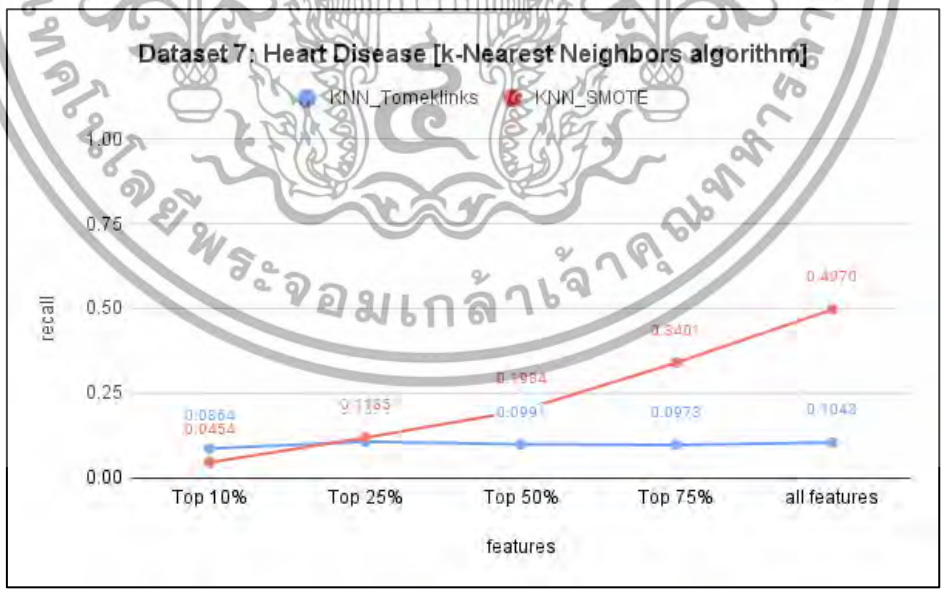

**รูปที่ 4.38 กราฟเส้นแสดงค่า Recall ในชุดข้อมูล Heart Disease ของอัลกอริทึม Decision Trees**

จากรูปที่ 4.38 ชุดข้อมูล Heart Disease ทดลองบนอัลกอริทึม Decision Trees พบว่า เทคนิค Tomek Links จำนวนคุณลักษณะที่ลดลง ประสิทธิภาพในการจำแนกของโมเดลค่า Recall ลดลงตามลำดับ ส่วนเทคนิค SMOTE จำนวนคุณลักษณะที่ลดลง ส่งผลให้โมเดลมีประสิทธิภาพในการจำแนกค่า Recall ที่สูงขึ้น และเทคนิค RUSBoostClassifier จำนวนคุณลักษณะที่ลดลงไม่ส่งผลต่อประสิทธิภาพการจำแนกของโมเดลค่า Recall ขึ้นอยู่กับความสัมพันธ์ระหว่างคุณลักษณะ นอกจากนี้สามารถทำงานกับอัลกอริทึม Decision Trees ได้อย่างมีประสิทธิภาพดีกว่าเทคนิคอื่น



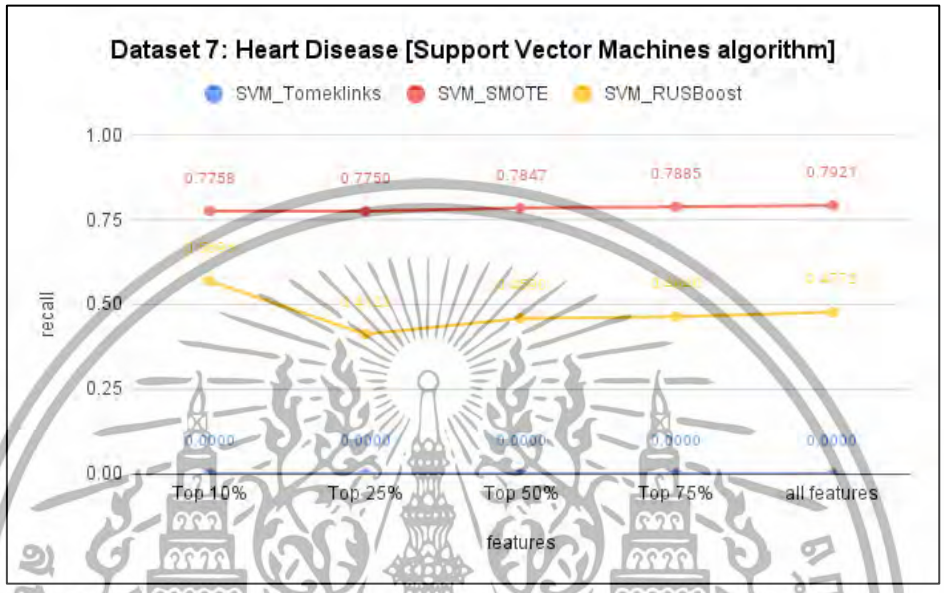

**รูปที่ 4.39 กราฟเส้นแสดงค่า Recall ในชุดข้อมูล Heart Disease ของอัลกอริทึม Naïve Bayes**

จากรูปที่ 4.39 ชุดข้อมูล Heart Disease ทดลองบนอัลกอริทึม Naïve Bayes พบว่าเทคนิค Tomek Links เทคนิค SMOTE และเทคนิค RUSBoostClassifier จำนวนคุณลักษณะที่ลดลง ส่งผลต่อประสิทธิภาพในการจำแนกของโมเดลมีค่า Recall ลดลงตามลำดับ นอกจากนี้เทคนิค SMOTE และเทคนิค RUSBoostClassifier ทำงานกับอัลกอริทึม Naïve Bayes ได้อย่างมีประสิทธิภาพ



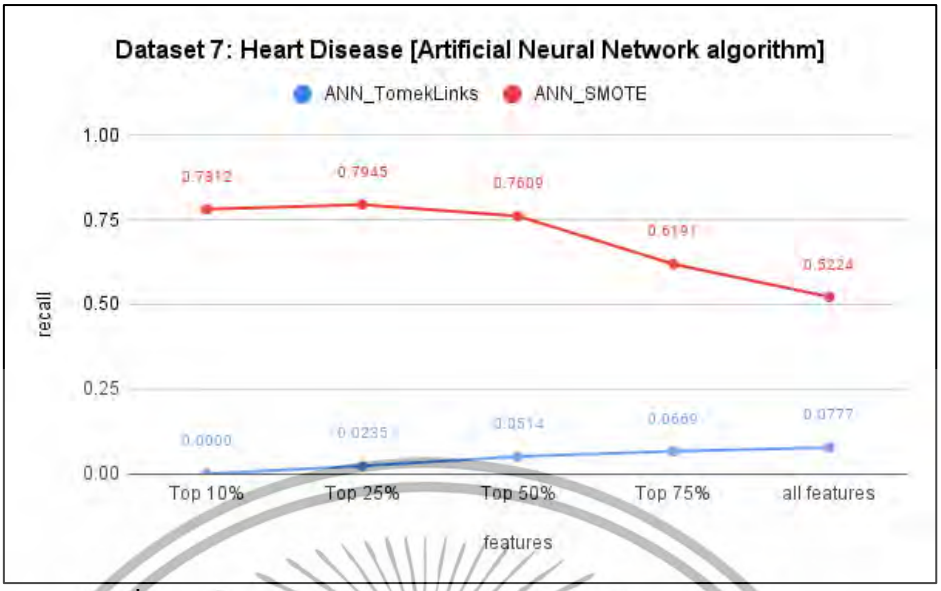

**รูปที่ 4.40 กราฟเส้นแสดงค่า Recall ในชุดข้อมูล Heart Disease ของอัลกอริทึม k-Nearest Neighbors**

จากรูปที่ 4.40 ชุดข้อมูล Heart Disease ทดลองบนอัลกอริทึม k-Nearest Neighbors พบว่าเทคนิค Tomek Links และเทคนิค SMOTE จำนวนคุณลักษณะที่ลดลง ส่งผลต่อประสิทธิภาพในการจำแนกของโมเดลมีค่า Recall ลดลงตามลำดับ เทคนิค SMOTE ทำงานร่วมกับอัลกอริทึม k-Nearest Neighborsได้อย่างมีประสิทธิภาพ




**รูปที่ 4.41 กราฟเส้นแสดงค่า Recall ในชุดข้อมูล Heart Disease ของอัลกอริทึม Support Vector Machines**

จากรูปที่ 4.41 ชุดข้อมูล Heart Disease ทดลองบนอัลกอริทึม Support Vector Machines พบว่าเทคนิค Tomek Links ทำงานร่วมกับอัลกอริทึมมีประสิทธิภาพในการจำแนกประเภทของโมเดลมีค่า Recall ต่ำ เทคนิคนี้ไม่เหมาะที่จะทำงานร่วมกับอัลกอริทึมนี้ ในทางตรงกันข้ามเทคนิค SMOTE กลับทำงานร่วมกับอัลกอริทึม Support Vector Machines ได้อย่างมีประสิทธิภาพและจำนวนคุณลักษณะไม่ส่งผลต่อประสิทธิภาพในการจำแนกของโมเดล และเทคนิค RUSBoostClassifier จำนวนคุณลักษณะที่ลดลงส่งผลให้มีประสิทธิภาพค่า Recall ที่เพิ่มขึ้น




**รูปที่ 4.42 กราฟเส้นแสดงค่า Recall ในชุดข้อมูล Heart Disease ของอัลกอริทึม Artificial Neural Networks**

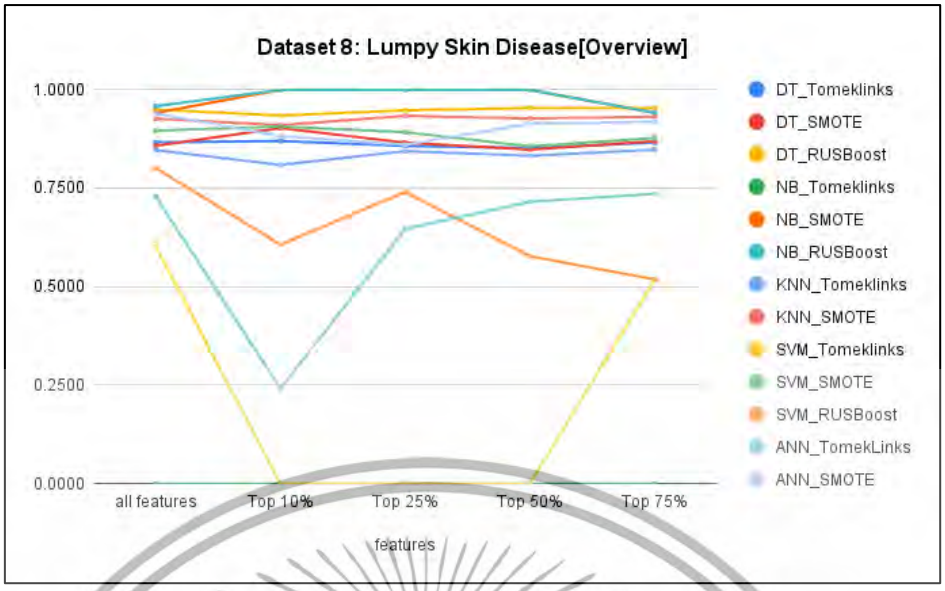
จากรูปที่ 4.42 ชุดข้อมูล Heart Disease ทดลองบนอัลกอริทึม Artificial Neural Networks พบว่าเทคนิค Tomek Links จำนวนคุณลักษณะที่ลดลง ส่งผลต่อประสิทธิภาพในการจำแนกของโมเดลมีค่า Recall ลดลงตามลำดับ ในทางตรงกันข้ามเทคนิค SMOTE จำนวนคุณลักษณะที่ลดลง กลับส่งผลให้ประสิทธิภาพในการจำแนกของโมเดลมีค่า Recall ที่เพิ่มขึ้น

### 4.1.8 ชุดข้อมูล Lumpy Skin Disease

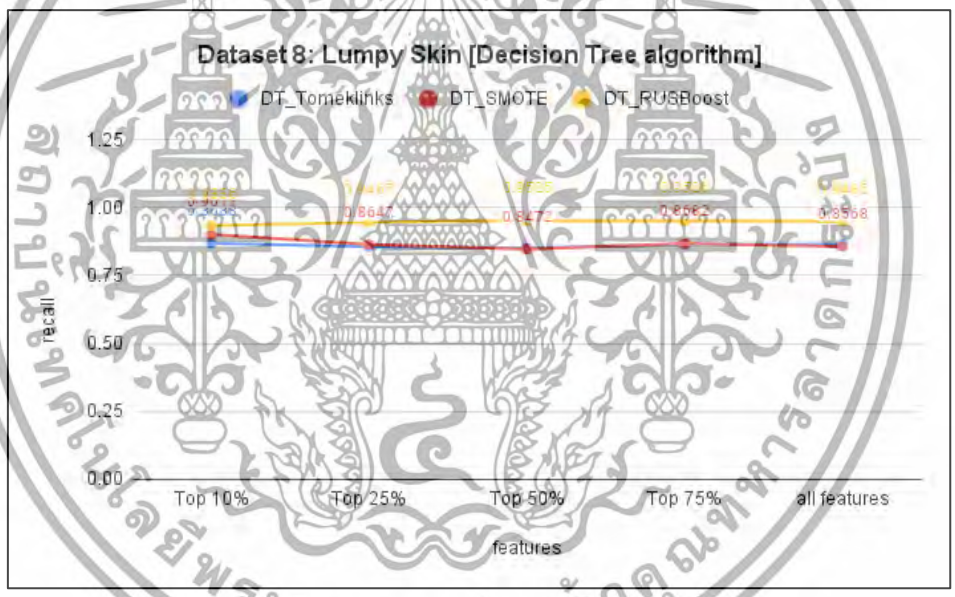
ตารางที่ 4.8 ค่า Recall ของชุดข้อมูลที่ไม่สมดุลชุดที่ 8 Lumpy Skin Disease

| Model | all features | Top 10% | Top 25% | Top 50% | Top 75% |
|---|---|---|---|---|---|
| DT_Tomeklinks | 0.8656±0.0181 | **0.8688±0.015** | 0.8564±0.024 | 0.8511±0.0224 | 0.8649±0.0205 |
| DT_SMOTE | 0.8568±0.0151 | **0.9017±0.0141** | 0.8647±0.0192 | 0.8472±0.0145 | 0.8682±0.0226 |
| DT_RUSBoost | 0.9485±0.0121 | 0.9336±0.0142 | 0.9467±0.013 | 0.9525±0.01 | **0.9526±0.0126** |
| NB_Tomeklinks | 0±0 | 0±0 | 0±0 | 0±0 | 0±0 |
| NB_SMOTE | 0.9401±0.0176 | **0.9985±0.0024** | **0.9985±0.0024** | **0.9985±0.0024** | 0.9405±0.0111 |
| NB_RUSBoost | 0.9573±0.019 | **0.9985±0.0024** | **0.9985±0.0024** | **0.9985±0.0024** | 0.9405±0.0111 |
| KNN_Tomeklinks | 0.8456±0.0209 | 0.8082±0.0296 | 0.8431±0.0192 | 0.8315±0.0233 | **0.8467±0.0228** |
| KNN_SMOTE | 0.9255±0.0118 | 0.909±0.0186 | **0.9326±0.0151** | 0.9258±0.0098 | 0.9305±0.0119 |
| SVM_Tomeklinks | **0.6009±0.0369** | 0±0 | 0±0 | 0±0 | 0.5158±0.0407 |
| SVM_SMOTE | 0.8951±0.0181 | **0.9056±0.0238** | 0.8913±0.0275 | 0.8554±0.019 | 0.8765±0.0161 |
| SVM_RUSBoost | **0.8001±0.1976** | 0.6061±0.2315 | 0.7399±0.2952 | 0.5767±0.1651 | 0.5181±0.2256 |
| ANN_TomekLinks | 0.728±0.0444 | 0.241±0.0469 | 0.6458±0.0487 | 0.715±0.0369 | **0.7349±0.0422** |
| ANN_SMOTE | **0.9388±0.0293** | 0.8811±0.033 | 0.8595±0.0269 | 0.9125±0.0213 | 0.918±0.0265 |

จากตารางที่ 4.8 แสดงการเปรียบเทียบค่า Recall ที่แสดงจำนวนคุณลักษณะส่งผลต่อประสิทธิภาพความสามารถในการจดจำ Positive class อย่างค่า Recall การจำแนกประเภทของโมเดลที่ใช้เทคนิคการสุ่มตัวอย่าง Tomek Links เทคนิค SMOTE และเทคนิค RUSBoostClassifier ในการจัดความสมดุลของชุดข้อมูลผ่านการเรียนรู้ของเครื่องโดยใช้อัลกอริทึม Decision Trees, Naïve Bayes, k-Nearest Neighbors, Support Vector Machines และ Artificial Neural Networks

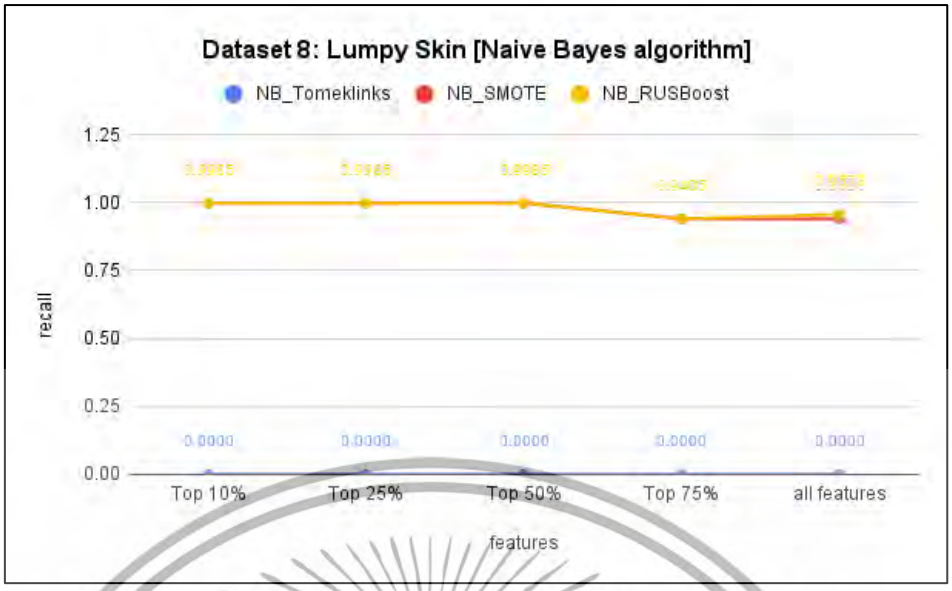



**รูปที่ 4.43 กราฟเส้นแสดงค่า Recall ในชุดข้อมูล Lumpy Skin Disease**



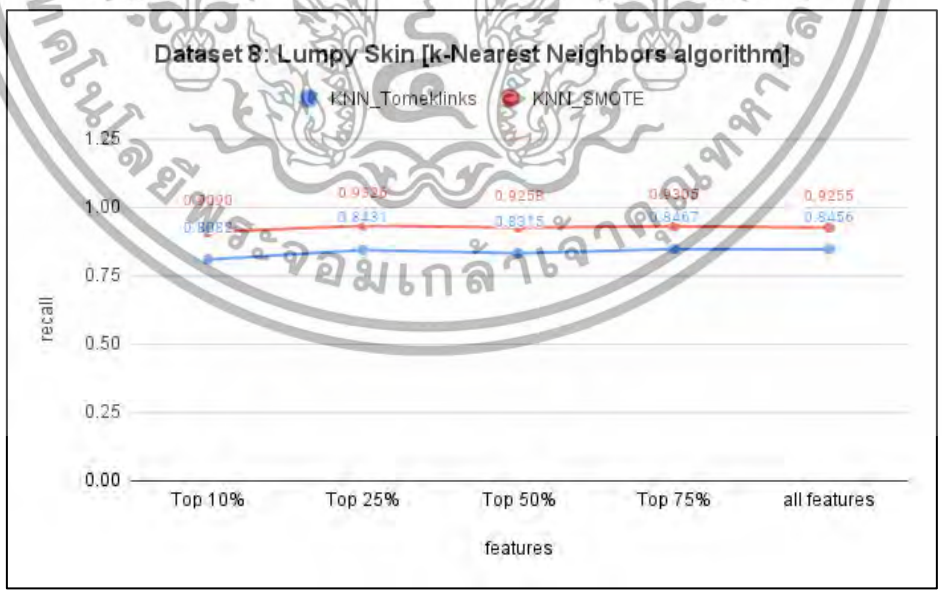

**รูปที่ 4.44 กราฟเส้นแสดงค่า Recall ในชุดข้อมูล Lumpy Skin Disease ของอัลกอริทึม Decision Trees**

จากรูปที่ 4.44 ชุดข้อมูล Lumpy Skin Disease ทดลองบนอัลกอริทึม Decision Trees พบว่าเทคนิค Tomek Links เทคนิค SMOTE และ เทคนิค RUSBoostClassifier จำนวนคุณลักษณะในชุดข้อมูลชุดนี้เป็นไม่ส่งผลต่อการจำแนกประเภทของโมเดลมีค่า Recall ที่แตกต่างกันมาก และนอกจากนี้ยังพบว่าโมเดลที่ใช้เทคนิคการเพิ่มประสิทธิภาพโมเดล เทคนิค RUSBoostClassifier มีประสิทธิภาพดีกว่าเมื่อเทียบกับเทคนิคอื่น



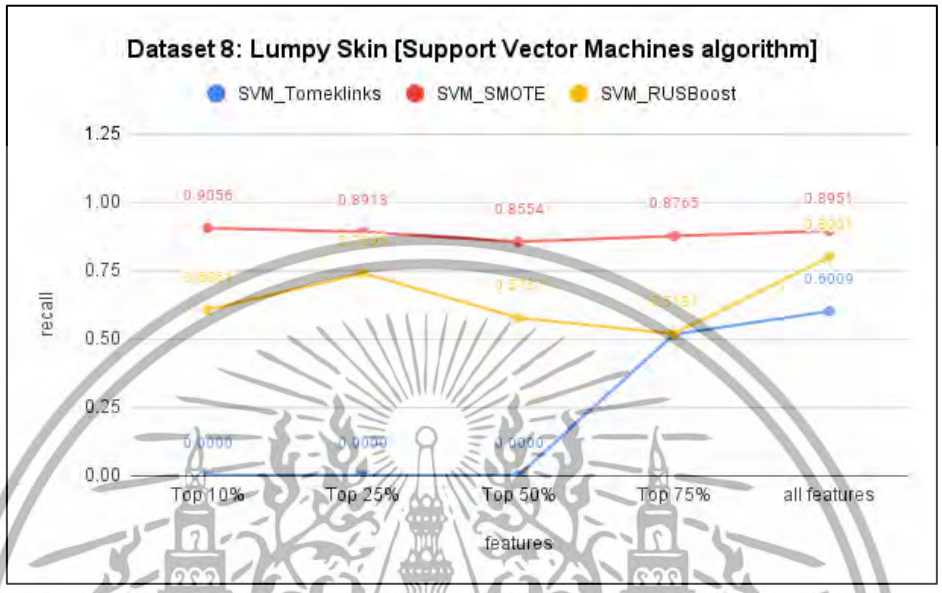

**รูปที่ 4.45 กราฟเส้นแสดงค่า Recall ในชุดข้อมูล Lumpy Skin Disease ของอัลกอริทึม Naïve Bayes**

จากรูปที่ 4.45 ชุดข้อมูล Lumpy Skin Disease ทดลองบนอัลกอริทึม Naïve Bayes พบว่า เทคนิค SMOTE และเทคนิค RUSBoostClassifier จำนวนคุณลักษณะมีผลต่อการจำแนกประเภทของโมเดล จำนวนคุณลักษณะที่ลดลงมีประสิทธิภาพในการจำแนกประเภทโมเดลมีค่า Recall ที่เพิ่มขึ้น ส่วนเทคนิค Tomek Links ไม่สามารถทำงานร่วมกับอัลกอริทึมได้อย่างมีประสิทธิภาพ ค่า Recall มีค่าเท่ากับ 0



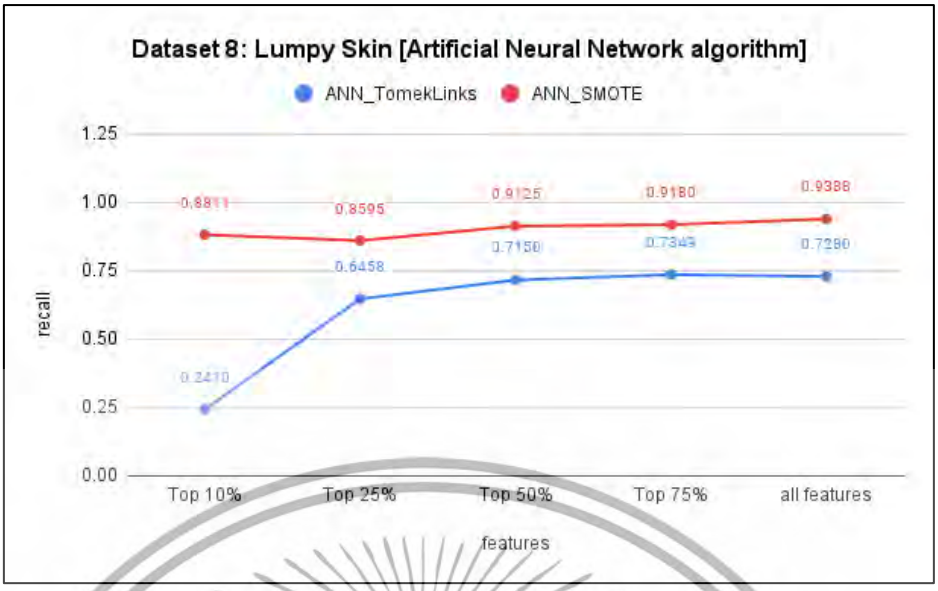

**รูปที่ 4.46 กราฟเส้นแสดงค่า Recall ในชุดข้อมูล Lumpy Skin Disease ของอัลกอริทึม k-Nearest Neighbors**

จากรูปที่ 4.46 ชุดข้อมูล Lumpy Skin Disease ทดลองบนอัลกอริทึม k-Nearest Neighbors พบว่าเทคนิค Tomek Links และเทคนิค SMOTE จำนวนคุณลักษณะไม่ได้ส่งผลต่อประสิทธิภาพในการจำแนกของโมเดลมีค่า Recall ที่แตกต่างกันมากในชุดข้อมูลนี้




**รูปที่ 4.47 กราฟเส้นแสดงค่า Recall ในชุดข้อมูล Lumpy Skin Disease ของอัลกอริทึม Support Vector Machines**

จากรูปที่ 4.47 ชุดข้อมูล Lumpy Skin Disease ทดลองบนอัลกอริทึม Support Vector Machines พบว่าเทคนิค Tomek Links ทำงานร่วมกับอัลกอริทึม Support Vector Machines ประเมินวัดประสิทธิภาพขึ้นอยู่กับจำนวน คุณลักษณะจำนวนคุณลักษณะที่ลดลง ประสิทธิภาพการจำแนกของโมเดลลดลงตามลำดับ เทคนิค SMOTE ทำงานร่วมกับอัลกอริทึม Support Vector Machines ได้อย่างมีประสิทธิภาพ และจำนวนคุณลักษณะที่ลดลง ส่งผลต่อประสิทธิภาพการจำแนกประเภทของโมเดลมีค่า Recall ที่เพิ่มขึ้น และเทคนิค RUSBoostClassifier จำนวนคุณลักษณะไม่ได้ส่งผลต่อประสิทธิภาพโมเดล แต่เป็นความสัมพันธ์ระหว่างคุณลักษณะที่ส่งผลต่อการจำแนกของโมเดล




**รูปที่ 4.48 กราฟเส้นแสดงค่า Recall ในชุดข้อมูล Lumpy Skin Disease ของอัลกอริทึม Artificial Neural Networks**

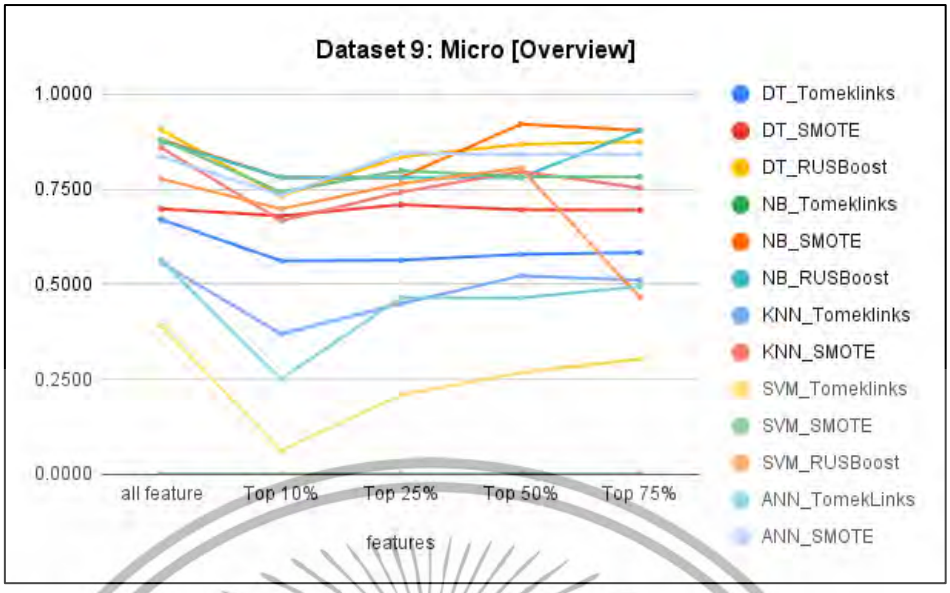
จากรูปที่ 4.48 ชุดข้อมูล Lumpy Skin Disease ทดลองบนอัลกอริทึม Artificial Neural Networks พบว่าเทคนิค Tomek Links จำนวนคุณลักษณะส่งผลต่อประสิทธิภาพในการจำแนกประเภทโมเดล Artificial Neural Networks เทคนิค SMOTE ทำงานร่วมกับอัลกอริทึม Artificial Neural Networks ได้อย่างมีประสิทธิภาพ

### 4.1.9 ชุดข้อมูล Microcalcification Classification

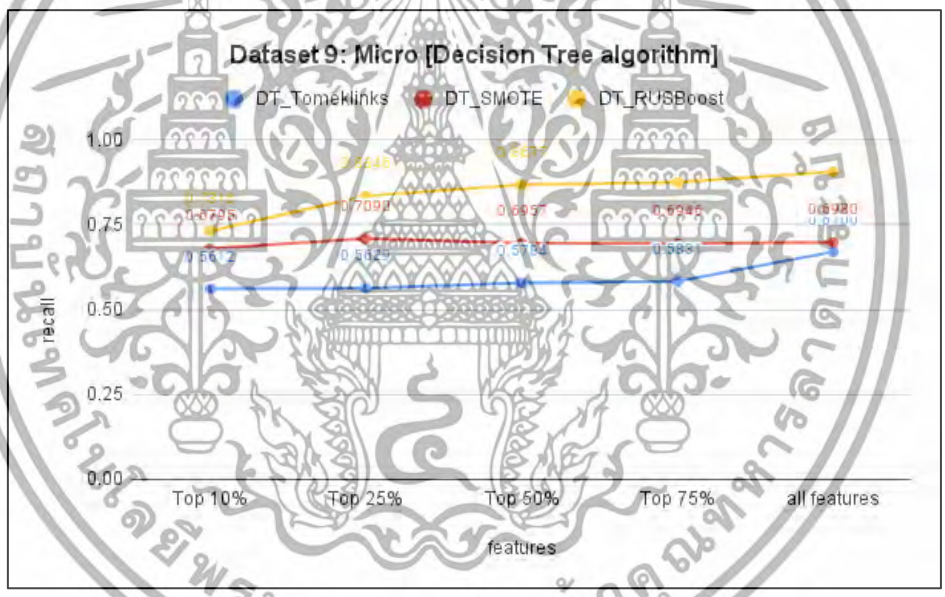
ตารางที่ 4.9 ค่า Recall ของชุดข้อมูลที่ไม่สมดุลชุดที่ 9 Microcalcification classification

| Model | all features | Top 10% | Top 25% | Top 50% | Top 75% |
|---|---|---|---|---|---|
| DT_Tomeklinks | **0.67±0.1544** | 0.5612±0.1017 | 0.5629±0.113 | 0.5784±0.0875 | 0.5831±0.0768 |
| DT_SMOTE | 0.698±0.1179 | 0.6795±0.1132 | **0.709±0.1168** | 0.6957±0.1061 | 0.6946±0.1303 |
| DT_RUSBoost | **0.906±0.056** | 0.7316±0.1175 | 0.8346±0.0937 | 0.8677±0.0831 | 0.8749±0.0683 |
| NB_Tomeklinks | 0±0.0199 | 0±0 | 0±0 | 0±0 | 0±0 |
| NB_SMOTE | 0.88±0 | 0.7801±0.1077 | 0.7801±0.1077 | **0.9208±0.065** | 0.9045±0.0681 |
| NB_RUSBoost | 0.876±0 | 0.7801±0.1077 | 0.7801±0.1077 | 0.7801±0.1077 | **0.9045±0.0681** |
| KNN_Tomeklinks | **0.556±0.1248** | 0.3685±0.1021 | 0.4483±0.1246 | 0.5216±0.1054 | 0.5104±0.1383 |
| KNN_SMOTE | **0.859±0** | 0.6669±0.1896 | 0.7422±0.1347 | 0.7968±0.0898 | 0.7527±0.0881 |
| SVM_Tomeklinks | **0.39±0** | 0.062±0.038 | 0.2087±0.0391 | 0.2666±0.0508 | 0.3033±0.0589 |
| SVM_SMOTE | **0.88±0.0898** | 0.7422±0.1044 | 0.7985±0.1006 | 0.7825±0.0958 | 0.7825±0.0958 |
| SVM_RUSBoost | 0.777±0.1544 | 0.6978±0.1408 | 0.7642±0.1588 | **0.8057±0.1129** | 0.4642±0.2771 |
| ANN_TomekLinks | **0.5631±0.1179** | 0.2507±0.121 | 0.4641±0.1312 | 0.4638±0.1213 | 0.4936±0.1481 |
| ANN_SMOTE | 0.8345±0.0732 | 0.7344±0.1132 | **0.8453±0.0932** | 0.8414±0.0956 | 0.8414±0.1249 |

จากตารางที่ 4.9 แสดงการเปรียบเทียบค่า Recall ที่แสดงจำนวนคุณลักษณะส่งผลต่อประสิทธิภาพความสามารถในการจดจำ Positive class อย่างค่า Recall การจำแนกประเภทของโมเดลที่ใช้เทคนิคการสุ่มตัวอย่าง Tomek Links เทคนิค SMOTE และเทคนิค RUSBoostClassifier ในการจัดความสมดุลของชุดข้อมูลผ่านการเรียนรู้ของเครื่องโดยใช้อัลกอริทึม Decision Trees, Naïve Bayes, k-Nearest Neighbors, Support Vector Machines และ Artificial Neural Networks

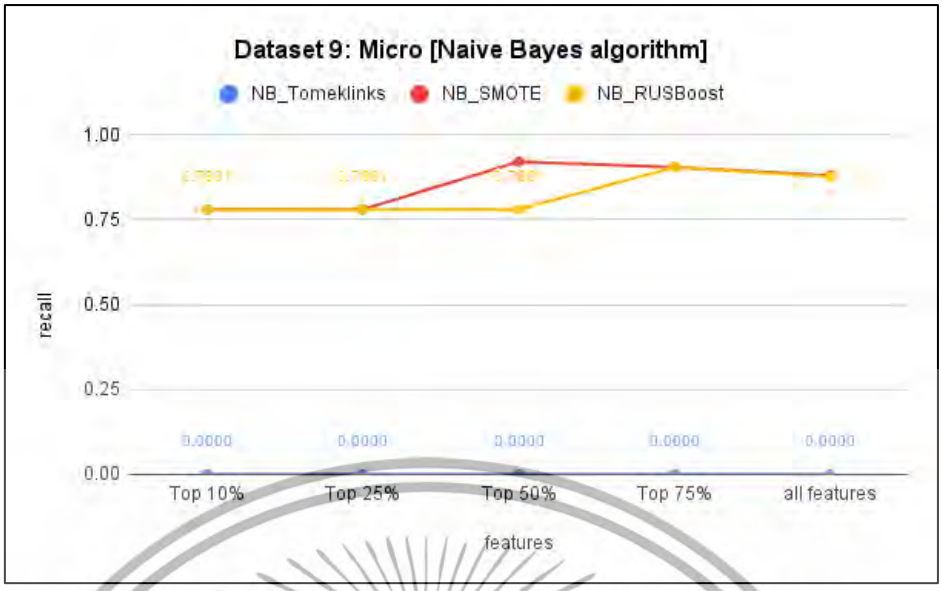



**รูปที่ 4.49 กราฟเส้นแสดงค่า Recall ในชุดข้อมูล Microcalcification Classification**



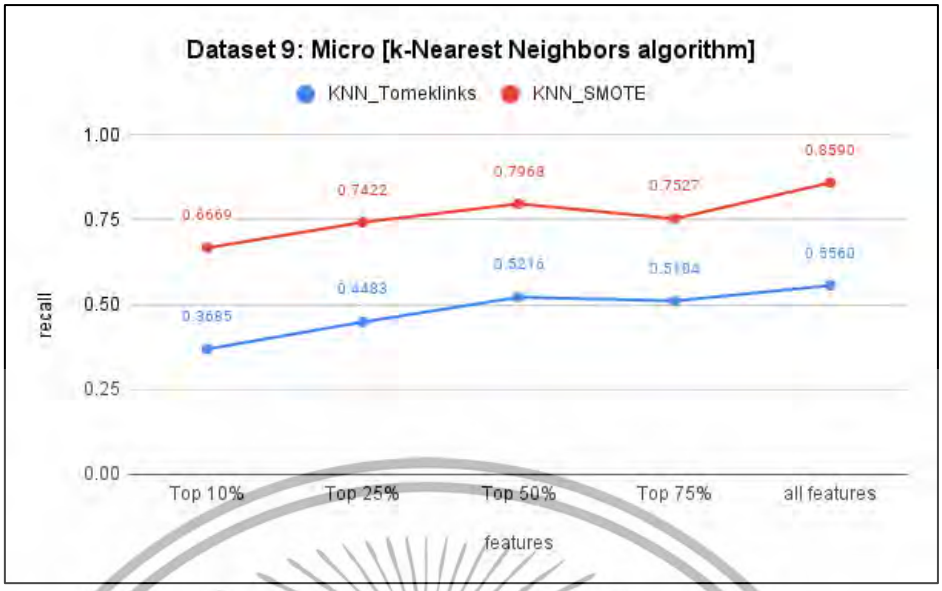

**รูปที่ 4.50 กราฟเส้นแสดงค่า Recall ในชุดข้อมูล Microcalcification Classification ของอัลกอริทึม Decision Trees**

จากรูปที่ 4.50 ชุดข้อมูล Microcalcification Classification ทดลองบนอัลกอริทึม Decision Trees พบว่าเทคนิค Tomek Links จำนวนคุณลักษณะที่ลดลง ส่งผลต่อโมเดลมีประสิทธิภาพในการจำแนกค่า Recall ที่ลดลงแปรผันตรงตามจำนวนคุณลักษณะ เทคนิค SMOTE มีประสิทธิภาพการจำแนกที่ดี และจำนวนคุณลักษณะที่ลดลง ส่งผลต่อประสิทธิภาพการจำแนกของโมเดลที่ค่า Recall ไม่แตกต่างกัน จำนวนคุณลักษณะจึงไม่มีผลต่อเทคนิค SMOTE บนข้อมูลชุดนี้ และเทคนิค RUSBoostClassifier จำนวนคุณลักษณะที่ลดลง ส่งผลให้ประสิทธิภาพในการจำแนกลดลงแต่มีประสิทธิภาพการจำแนกของโมเดลมีค่า Recall สูงที่สุดเมื่อเทียบเทคนิคการสุ่มตัวอย่างอื่น



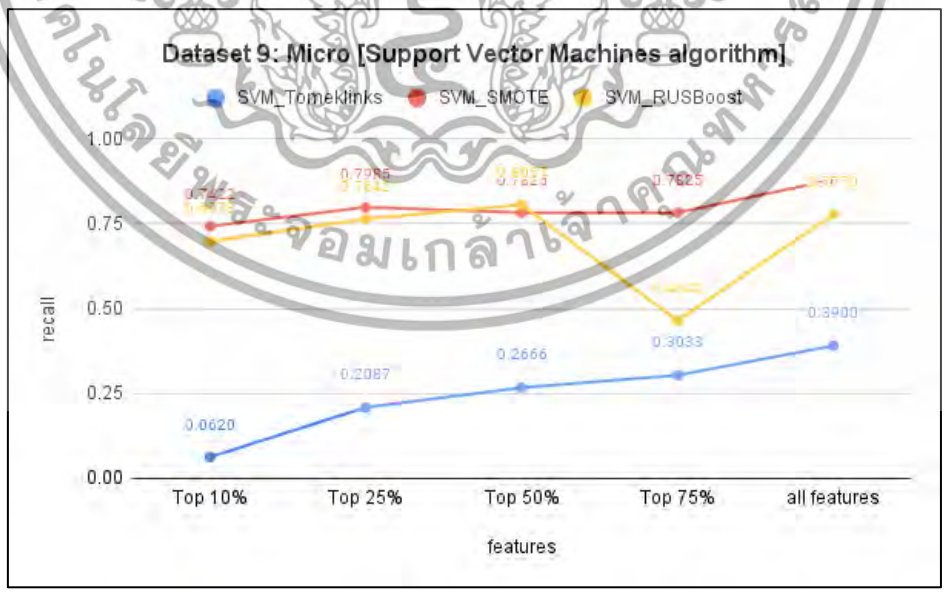

**รูปที่ 4.51 กราฟเส้นแสดงค่า Recall ในชุดข้อมูล Microcalcification Classification ของอัลกอริทึม Naïve Bayes**

จากรูปที่ 4.51 ชุดข้อมูล Microcalcification Classification ทดลองบนอัลกอริทึม Naïve Bayes พบว่าเทคนิค Tomek Links จำนวนคุณลักษณะที่ลดลง ส่งผลต่อโมเดลมีประสิทธิภาพในการจำแนกประสิทธิภาพค่า Recall ที่ลดลงแปรผันตรงตามจำนวนคุณลักษณะ เทคนิค SMOTE มีประสิทธิภาพการจำแนกที่ดี และจำนวนคุณลักษณะที่ลดลง ส่งผลต่อประสิทธิภาพการจำแนกประเภทของโมเดลมีประสิทธิภาพค่า Recall ลดลงตามลำดับ และเทคนิค RUSBoostClassifier จำนวนคุณลักษณะส่งผลต่อประสิทธิภาพในการจำแนกที่แปรผันตรงกัน



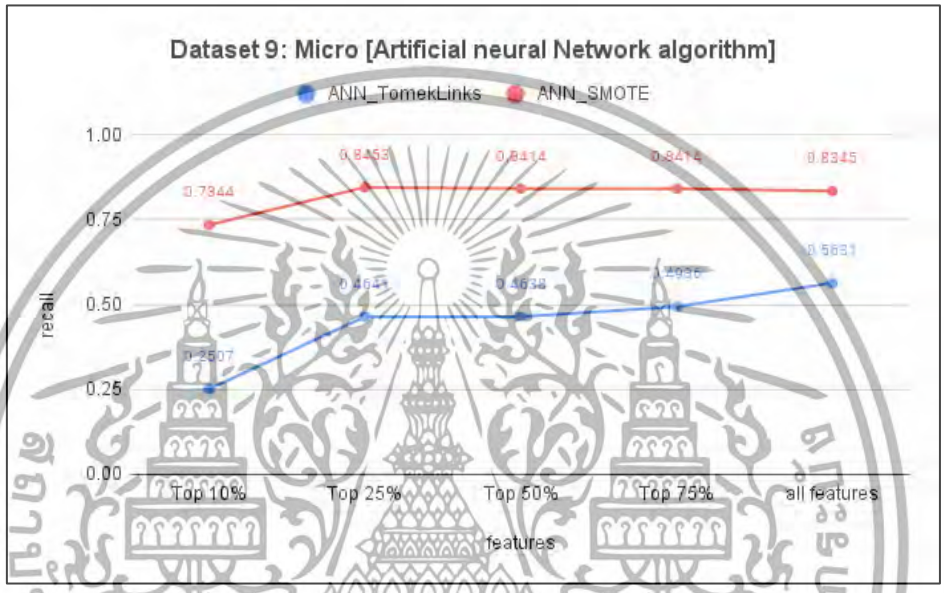

**รูปที่ 4.52 กราฟเส้นแสดงค่า Recall ในชุดข้อมูล Microcalcification Classification ของอัลกอริทึม k-Nearest Neighbors**

จากรูปที่ 4.52 ชุดข้อมูล Microcalcification Classification ทดลองบนอัลกอริทึม k-Nearest Neighbors พบว่าเทคนิค Tomek Links จำนวนคุณลักษณะที่ลดลง ส่งผลต่อโมเดลมีประสิทธิภาพในการจำแนกประสิทธิภาพค่า Recall ที่ลดลงแปรผันตรงกัน ส่วนเทคนิค SMOTE มีประสิทธิภาพการจำแนกที่ดีกว่าเทคนิค Tomek Links และจำนวนคุณลักษณะที่ลดลง ส่งผลต่อประสิทธิภาพการจำแนกประเภทของโมเดลมีประสิทธิภาพค่า Recall ลดลงตามลำดับ




**รูปที่ 4.53 กราฟเส้นแสดงค่า Recall ในชุดข้อมูล Microcalcification Classification ของอัลกอริทึม Support Vector Machines**

จากรูปที่ 4.53 ชุดข้อมูล Microcalcification Classification ทดลองบนอัลกอริทึม Support Vector Machines พบว่าเทคนิค Tomek Links จำนวนคุณลักษณะที่ลดลง ส่งผลต่อโมเดลมีประสิทธิภาพในการจำแนกประสิทธิภาพค่า Recall ที่ลดลงแปรผันตรงกัน ส่วนเทคนิค SMOTE จำนวนคุณลักษณะที่ลดลง ส่งผลต่อประสิทธิภาพการจำแนกประเภทของโมเดลมีประสิทธิภาพค่าRecall ที่ลดลงตามลำดับ และเทคนิค RUSBoostClassifier จำนวนคุณลักษณะที่มีความสัมพันธ์ระหว่างคุณลักษณะน้อย ส่งผลต่อประสิทธิภาพในการจำแนกประเภทข้อมูล




**รูปที่ 4.54 กราฟเส้นแสดงค่า Recall ในชุดข้อมูล Microcalcification Classification ของอัลกอริทึม Artificial Neural Networks**

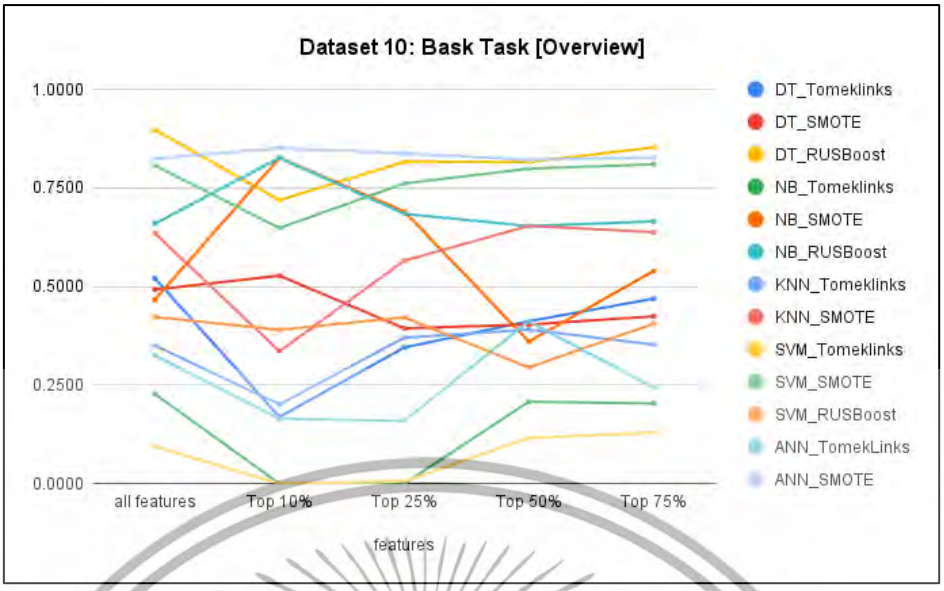
จากรูปที่ 4.54 ชุดข้อมูล Microcalcification Classification ทดลองบนอัลกอริทึม Artificial Neural Networks พบว่าเทคนิค Tomek Links จำนวนคุณลักษณะที่ลดลง ส่งผลต่อโมเดลมีประสิทธิภาพในการจำแนกประสิทธิภาพค่า Recall ที่ลดลงแปรผันตรงกัน ส่วนเทคนิค SMOTE มีประสิทธิภาพการจำแนกที่ดีกว่าเทคนิค Tomek Links และจำนวนคุณลักษณะที่ลดลงส่งผลต่อประสิทธิภาพการจำแนกประเภทของโมเดลมีประสิทธิภาพค่า Recall ที่ลดลงตามลำดับ

### 4.1.10 ชุดข้อมูล Bank Marketing Task

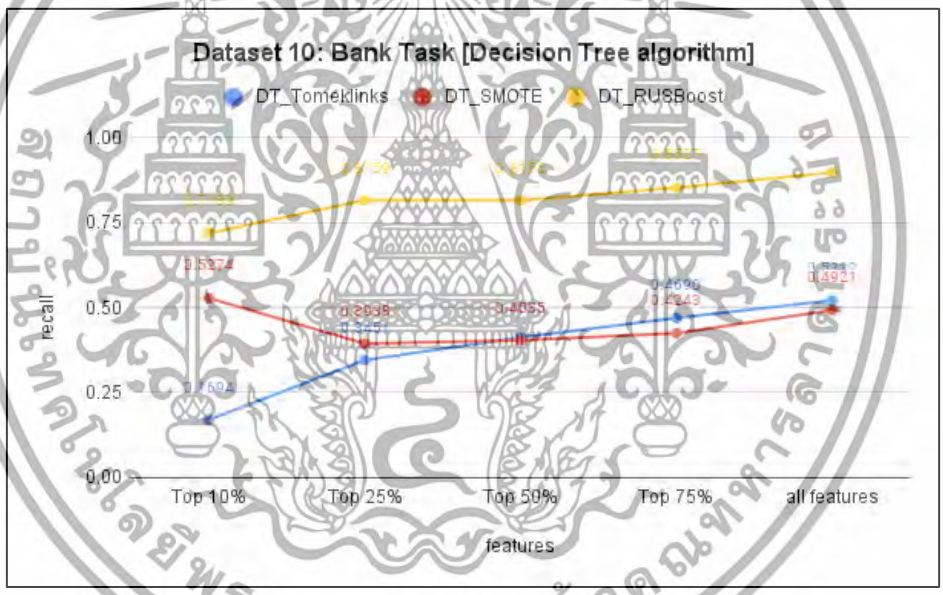
ตารางที่ 4.10 ค่า Recall ของชุดข้อมูลที่ไม่สมดุลชุดที่ 10 Bank Marketing Task

| Model | all features | Top 10% | Top 25% | Top 50% | Top 75% |
|---|---|---|---|---|---|
| DT_Tomeklinks | **0.5212±0.0116** | 0.1694±0.0197 | 0.3451±0.0214 | 0.412±0.0192 | 0.469±0.0356 |
| DT_SMOTE | 0.4921±0.0157 | **0.5274±0.0272** | 0.3938±0.0215 | 0.4035±0.0276 | 0.4243±0.0202 |
| DT_RUSBoost | **0.8975±0.0088** | 0.7189±0.0494 | 0.8159±0.0255 | 0.8155±0.0177 | 0.8527±0.0137 |
| NB_Tomeklinks | **0.2277±0.0215** | 0±0 | 0±0 | 0.2079±0.02 | 0.2034±0.0186 |
| NB_SMOTE | 0.4664±0.0311 | **0.8262±0.0186** | 0.6906±0.0263 | 0.3605±0.0272 | 0.5392±0.0331 |
| NB_RUSBoost | 0.6593±0.026 | **0.8262±0.0186** | 0.6843±0.0243 | 0.653±0.0236 | 0.6651±0.0233 |
| KNN_Tomeklinks | 0.3504±0.0224 | 0.201±0.0126 | 0.3701±0.0242 | **0.3907±0.0227** | 0.3528±0.0218 |
| KNN_SMOTE | 0.6354±0.0189 | 0.3357±0.0194 | 0.5653±0.0179 | **0.6538±0.0145** | 0.6377±0.0242 |
| SVM_Tomeklinks | 0.0946±0.0112 | 0±0 | 0.0058±0.0092 | 0.1161±0.0149 | **0.1296±0.0137** |
| SVM_SMOTE | 0.807±0.0147 | 0.6482±0.0165 | 0.7612±0.0248 | 0.7988±0.0152 | **0.8096±0.0205** |
| SVM_RUSBoost | **0.4225±0.2673** | 0.3903±0.0413 | 0.4216±0.0598 | 0.2948±0.1073 | 0.4055±0.2691 |
| ANN_TomekLinks | 0.3247±0.1526 | 0.1647±0.1339 | 0.1594±0.0982 | **0.4116±0.1372** | 0.244±0.1049 |
| ANN_SMOTE | 0.824±0.061 | **0.8506±0.0504** | 0.8376±0.0314 | 0.8211±0.0352 | 0.8276±0.0721 |

จากตารางที่ 4.10 แสดงการเปรียบเทียบค่า Recall ที่แสดงจำนวนคุณลักษณะส่งผลต่อประสิทธิภาพความสามารถในการจดจำ Positive class อย่างค่า Recall การจำแนกประเภทของโมเดลที่ใช้เทคนิคการสุ่มตัวอย่าง Tomek Links เทคนิค SMOTE และเทคนิค RUSBoostClassifier ในการจัดความสมดุลของชุดข้อมูลผ่านการเรียนรู้ของเครื่องโดยใช้อัลกอริทึม Decision Trees, Naïve Bayes, k-Nearest Neighbors, Support Vector Machines และ Artificial Neural Networks

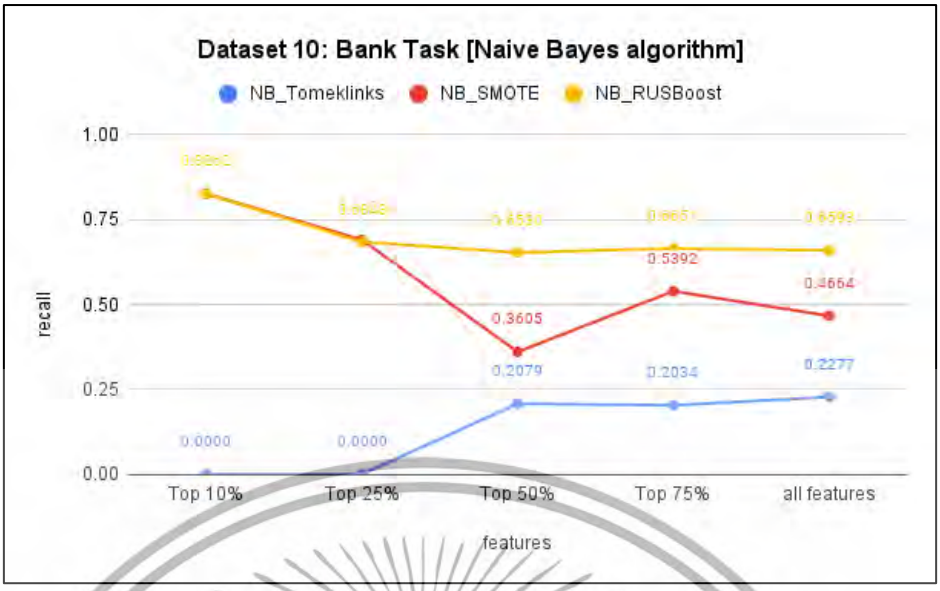



**รูปที่ 4.55 กราฟเส้นแสดงค่า Recall ในชุดข้อมูล Bank Marketing Task**



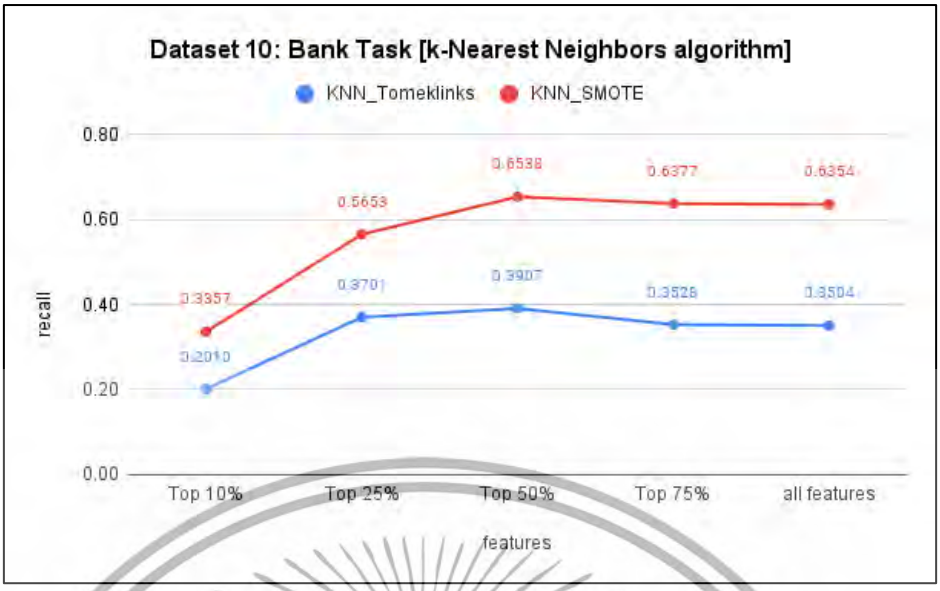

**รูปที่ 4.56 กราฟเส้นแสดงค่า Recall ในชุดข้อมูล Bank Marketing Task ของอัลกอริทึม Decision Trees**

จากรูปที่ 4.56 ชุดข้อมูล Bank Marketing Task ทดลองบนอัลกอริทึม Decision Trees พบว่าเทคนิค Tomek Links จำนวนคุณลักษณะมีผลต่อการจำแนกประเภทข้อมูลจำนวนคุณลักษณะที่ลดลงโมเดลมีประสิทธิภาพการจำแนกค่า Recall ที่ลดลงลำดับ เทคนิค SMOTE ความสัมพันธ์ระหว่างคุณลักษณะ ส่งผลต่อประสิทธิภาพในการจำแนกประเภทของโมเดล และเทคนิค RUSBoostClassifier จำนวนคุณลักษณะที่ลดลง ส่งผลต่อประสิทธิภาพการจำแนกของโมเดลมีค่า Recall ลดลงที่แปรผันตรงกัน



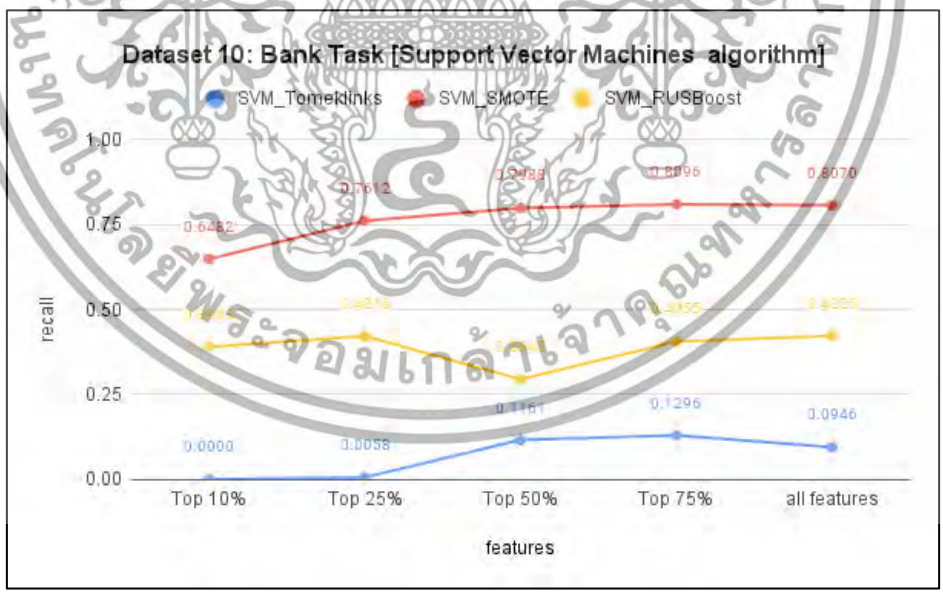

**รูปที่ 4.57 กราฟเส้นแสดงค่า Recall ในชุดข้อมูล Bank Marketing Task ของอัลกอริทึม Naïve Bayes**

จากรูปที่ 4.57 ชุดข้อมูล Bank Marketing Task ทดลองบนอัลกอริทึม Naïve Bayes พบว่า เทคนิค Tomek Links จำนวนคุณลักษณะมีผลต่อประสิทธิภาพการจำแนกประเภทข้อมูลของโมเดล จำนวนคุณลักษณะที่เพิ่มขึ้น ประสิทธิภาพการจำแนกประเภทของโมเดลมีค่า Recall เพิ่มขึ้น ส่วนเทคนิค SMOTE จำนวนคุณลักษณะมีความสัมพันธ์ที่สัมพันธ์กัน ส่งผลให้ประสิทธิภาพการจำแนกของโมเดลมีค่า Recall ที่เพิ่มขึ้น ส่วนเทคนิค RUSBoostClassifier จำนวนคุณลักษณะที่ลดลง โมเดลมีประสิทธิภาพในการจำแนกค่า Recall ที่เพิ่มขึ้นแปรผกผันกับจำนวนคุณลักษณะ



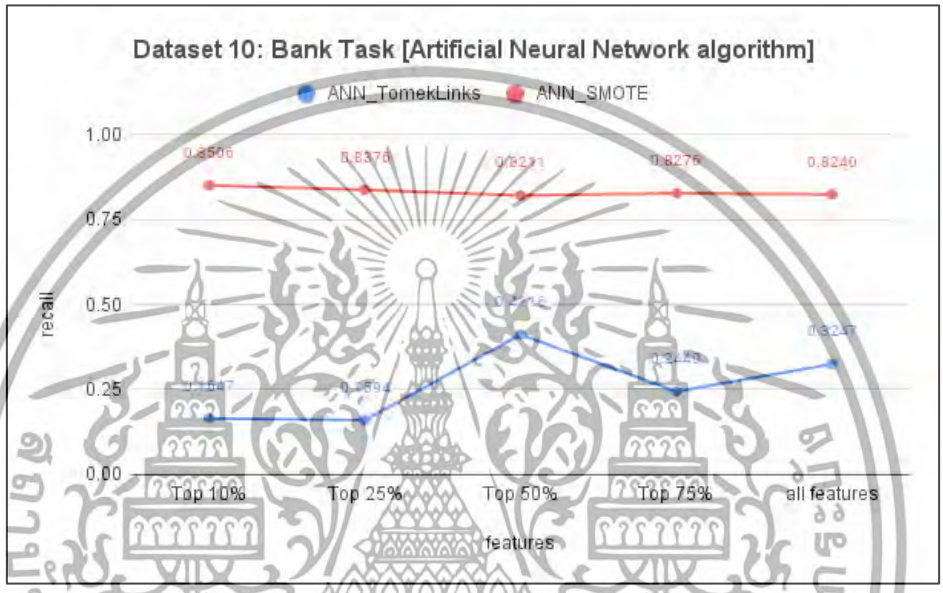

**รูปที่ 4.58 กราฟเส้นแสดงค่า Recall ในชุดข้อมูล Bank Marketing Task ของอัลกอริทึม k-Nearest Neighbors**

จากรูปที่ 4.58 ชุดข้อมูล Bank Marketing Task ทดลองบนอัลกอริทึม k-Nearest Neighbors พบว่าเทคนิค Tomek Links และเทคนิค SMOTE จำนวนคุณลักษณะที่ลดลงกลับส่งผลต่อโมเดลให้มีประสิทธิภาพค่า Recall ที่ลดลงตามลำดับ




**รูปที่ 4.59 กราฟเส้นแสดงค่า Recall ในชุดข้อมูล Bank Marketing Task ของอัลกอริทึม Support Vector Machines**

จากรูปที่ 4.59 ชุดข้อมูล Bank Marketing Task ทดลองบนอัลกอริทึม Support Vector Machines พบว่าเทคนิค Tomek Links จำนวน คุณลักษณะที่ลดลง ส่งผลให้การวัดประสิทธิภาพโมเดลมีค่า Recall ที่ลดลงตามลำดับ ส่วนเทคนิค SMOTE สามารถทำงานร่วมกับอัลกอริทึม Support Vector Machines ได้อย่างมีประสิทธิภาพ และจำนวนคุณลักษณะที่เพิ่มขึ้น ส่งผลให้โมเดลมีประสิทธิภาพการจำแนกค่า Recall ที่เพิ่มขึ้น และเทคนิค RUSBoostClassifier จำนวนคุณลักษณะที่ลดลง ส่งผลต่อประสิทธิภาพการจำแนกประเภทของโมเดลมีค่า Recall ที่ลดลง




**รูปที่ 4.60 กราฟเส้นแสดงค่า Recall ในชุดข้อมูล Bank Marketing Task ของอัลกอริทึม Artificial Neural Networks**

จากรูปที่ 4.60 ชุดข้อมูล Bank Marketing Task ทดลองบนอัลกอริทึม Artificial Neural Networks พบว่าเทคนิค Tomek Links ความสัมพันธ์ของคุณลักษณะส่งผลต่อการการจำแนกประเภทของโมเดล Artificial Neural Networks ส่วนเทคนิค SMOTE ทำงานร่วมกับอัลกอริทึม Artificial Neural Networks ได้อย่างมีประสิทธิภาพ และจำนวนคุณลักษณะในข้อมูลชุดนี้ ไม่ได้ส่งผลต่อประสิทธิภาพการจำแนกประเภทของโมเดลมีค่า Recall ที่แตกต่างกันมาก

## 4.2 อภิปรายผลการทดลอง

จากผลการทดลองทั้ง 10 ชุดข้อมูลที่ไม่สมดุล นำมาสรุปผลในรูปแบบตาราง ประกอบด้วย ชื่อชุดข้อมูล จำนวนข้อมูล จำนวนคุณลักษณะ อัตราส่วนความสมดุลกันระหว่างกลุ่ม Majority Class และกลุ่ม Minority Class ประเภทคุณลักษณะ และอันดับโมเดล 3 อันดับแรกที่มีประสิทธิในการจำแนกชุดข้อมูลที่ไม่สมดุล

ตารางที่ 4.11 แสดงผลการทดลองจากชุดข้อมูลที่ไม่สมดุลทั้งหมด 10 ชุดข้อมูล

| no. | Dataset | Instance | Features | Im. Ratio | Attribute types | model rank 1 (Recall) | model rank 2 (Recall) | model rank 3 (Recall) |
|---|---|---|---|---|---|---|---|---|
| 1 | Stroke Prediction Dataset | 5,110 | 11 | 19.52 | Nominal(6) Numeric(3) Binary(2) | SVM+SMOTE (79.60%) | DT+RUSBoost (77.80%) | NB+RUSBoost (65.9%) |
| 2 | COVID - 19 Dataset | 125,152 | 17 | 12.72 | Nominal(16) Numeric(1) | DT+RUSBoost (92.19%) | SVM+SMOTE (91.12%) | ANN+SMOTE (90.32%) |
| 3 | Diabetes Prediction Dataset | 100,000 | 8 | 10.76 | Nominal(2) Numeric(4) Binary(2) | ANN+SMOTE (90.13%) | DT+RUSBoost (90.00%) | SVM+SMOTE (86.00%) |
| 4 | Water Quality | 7,999 | 16 | 7.77 | Numeric(16) | DT+RUSBoost (82.1%) | ANN+SMOTE (77.65%) | NB+RUSBoost (73.80%) |
| 5 | Credit Card Fraud | 100,000 | 7 | 10.44 | Nominal(4) Numeric(3) | DT+RUSBoost (99.99%)) | DT+SMOTE (99.93%)) | DT+TomekLinks (99.91%) |
| 6 | Bank Marketing Dataset | 45,211 | 16 | 7.55 | Nominal(10) Numeric(6) | DT+RUSBoost (90.50%) | ANN+SMOTE (82.09%) | SVM+SMOTE (78.30%) |
| 7 | Heart Disease | 119,795 | 17 | 10.68 | Nominal(13) Numeric(4) | SVM+SMOTE (79.21%) | DT+RUSBoost (76.42%) | NB+RUSBoost (65.85%) |
| 8 | Lumpy Skin Disease | 24,803 | 16 | 7.16 | Numeric(16) | NB+RUSBoost (95.73%) | DT+RUSBoost (94.85%) | ANN+SMOTE (93.88%) |
| 9 | Microcalcification classification | 11,183 | 5 | 42.01 | Numeric(5) | DT+RUSBoost (90.60%) | SVM+SMOTE (88.00%) | ANN+SMOTE (83.45%) |
| 10 | Bank Marketing Task | 4,521 | 16 | 7.68 | Nominal(9) Numeric(7) | DT+RUSBoost (89.75%) | ANN+SMOTE (82.40%) | SVM+SMOTE (80.70%) |

จากตารางที่ 4.11 พบว่าเทคนิคการสุ่มตัวอย่างโดยใช้เทคนิคการเพิ่มประสิทธิภาพโมเดลทำงานร่วมกันอย่างเทคนิค RUSBoostClassifier ทำงานร่วมกันได้ดีกับอัลกอริทึม Decision Trees ได้อย่างมีประสิทธิภาพ และเทคนิคการสุ่มตัวอย่างแบบเทคนิค SMOTE ทำงานร่วมกับอัลกอริทึม Support Vector Machines และ Artificial Neural Networks ได้อย่างมีประสิทธิภาพ จึงได้ข้อสังเกตว่าเทคนิค RUSBoostClassifier เหมาะที่จะทำงานร่วมกับอัลกอริทึมที่ใช้หลักความน่าจะเป็นในการจำแนกประเภทข้อมูล ส่วนเทคนิค SMOTE เหมาะกับอัลกอริทึมที่ใช้หลักการเชิงคณิตศาสตร์ในการคำนวณหาระยะห่างของจุดข้อมูล ในส่วนเทคนิคการสุ่มตัวอย่างแบบเทคนิค Tomek Links สามารถทำงานร่วมกับอัลกอริทึม k-Nearest Neighbors มีประสิทธิภาพมากกว่าการทำงานร่วมกับอัลกอริทึมอื่น นอกจากนี้พบว่าประเภทคุณลักษณะข้อมูลที่เป็น Nominal หรือ Numeric เหมาะสมที่จะใช้เทคนิค SMOTE และเทคนิค RUSBoostClassifier เพื่อจัดความสมดุลให้กับชุดข้อมูล

สำหรับการทดลองนำชุดข้อมูลที่ไม่สมดุลผ่านการคัดเลือกคุณลักษณะ พบว่าจำนวนคุณลักษณะกลับส่งผลต่ออัลกอริทึมในการจัดการระดับข้อมูลอย่างเทคนิคการสุ่มตัวอย่างสังเคราะห์ใหม่ เทคนิค SMOTE ที่จำนวนคุณลักษณะส่งผลต่อประสิทธิภาพในการจำแนกประเภทข้อมูลของโมเดลที่ใช้เทคนิคนี้มากที่สุด สังเกตจากตารางเปรียบเทียบค่า Recall ในแต่ละชุดข้อมูล แสดงให้เห็นถึงจำนวนคุณลักษณะที่เกี่ยวข้องมีผลต่อการจำแนกของโมเดลที่ใช้เทคนิค SMOTE มีประสิทธิภาพที่เพิ่มขึ้น และในทางตรงกันข้ามเทคนิคการสุ่มตัวอย่างแบบลดจำนวนตัวอย่างกลุ่มคลาสส่วนใหญ่แบบเทคนิค Tomek Links จำนวนคุณลักษณะที่ลดลงกลับส่งผลต่อประสิทธิภาพในการจำแนกของโมเดลลดลงตามลำดับ สรุปได้ว่าการใช้เทคนิค Tomek Links จำนวนคุณลักษณะแปรผันตรงกับประสิทธิภาพในการจำแนกของโมเดล

ในกรณีที่จัดการชุดข้อมูลที่มีประเภทคุณลักษณะทั้งแบบ Nominal และ Numeric ในถูกเก็บในชุดข้อมูล ต้องการทราบว่าเทคนิคหรืออัลกอริทึมการเรียนรู้ของเครื่องอัลกอริทึมใดเหมาะสมกับชุดข้อมูลที่มีประเภทคุณลักษณะทั้ง 2 ประเภทนี้ จากผลการทดลองที่ได้จากงานวิจัย สามารถเลือกใช้เทคนิค SMOTE กับอัลกอริทึม Support Vector Machines และเทคนิค RUSBoostClassifier กับอัลกอริทึม Decision Trees มาทดสอบเพื่อเปรียบเทียบประสิทธิภาพโมเดลเป็นขั้นตอนที่ง่ายสำหรับการตัดสินใจเลือกใช้โมเดลที่เหมาะสมกับชุดข้อมูล

# บทที่ 5
# สรุปผลการทดลองและข้อเสนอแนะ

## 5.1 สรุปผลการทดลอง

งานวิจัยนี้มีจุดมุ่งหมายในการศึกษาอัลกอริทึมที่เกี่ยวกับการปรับปรุงปัญหาชุดข้อมูลที่ไม่สมดุลและเปรียบเทียบประสิทธิภาพการทำงานของอัลกอริทึม โดยใช้เทคนิคการสุ่มตัวอย่างการเพิ่มตัวอย่างสังเคราะห์ SMOTE เทคนิคการสุ่มตัวอย่างแบบจับคู่คลาสที่ตรงกันข้ามที่ใกล้ที่สุด Tomek Links และเทคนิคการเพิ่มประสิทธิภาพโมเดลแบบ AdaBoost ที่ผสมผสานเทคนิคการสุ่มตัวอย่างแบบลบตัวอย่างกลุ่มคลาสส่วนใหญ่อย่างเทคนิค RUSBoostClassifier เทคนิคที่กล่าวมาข้างต้น เป็นเทคนิคที่ใช้ในการจัดการความไม่สมดุลของชุดข้อมูล เพื่อให้สามารถใช้ประโยชน์จากชุดข้อมูลได้ดียิ่งขึ้น โดยจะทำงานร่วมกับอัลกอริทึม Decision Trees, Naïve Bayes, k-Nearest Neighbors, Support Vector Machines และ Artificial Neural Networks ในการสร้างโมเดลในการจำแนกประเภทข้อมูลซึ่งจะนำโมเดลที่ประกอบไปด้วยอัลกอริทึมเหล่านี้ มาเปรียบประสิทธิภาพโมเดลผ่านชุดข้อมูลที่ไม่สมดุลจากเว็บไซต์ Kaggle จำนวน 10 ชุดข้อมูลที่ผ่านการคัดเลือกคุณลักษณะที่เกี่ยวข้องอย่างมีนัยสำคัญ

สำหรับขั้นตอนการดำเนินงานวิจัยเริ่มต้นจากการเตรียมชุดข้อมูลที่ไม่สมดุลให้พร้อมสำหรับการเรียนรู้ของเครื่อง และทำการคัดเลือกคุณลักษณะของชุดข้อมูลผ่านการคัดเลือกแบบ ANOVA โดยพิจารณาจากค่า F-statistic และค่า p-value ที่คุณลักษณะเกี่ยวข้องกับคลาสเป้าหมายในการจำแนกประเภทที่สุด โดยจะจัดการปรับความสมดุลของชุดข้อมูลผ่านเทคนิคการสุ่มตัวอย่าง SMOTE Tomek Links และเทคนิคเพิ่มประสิทธิภาพโมเดล RUSBoostClassifier และให้เครื่องเรียนรู้ผ่านการฝึกสอนจากอัลกอริทึมดังนี้ Decision Trees, Naïve Bayes, k-Nearest Neighbors, Support Vector Machines และ Artificial Neural Networks เพื่อเปรียบเทียบประสิทธิภาพโมเดล เทคนิคที่ใช้ในการปรับความสมดุลของชุดข้อมูลมีความเหมาะสมในชุดข้อมูลลักษณะใด และจำนวนคุณลักษณะส่งผลต่อการต่อจำแนกประเภทข้อมูลในการเรียนรู้ของเครื่อง

จากการทดลองที่นำชุดข้อมูลผ่านเทคนิคการสุ่มตัวอย่างพบว่าเทคนิค RUSBoostClassifier ทำงานได้ดีกับโมเดลที่ใช้ความน่าจะเป็นในการจำแนกข้อมูลอย่าง Decision Trees และ Naïve Bayes ส่วนเทคนิค SMOTE ทำงานได้ดีกับโมเดลที่ใช้การคำนวณเชิงคณิตศาสตร์ ในการคำนวณหาระยะห่างระหว่างข้อมูลในการพิจารณาการจำแนกประเภทข้อมูลอย่างเช่น Support Vector Machines, Artificial Neural Networks และ k-Nearest Neighbors ส่วนเทคนิค Tomek Links จำนวนคุณลักษณะส่งผลต่อประสิทธิภาพการจำแนกข้อมูล ถ้าจำนวนคุณลักษณะลดลง ประสิทธิภาพการจำแนกจะลดลงแปรผันตรงตามจำนวนคุณลักษณะ นอกจากนี้เทคนิค Tomek Links เหมาะที่จะทำงานร่วมกับอัลกอริทึม k-Nearest Neighbors เพราะทำให้มีประสิทธิภาพในการจำแนกที่เพิ่มขึ้น

และจากการทดลองประเภทคุณลักษณะของชุดข้อมูลแบบ Nominal และ Numeric เหมาะสมกับเทคนิค SMOTE และเทคนิค RUSBoostClassifier เป็นเทคนิคที่ใช้ในการจัดความสมดุลของชุดข้อมูลให้มีประสิทธิภาพเพิ่มขึ้น

## 5.2 ปัญหาและข้อเสนอแนะ

- อัลกอริทึม k-Nearest Neighbors และ Artificial Neural Networks ไม่สามารถทำงานร่วมกับเทคนิค RUSBoostClassifier เนื่องจากการเรียก Library ของทั้ง 2 อัลกอริทึมที่กล่าวมาข้างต้น ไม่สนับสนุนการทำงานร่วมกับเทคนิค RUSBoostClassifier
- อัลกอริทึม Support Vector Machines เป็นอัลกอริทึมที่ใช้เวลาการหาคำตอบที่ใช้เวลาในการหาคำตอบค่อนข้างนาน ดังนั้นจึงมีการปรับเปลี่ยนพารามิเตอร์ให้มีการทำงานแบบขนานเพื่อลดเวลาในการจำแนกประเภทข้อมูลของอัลกอริทึม

# เอกสารอ้างอิง


ไพโรจน์ เจริญศรี. 2023. **ประวัติและหลักการของเครือข่ายประสาทเทียม (Artificial Neural Networks – ANNs)**. [Online]. Available: https://bigdata.go.th/big-data-101/neural-network/. สืบค้นเมื่อ 12 ตุลาคม 2566.

รัสรินทร์ เมธาเฉลิมพัฒน์. **การประยุกต์ใช้ Machine Learning กับ ภาคอุตสาหกรรม ตอนที่ 1**. [Online]. Available: https://www.nectec.or.th/wp-content/uploads/2022/08/CPS-ML-manufacturing.pdf. สืบค้นเมื่อ 12 ตุลาคม 2566.

Eakasit Pacharawongsakda, Ph.D.. 2015. **การแบ่งข้อมูลเพื่อนำมาทดสอบประสิทธิภาพของโมเดล**. [Online]. Available: https://th.linkedin.com/pulse/การแบ่งข้อมูลเพื่อนำมาทดสอบประสิทธิภาพของโมเดล-eakasit-pacharawongsakda. สืบค้นเมื่อ 11 สิงหาคม 2566

IBM. **What is a Decision Trees?**. [Online]. Available: https://www.ibm.com/topics/decision-trees. สืบค้นเมื่อ 12 ตุลาคม 2566.

IBM. **What is the k-nearest neighbors (KNN) algorithm?**. [Online]. Available: https://www.ibm.com/topics/knn. สืบค้นเมื่อ 12 ตุลาคม 2566.

IBM. **What are Naïve Bayes classifiers?**. [Online]. Available: https://www.ibm.com/topics/Naïve-bayes. สืบค้นเมื่อ 12 ตุลาคม 2566

Isak Forslund. 2022. **Modification of the RUSBoost algorithm A comparison of classifiers on imbalanced data**. [Online]. Available: https://www.diva-portal.org/smash/get/diva2:1672924/FULLTEXT01.pdf. สืบค้นเมื่อ 11 ตุลาคม 2566.

Matthew Bernstein. 2017. **AdaBoost**. [Online]. Available: https://mbernste.github.io/files/notes/AdaBoost.pdf. สืบค้นเมื่อ 12 ตุลาคม 2566

Nuthdanai WangPratham. 2021. **Feature Selection in Python**. [Online]. Available: https://medium.com/qunt-i-love-u/feature-selection-in-python-9f79341b144c. สืบค้นเมื่อ 17 มกราคม 2567.

Rafael Alencar. **Resampling strategies for imbalanced datasets.** [Online]. Available: https://www.kaggle.com/code/rafjaa/resampling-strategies-for-imbalanced-datasets/notebook. สืบค้นเมื่อ 11 ตุลาคม 2566.

scikit-learn developers (BSD License). **Decision Trees**. [Online]. Available: https://scikit-learn.org/stable/modules/tree.html#classification. สืบค้นเมื่อ 11 พฤศจิกายน 2566

scikit-learn developers (BSD License). **KNeighborsClassifier**. [Online]. Available: https://scikitlearn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html#sklearn.neighbors.KNeighborsClassifier. สืบค้นเมื่อ 11 พฤศจิกายน 2566

scikit-learn developers (BSD License). **sklearn.svm.SVC**. [Online]. Available: https://scikit-learn.org/stable/ modules/generated/sklearn.svm.SVC.html#. สืบค้นเมื่อ 11 พฤศจิกายน 2566

The imbalanced-learn developers. **RUSBoostClassifier**. [Online]. Available: https://imbalanced-learn.org/stable/references/generated/imblearn.ensemble.RUSBoostClassifier.html. สืบค้นเมื่อ 11 สิงหาคม 2566.

The imbalanced-learn developers. **SMOTE**. [Online]. Available: https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.SMOTE.html. สืบค้นเมื่อ 11 สิงหาคม 2566.

The imbalanced-learn developers. **Tomek Links**. [Online]. Available: https://imbalanced-learn.org/stable/references/generated/imblearn.under_sampling.TomekLinks.html. สืบค้นเมื่อ 11 สิงหาคม 2566.

งานทะเบียนคณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

คำรับรองเล่มปัญหาพิเศษ

วันที่ 23 เดือน เมษายน พ.ศ 2567

ข้าพเจ้า นางสาว..สิริรัตน์....ไชยธงรัตน์..... รหัสประจำตัว.....63050201.....

นักศึกษาหลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชา.....วิทยาการคอมพิวเตอร์.....ภาควิชา...วิทยาการคอมพิวเตอร์ ขอรับรองว่าปัญหาพิเศษ เรื่อง

ชื่อภาษาไทย.....การศึกษาอัลกอริทึมระดับข้อมูลสำหรับชุดข้อมูลที่ไม่สมดุล.....

ชื่อภาษาอังกฤษ.....The Study Of Data Level Algorithms For Imbalanced Dataset.....

ปีการศึกษา.....2566.....

เป็นผลงานวิจัยที่มิได้คัดลอกหรือละเมิดลิขสิทธิ์ของผู้อื่นและได้ผ่านการตรวจสอบความซ้ำซ้อนเรียบร้อยแล้ว และได้แนบเอกสารการตรวจสอบการลอกเลียนงานวรรณกรรมที่ตรวจสอบจากเล่มปัญหาพิเศษฉบับสมบูรณ์แล้ว

โปรแกรมอักขราวิสุทธิ์.....0.24.....% หรือโปรแกรม Turnitin.....%

ลงชื่อ.....สิริรัตน์ ไชยธงรัตน์.....

( สิริรัตน์ ไชยธงรัตน์ )

นักศึกษา

ข้าพเจ้า ผศ.ดร.อ.อนันตพร....หรรษคุณาภัย..... อาจารย์ที่ปรึกษาปัญหาพิเศษ ได้ตรวจสอบปัญหาพิเศษของนักศึกษาข้างต้น แล้ว ขอรับรองว่าเป็นผลงานวิจัยของนักศึกษาจริงและมีเนื้อหาสมบูรณ์ จึงลงชื่อไว้เป็นหลักฐาน

ลงชื่อ.....

อาจารย์ที่ปรึกษา