การเปรียบเทียบวิธีการแก้ไขตัวอย่างที่ไม่สมดุลสำหรับกระบวนการจำแนกประเภท

นายวัชรากร บุทธิจักร 59070502436

นายสุรกานต์ สุขสิริโสภาค 59070502445

นายกฤษฎา สุชาติปัทมกุล 59070502454

การศึกษาโครงงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตร์บัณฑิต

ภาควิชาวิศวกรรมอิเล็กทรอนิกส์และโทรคมนาคม

คณะวิศวกรรมศาสตร์

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

ภาคการศึกษาที่ 1/2562

Comparison of methods to tackle class imbalance in binary classification

Watcharakorn	Buttijak	59070502436
Surakarn	Suksirisophak	59070502445
Krissada	Suchatpattamakul	59070502454

A project study is submitted in partial fulfillment of the requirements for the degree of Bachelor of Engineering

Department of Electronic and Telecommunication Engineering
Faculty of Engineering
King Mongkut's University of Technology Thonburi

Semester 1/2019

หัวข้อโครงงาน

การเปรียบเทียบวิธีการแก้ไขตัวอย่างที่ไม่สมดุลสำหรับกระบวนการจำแนกประเภท

หน่วยกิตของโครงงาน 1

จัดทำโดย นายวัชรากร บุทธิจักร

นายสุรกานต์ สุขสิริโสภาค

นายกฤษฎา สุชาติปัทมกุล

อาจารย์ที่ปรึกษา ผศ.ดร.วัชรพันธ์ สุวรรณสันติสุข

อาจารย์ที่ปรึกษาร่วม -

ระดับการศึกษา วิศวกรรมศาสตรบัณฑิต

ภาควิชา วิศวกรรมอิเล็กทรอนิกส์และโทรคมนาคม

ภาคปีการศึกษา 1/2562

บทคัดย่อ

การจำแนกประเภทตัวอย่างออกเป็นหนึ่งประเภทจากสองประเภทที่เป็นไปได้ (binary classification) เป็นเทคนิคที่มีประโยชน์ในหลายสาขาวิชา การจำแนกประเภททำโดยเรียนรู้ตัวจำแนกประเภท (classifier) จากตัวอย่าง (sample) หลาย ๆ ตัวอย่างที่ได้เก็บไว้ล่วงหน้าและได้ระบุประเภทของแต่ละตัวอย่างไว้แล้ว (supervised learning) ตัวอย่างเหล่านี้อาจมีประเภทซึ่งไม่สมดุลกัน กล่าวคือตัวอย่างในประเภทที่หนึ่ง อาจน้อยกว่าตัวอย่างในอีกประเภทอยู่มาก ความไม่สมดุลมีผลเสีย ทำให้ตัวจำแนกประเภทที่เรียนรู้จาก ชุดข้อมูลนั้น มีความแม่นยำ (accuracy) ต่ำ โครงการนี้เปรียบเทียบวิธีการแก้ไขตัวอย่างที่ไม่สมดุลหลาย ๆ วิธี วิธีดำเนินโครงการเริ่มจากการพัฒนาโปรแกรม Matlab สำหรับตัวจำแนกประเภท เช่น support vector machine พัฒนาโปรแกรม Matlab สำหรับวิธีแก้ไขตัวอย่างที่ไม่สมดุล และประมาณความแม่นยำ ความอ่อนไหว (sensitivity) และความจำเพาะ (specificity) ของตัวจำแนกประเภทแต่ละตัว ซึ่งใช้คู่กับวิธีแก้ไขตัวอย่างที่สมดุล แต่ละวิธี ชุดข้อมูลที่ใช้เรียนรู้ตัวจำแนกประเภทคือชุดข้อมูลซึ่งนักวิจัยอื่นได้เก็บและเผยแพร่เป็นสาธารณะ โครงการนี้ช่วยในการเลือกวิธีแก้ไขตัวอย่างที่ไม่สมดุลและเพิ่มความแม่นยำของการจำแนกประเภท

Project Title	Comparison of methods to tackle class imbalance in binary classification
Project Credits	1
Project Participants	Watcharakorn Buttijak Surakarn Suksirisophak Krissada Suchatpattmakul
Advisor	Assist. Prof. Watcharapan Suwansantisuk, Ph.D.
Co-Advisor	-
Degree of Study	Bachelor of Engineering
Department	Electronic and Telecommunication Engineering
Semester	1/2019

Abstract

Binary classification has tremendous applications in several fields and can be achieved by a method of supervised learning from training datasets. A training dataset that is imbalanced leads undesirably to an inaccurate classifier. In this project, we compare performance of existing methods that tackle the class-imbalance problem. We implement some benchmark classifiers, such as support vector machine, and implement existing methods that tackle class imbalance in Matlab. Then, we estimate accuracy, sensitivity, and specificity of each classifier, under each method that tackles class imbalance. The datasets for learning are labelled and publicly available datasets. The project helps researchers to choose an appropriate method to mitigate a detrimental effect of class imbalance and leads to improvement of classification accuracy.

สารบัญ

บทที่ 1
บทนำ

1.1 ความเป็นมาของงานวิจัย

ในปัจจุบัน วิธีการจำแนกประเภทของข้อมูลนั้นมีอยู่หลายรูปแบบด้วยกัน ซึ่งแต่ละวิธีก็จะมีการดำเนินการเป็นรูปแบบเฉพาะเป็นของตนเอง การเลือกใช้วิธีให้เหมาะสมกับชุดข้อมูลจึงเป็นสิ่งที่สำคัญมาก และปัญหาหลักของการจำแนกประเภทของข้อมูล จะมาจากตัวชุดข้อมูลเป็นส่วนใหญ่ และหนึ่งในปัญหาที่พบเจอได้มากที่สุดคือ ความไม่สมดุลของชุดข้อมูล ซึ่งเป็นสาเหตุที่ทำให้ตัวจัดหมวดหมู่ สูญเสียความสามารถในการจำแนกข้อมูล

ดังนั้นคณะผู้ศึกษาจึงต้องการพิจารณาวิธีการเลือกใช้ข้อมูลที่สอดคล้องกับประเภทของข้อมูล พร้อมทั้งเลือกใช้วิธีการแก้ปัญหาความไม่สมดุลของชุดข้อมูลที่สามารถให้ผลลัพธ์ได้ดีที่สุดสำหรับการแก้ปัญหา รวมถึง พัฒนาโปรแกรมที่ใช้จำแนกประเภทข้อมูลที่มีอยู่

1.2 วัตถุประสงค์

1.2.1 พัฒนาโปรแกรมคอมพิวเตอร์เพื่อจำลองวิธีการจำแนกประเภทข้อมูลที่มีอยู่

1.2.2 พัฒนาโปรแกรมคอมพิวเตอร์เพื่อช่วยแก้ไขปัญหาความไม่สมดุลของชุดข้อมูล

1.2.3 เปรียบเทียบวิธีการแก้ไขปัญหาความไม่สมดุลของชุดข้อมูล โดยใช้เกณฑ์ได้แก่ความแม่นยา (Accuracy [8]), ความอ่อนไหว (Sensitivity [8]), ความจำเพาะ (Specificity [8])

1.3 ขอบเขตของงานวิจัย

1.3.1 การเลือกใช้ข้อมูลที่สามารถจำแนกได้เป็น 2 ประเภท (Binary Classification) ในการจำแนกประเภทของข้อมูล ตัวอย่างเช่น ข้อมูลของผู้ที่เป็นโรคมะเร็งและผู้ที่ไม่เป็นโรคมะเร็ง, ข้อมูลของผู้ที่คาดว่าน่าจะมีหนี้กับผู้ที่คาดว่าน่าจะไม่มีหนี้ เป็นต้น

1.3.2 เลือกใช้วิธีการจำแนกประเภทข้อมูลที่เป็นวิธีแบบ Support Vector Machine (SVM) [3]

1.3.3 เลือกใช้ชุดข้อมูลสำหรับทดสอบ เป็นชุดข้อมูลประเภทที่มีอยู่แล้วทั่วไปในสาธารณะ

1.3.4 โปรแกรมที่ใช้ในการทดสอบและพัฒนาวิธีการจำแนกประเภทข้อมูล ได้แก่ โปรแกรม Matlab

1.4 ข้อจำกัด

1.4.1 วิธีการจำแนกประเภทข้อมูลและโปรแกรมที่เลือกใช้ในการสร้างแบบจำลองมีความยาก

1.4.2 ต้องใช้คอมพิวเตอร์ที่มีประสิทธิภาพดี สเปคสูง หน่วยความจำมาก

1.4.3 จำเป็นต้องเลือกใช้ Kernel function ที่ดี

บทที่ 2

ทฤษฎีที่เกี่ยวข้องและรายละเอียดต่างๆ

2.1 ซอฟแวร์ที่เกี่ยวข้อง

ซอฟแวร์ (Software) [4] หมายถึง ชุดคำสั่งหรือโปรแกรมสำหรับใช้สั่งงานหรือควบคุมคอมพิวเตอร์ ให้ทำงานเป็นลำดับขั้นตอน โดยเป็นชุดคำสั่งที่เขียนขึ้นมาจากภาษาคอมพิวเตอร์ ซึ่ง ซอฟแวร์ที่เกี่ยวข้องกับโครงงาน มีดังนี้

Matlab [7] ย่อมาจาก Matrix Laboratory โดย Matlab เป็นภาษาที่ใช้ในการคำนวณทางเทคนิคที่มีประสิทธิภาพสูง รวมถึงสามารถใช้จำลอง ออกแบบและเขียนโปรแกรม ทั้งยังเป็นตัว Software ที่รองรับภาษา สำหรับการเขียนโปรแกรมด้วย

2.2 วิธีการจำแนกประเภท

Classification หมายถึง การจำแนกประเภท โดยในที่นี้จะกล่าวถึงการจำแนกประเภทของข้อมูล หรือ เรียกว่าเทคนิคการจำแนกประเภทข้อมูลซึ่งเป็นกระบวนการสร้างโมเดลสำหรับจัดการข้อมูลให้อยู่ภายใต้เงื่อนไขหรือกลุ่มที่กำหนดมาให้จากกลุ่มตัวอย่างข้อมูลที่เรียกว่าชุดข้อมูลฝึกฝน (Training data) โดยแต่ละแถวของข้อมูลจะประกอบด้วย Field หรือ Attribute จำนวนมาก ซึ่งจะมี Classify Attribute เป็นตัวบ่งชี้ Class ของข้อมูล การจำแนกประเภทข้อ มูลมีจุดประสงค์คือการสร้างโมเดลการแยก Attribute อันหนึ่งโดยขึ้นกับ Attribute อื่นซึ่งโมเดลหรือผลลัพธ์ที่ได้จะสามารถนำไปใช้ได้กับข้อมูลที่ยังไม่เกิดการแบ่งกลุ่มได้ในอนาคต

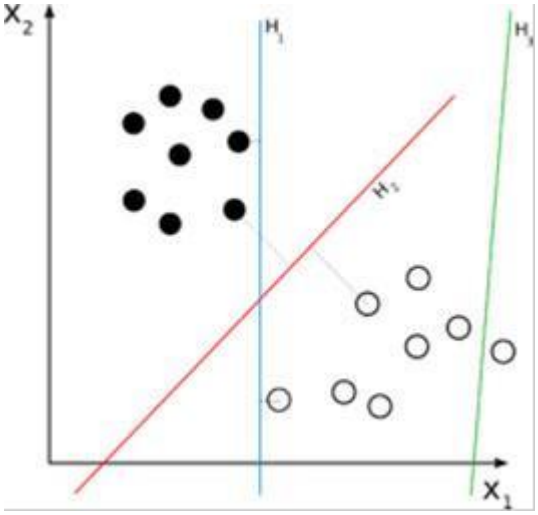
สำหรับเทคนิคในการจำแนกประเภทข้อมูลที่จะเลือกใช้ในวิจัยชิ้นนี้ได้แก่

SVM (Support Vector Machine)

เป็นวิธีการจำแนกประเภทอย่างหนึ่ง ซึ่งอาศัยหลักการของการหาสัมประสิทธิ์ของสมการเพื่อสร้างเส้นที่ใช้ในการแบ่งแยกประเภท (Hyperplane) ของกลุ่มข้อมูลที่ถูกป้อนเข้าสู่กระบวนการสอนระบบให้เกิดการเรียนรู้ โดยแนวคิดก็คือ การนำค่าของกลุ่มมาวางลงบนฟีเจอร์สเปซ (Feature Space) จากนั้นจึงสร้างเส้นที่ใช้ในการแบ่งข้อมูล (Hyperplane) ซึ่งเป็นเส้นตรงขึ้นมา และเลือกใช้เส้นตรงที่สามารถแบ่งกลุ่มของข้อมูลออกจากกันได้ดีที่สุด

การจำแนกข้อมูลที่เป็นข้อมูลหลายมิติ จะมีโครงสร้างในการคัดเลือก (Feature Selection) ในการช่วย เลือกส่วนที่มีความเหมาะสมที่สุด ซึ่งโครงสร้างนี้จะมาจากข้อมูลที่ใช้สอนระบบ โดยจำนวนเซตของโครงสร้าง ที่ใช้อธิบายกรณีหนึ่งเรียกว่า เวกเตอร์ (Vector) ดังนั้นเป้าหมายของ SVM ก็คือ แบ่งแยกกลุ่มของเวกเตอร์ ในกรณีนี้ด้วยหนึ่งกลุ่มของตัวแปรเป้าหมายที่อยู่ข้างหนึ่งของระนาบ

และอีกกลุ่มที่อยู่ระนาบที่ต่างกัน ซึ่ง เวกเตอร์ที่อยู่ข้างระนาบหลายมิติทั้งหมดนี้เรียกว่า ซัพพอร์ตเวกเตอร์ (Support Vectors) [3]



รูปที่ 2.1 : แสดงให้เห็นถึงเส้น Hyperplane จำนวน 3 เส้น ได้แก่ H1 H2 และ H3 (ที่มา : [3])

ตัวอย่างเช่น เราต้องการคัดแยกข้อมูลเป็นสองกลุ่มโดยมีเส้นตรงใช้ในการแบ่ง ซึ่งมีเส้นตรงจำนวนมากที่สามารถแบ่งกลุ่มได้ แต่เส้นตรงเส้นที่ดีสุดเราจะนิยามให้ Margin เป็นผลรวมของระยะห่างของเส้นตรงที่เป็นเส้นแบ่ง ถึงเส้นตรงที่ผ่านข้อมูลที่ใกล้ที่สุดและขนานกับเส้นแบ่งของทั้งสองกลุ่ม ซึ่งจากรูปจะเห็นได้ว่า H1 จะสามารถแบ่งข้อมูลทั้งสองออกได้เช่นกัน แต่ระยะจากเส้นแบ่งไปถึงเส้นตรงที่ใกล้ข้อมูลนั้นมีขนาดน้อย แต่เส้น H2 จะเป็นเส้นแบ่งกลุ่มที่กว้างมากที่สุดของทั้งสองกลุ่ม และให้ค่าเป็น Maximum margin เราเรียกข้อมูล บน Margin นี้ว่า Support Vector

2.3 ปัญหาความไม่สมดุลในการจำแนกประเภท

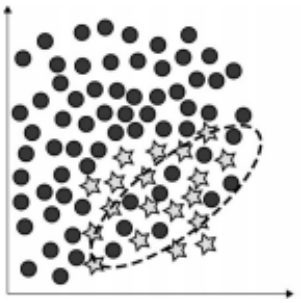
2.3.1 ปัญหาของชุดข้อมูลที่ไม่สมดุล

ชุดข้อมูลที่ไม่สมดุล ในที่นี้จะกล่าวถึงข้อมูลที่สามารถจำแนกได้เป็น 2 ประเภท (Binary Classification) ซึ่งการที่เกิดความไม่สมดุล หมายถึง การที่ตัวอย่างข้อมูลที่เป็นตัวแทนของข้อมูลประเภทหนึ่งมีจำนวนน้อยกว่าตัวอย่างข้อมูลที่เป็นตัวแทนของข้อมูลอีกประเภทหนึ่ง ตัวอย่างเช่น ข้อมูลประเภท A มีจำนวนตัวอย่างข้อมูล 10 ตัวอย่าง ข้อมูลประเภท B มีจำนวนตัวอย่างข้อมูล 10000 ตัวอย่าง จะเห็นได้ว่าอัตราส่วนของข้อมูล A : B มีจำนวนต่างกันถึง 1000 เท่า ซึ่งความไม่สมดุลของชุดข้อมูลนี้ ทำให้ตัวจำแนกประเภทข้อมูลนั้นมองชุดข้อมูลที่มีจำนวนตัวอย่างน้อยกว่าเป็นสัญญาณรบกวนซึ่งทำให้เกิดการผิดพลาด

จากการจำแนกประเภทกับข้อมูลที่มีจำนวนตัวอย่างน้อยอยู่บ่อยเสมอ ซึ่งปัญหาที่กล่าวมา เกิดจาก 3 สาเหตุหลักๆ ได้แก่

1) ตัวอย่างที่มีจำนวนน้อย

โดยปกติแล้ว ชุดข้อมูลที่ไม่สมดุลมักจะมีตัวอย่างของข้อมูลประเภทหนึ่งน้อย ซึ่งส่งผลให้เกิดการเรียนรู้ที่ดีกว่าจากรูปแบบการจำแนกประเภทที่ถูกกำหนดโดยชุดข้อมูลที่มีจำนวนมากกว่า

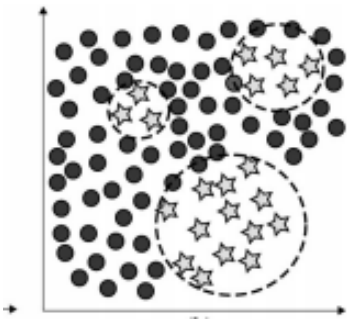


2) การซ้อนทับกันหรือการแยกประเภท

รูปที่ 2.2 : แสดงการซ้อนทับกันของตัวอย่าง (Overlapping) (ที่มา : [2])

เมื่อเกิดปัญหาเหล่านี้ขึ้น จะเป็นเรื่องยากในการใช้กฎของการแยกประเภท ซึ่งส่งผลให้เกิดการจำแนกประเภทที่ผิดพลาดกับตัวอย่างที่มีจำนวนน้อย แต่ถ้าไม่เกิดการซ้อนทับกันของข้อมูลตัวจำแนกประเภทสามารถเรียนรู้การจำแนกประเภทที่เหมาะสมได้โดยไม่คำนึงถึงการกระจายตัวของข้อมูล

3) Small disjunct



รูปที่ 2.3 : แสดงการเกิด Small disjuncts (ที่มา : [2])

เกิดขึ้นได้เมื่อ Concepts ของตัวอย่างที่มีจำนวนน้อย ถูกทำให้อยู่ในรูป Subconcepts [2] ซึ่งการเกิด Small disjuncts ยังทำให้เกิดความซับซ้อนที่เพิ่มขึ้น อันเนื่องมาจากจำนวนตัวอย่างที่ไม่สมดุลกัน

2.3.2 การประเมินประสิทธิภาพของ Domain ที่ไม่สมดุล

การประเมินผลนั้นเป็นตัวช่วยในการสร้างโมเดลและการพิจารณาประสิทธิภาพของตัวจำแนกประเภท อย่างไรก็ตาม ในชุดข้อมูลที่ไม่สมดุลนั้น ความถูกต้อง (Accuracy) จึงไม่ใช่ตัวชี้วัดที่เหมาะสม เช่น การที่ตัว จำแนกประเภทสามารถจำแนกได้อย่างถูกต้องถึง 90 % แต่มีค่า IR (Imbalance Ratio = Negative class instances (ข้อมูลที่มีจำนวนตัวอย่างเยอะ) : Positive class instances (ข้อมูลที่มีจำนวนตัวอย่างน้อย)) [2] เท่ากับ 9 นั่น หมายความว่าการจำแนกจากตัวจำแนกนั้นไม่ถูกต้อง ถ้าจำแนกประเภทของทุกตัวอย่างเป็น Negative class

ตารางที่ 2.1 : Confusion Matrix สำหรับปัญหาแบบ 2 Class (ที่มา : [2])

	Positive prediction	Negative prediction
Positive class	True Positive (TP)	False Negative (FN)
Negative class	False Positive (FP)	True Negative (TN)

$$\text{Acc} = \frac{TP + TN}{TP + FN + FP + TN}.$$

รูปที่ 2.4 : สมการค่าความถูกต้อง (Accuracy) (ที่มา : [2])

1) *True positive rate* $TP_{\text{rate}} = \frac{TP}{TP+FN}$ is the percentage of positive instances correctly classified.
2) *True negative rate* $TN_{\text{rate}} = \frac{TN}{FP+TN}$ is the percentage of negative instances correctly classified.
3) *False positive rate* $FP_{\text{rate}} = \frac{FP}{FP+TN}$ is the percentage of negative instances misclassified.
4) *False negative rate* $FN_{\text{rate}} = \frac{FN}{TP+FN}$ is the percentage of positive instances misclassified.

รูปที่ 2.5 : แสดงหลักการคำนวณค่าจากสมการความถูกต้อง ซึ่งแบ่งออกเป็น 4 ตัวแปร (ที่มา : [2])

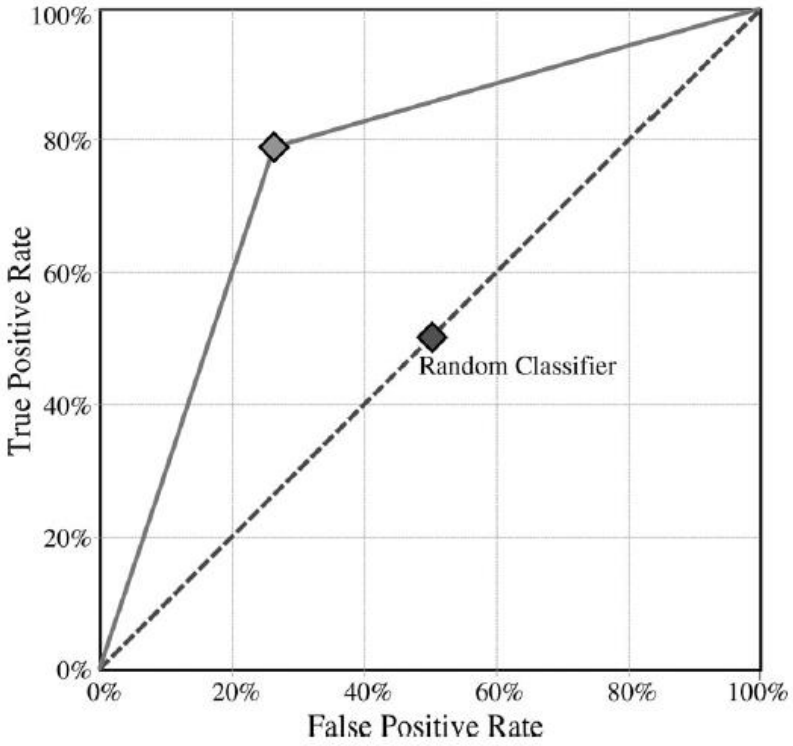
1 : อัตราการจำแนกประเภทที่ถูกต้องสำหรับตัวอย่างที่มีจำนวนน้อย

2 : อัตราการจำแนกประเภทที่ถูกต้องสำหรับตัวอย่างที่มีจำนวนมาก

3 : อัตราการจำแนกประเภทที่ผิดพลาดสำหรับตัวอย่างที่มีจำนวนมาก

4 : อัตราการจำแนกประเภทที่ผิดพลาดสำหรับตัวอย่างที่มีจำนวนน้อย

ต่อมา การหาความสัมพันธ์ระหว่าง Benefits (TP rate) กับ Costs (FP rate) เพื่อประเมินประสิทธิภาพ สามารถใช้กราฟที่มีชื่อว่า ROC (Receiver Operating Characteristic) สำหรับหาค่า Trade-off ระหว่างสองตัวด้านบนได้ ซึ่งพื้นที่ใต้กราฟ หรือ AUC (The Area Under the ROC Curve) จะมีประสิทธิภาพในการประเมินผลมากกว่าการวัดความถูกต้องแบบธรรมดา



รูปที่ 2.6 : ตัวอย่างของการ Plot ROC สำหรับตัวจำแนกประเภท 2 ตัว ซึ่งแสดงผลเป็นจำนวน 2 กราฟ โดยกราฟเส้นประแทนตัวจำแนกประเภทที่เป็นแบบ Random และเส้นตรงแทนตัวจำแนกประเภทที่มีประสิทธิภาพมากกว่าตัวจำแนกประเภทแบบ Random (ที่มา : [2])

จากรูปที่ จะเห็นได้ว่าที่จุด (0,0) และ (1,1) จะเป็นการทำนายที่เป็นไปได้แค่สำหรับตัวอย่างที่มีจำนวนมากและจำนวนน้อยตามลำดับ และที่จุด (0,1) แสดงค่าการจำแนกประเภทที่เหมาะสมที่สุด คือไม่มีความผิดพลาดเลย

$$\mathrm{AUC} = \frac{1 + TP_{\mathrm{rate}} - FP_{\mathrm{rate}}}{2}.$$

รูปที่ 2.7 : แสดงสมการสำหรับการคำนวณค่าของพื้นที่ใต้กราฟ ROC (ที่มา : [2])

2.3.3 การแก้ไขปัญหาความไม่สมดุลของชุดข้อมูล

สามารถแบ่งออกได้เป็น 3 กลุ่มหลัก ซึ่งแต่ละกลุ่มจะมีวิธีการแก้ไขปัญหาที่ต่างกันออกไป ดังนี้

1) Algorithm level approaches (Internal)

เป็นเทคนิคที่ปรับใช้ตัวจำแนกประเภทที่มีอยู่ เพื่อ Bias การเรียนรู้ที่เกี่ยวกับตัวอย่างกลุ่มน้อย หรือตัวอย่างที่มีจำนวนน้อย (Minority Class) ซึ่งเทคนิคนี้ยังต้องการความรู้ที่เกี่ยวกับตัวจำแนกประเภทที่ตรงกัน (Corresponding Classifier) [2] และ ขอบเขตของแอพพลิเคชั่น (Application Domain) [2] เพื่อให้เกิดความเข้าใจว่าทำไมตัวจำแนกประเภทถึงทำงานผิดพลาด เมื่อการกระจายตัวของ Class ไม่สม่ำเสมอ

2) Data level approaches (External)

เป็นเทคนิคที่ทำให้ชุดข้อมูลเกิดการกระจายแบบสมดุล โดยทำการ Resampling data space ซึ่งวิธีนี้จะหลีกเลี่ยงการแก้ไขวิธีการจำแนกประเภทข้อมูล โดยการช่วยลดผลกระทบจากความไม่สมดุลของข้อมูลในช่วงก่อนประมวลผล (Preprocessing) [2] ซึ่งเทคนิคนี้ยังขึ้นอยู่กับวิธีการจำแนกข้อมูลแต่ละประเภทด้วย

3) Cost-sensitive learning framework falls between data and algorithm level approaches

เป็นเทคนิคที่รวมทั้งการแปลง Data level (โดยการเพิ่ม Cost ลงไปในแต่ละตัวอย่าง) และการแก้ไข Algorithm level (โดยแก้ไขกระบวนการเรียนรู้ เพื่อให้เข้ากับ Cost ที่ใส่ลงไป) เข้าด้วยกัน ซึ่งวิธีนี้จะช่วย Bias การจำแนกประเภทของตัวอย่างกลุ่มน้อย (Minority Class) จากนั้นทำการสมมุติ Cost ที่ทำให้จำแนกประเภทได้แย่กว่า, ผิดพลาด (Higher Misclassification) ใส่ลงไปในตัวอย่างนี้ และจะทำการลดข้อผิดพลาดโดยรวมของตัวอย่างทั้ง 2 ประเภท โดยข้อเสียของเทคนิคนี้คือต้องทำการนิยาม Cost สำหรับการจำแนกประเภทที่ผิดพลาด

2.3.4 เทคนิค Preprocessing

เป็นเทคนิคที่ช่วยในเรื่องความสมดุลของการกระจายข้อมูลก่อนการทำงาน ซึ่งเทคนิคการ Resampling นั้นพบว่าสามารถช่วยในเรื่องของผลกระทบของการกระจายของข้อมูล ซึ่งสามารถจัดการกับความไม่สมดุลของข้อมูลได้ อย่างไรก็ตาม เทคนิคการ Resampling สามารถแบ่งออกเป็น 3 กลุ่มหลัก

1) วิธีการแบบ Undersampling

ทำได้โดยการสร้าง Subset จาก Data-set ที่มีโดยการกำจัดตัวอย่างที่มีจำนวนมาก (Negative class instances) ออกไปบางส่วน

2) วิธีการแบบ Oversampling

ทำได้โดยการสร้าง Superset จาก Data-set ที่มีโดยเพิ่มจำนวนของตัวอย่างบางส่วน ด้วยการสร้างขึ้นมาใหม่ หรือก็อปปี้จากตัวอย่างที่มีอยู่

3) วิธีการแบบผสม

เป็นวิธีการที่รวมการกำจัดและสร้างใหม่เข้าด้วยกัน

4) ตัวอย่างของเทคนิคที่สามารถใช้ร่วมกันกับวิธีการแบบ Ensemble ได้

- Random Undersampling

  เป็นการปรับสมดุลการกระจายของข้อมูลโดยการสุ่มกำจัดตัวอย่างที่มีจำนวนมากบางส่วน

- Random Oversampling

  เป็นการปรับสมดุลการกระจายของข้อมูลโดยการสุ่มสร้างตัวอย่างที่มีจำนวนน้อย โดยการเลือกใช้จากตัวอย่างที่มีอยู่บางตัว

- Synthetic Minority Oversampling Technique (SMOTE)

  เป็นเทคนิคแบบ Oversampling ที่จะทำการสร้างตัวอย่างที่มีจำนวนน้อยขึ้นมาใหม่ด้วยการเลือกใช้ค่า k จากตัวที่อยู่ใกล้ที่สุด kNN (k Nearest Neighbos) จากตัวอย่างที่มีและนำไปแทรกตามข้อมูลที่มีอยู่

- Modified Synthetic Minority Oversampling Technique (MSMOTE)

  เป็นเทคนิคที่ปรับปรุงตัว SMOTE ใหม่ โดยการแบ่งกลุ่มของตัวอย่างที่มีจำนวนน้อยออกเป็นสามกลุ่ม ได้แก่ Safe , Border และ Latent Noise และทำการสร้างตัวอย่างใหม่ ซึ่งเงื่อนไขการสร้างใหม่จะแตกต่างกันออกไป ได้แก่ ทำการเลือกค่าจาก kNN, เลือกจากตัวที่อยู่ใกล้ที่สุด และไม่ทำอะไรเลยตามลำดับ

- Selective Preprocessing of Imbalanced Data (SPIDER)

  เป็นการรวมเทคนิคแบบ Local Oversampling ของตัวอย่างที่มีจำนวนน้อย กับ Filtering difficult ของตัวอย่างที่มีจำนวนมากเข้าด้วยกัน โดยแบ่งออกเป็น 2 ส่วน ได้แก่ส่วน Identification ซึ่งจะทำหน้าที่ในการจับข้อมูลให้กลายเป็น Noise ก่อน จากนั้น ส่วนต่อมาคือ Preprocessing จะทำการสร้างข้อมูลเหล่านั้นเป็น 3 ส่วน ได้แก่ Weak, Relabel, Strong ซึ่งทำหน้าที่ Amplifier ข้อมูลที่มีจำนวนน้อย, Amplifier ข้อมูลที่มีจำนวนน้อย และ ทำการ Label ข้อมูลใหม่ และสุดท้ายคือ Strongly Amplifier ตัวอย่างที่มีจำนวนน้อย เมื่อเสร็จกระบวนการเหล่านี้แล้ว ตัวอย่างที่เปรียบเสมือนสัญญาณรบกวนในข้อมูลที่มีจำนวนมากจะถูกกำจัดไปจากชุดข้อมูล

2.4 การทำเหมืองข้อมูล (Data Mining)

Data Mining [5] หรือ การทำเหมืองข้อมูล คือ กระบวนการที่กระทำกับชุดข้อมูลจำนวนมาก เพื่อค้นหารูปแบบและความสัมพันธ์ที่อยู่ในชุดข้อมูลนั้น ซึ่งเปรียบเสมือนวิวัฒนาการในการจัดเก็บ และ

ตีความหมายข้อมูลโดยเป็นการจัดเก็บแบบทำฐานข้อมูลที่สามารถถึงข้อมูลออกมาใช้จนถึงการที่สามารถพบความรู้ใหม่ๆ ที่ซ่อนอยู่ในข้อมูล

สำหรับเทคนิคในการทำ Data Mining จะมีอยู่ด้วยกัน 3 เทคนิคหลักๆที่ต้องใช้ ได้แก่

2.4.1 กฎความสัมพันธ์ (Association rule)

แสดงความสัมพันธ์ของเหตุการณ์หรือวัตถุที่เกิดขึ้นพร้อมกัน เช่น การวิเคราะห์ข้อมูลการขายสินค้าโดยเก็บข้อมูลจากระบบ ณ จุดขาย (POS) เช่น ถ้าพบว่าคนซื้อเทปวิดิโอมักจะซื้อเทปกาวด้วย ร้านค้าก็อาจจะจัดให้สินค้าทั้งสองชนิดอยู่ใกล้กัน เพื่อประโยชน์ที่จะเกิดขึ้น เป็นต้น

2.4.2 การจำแนกประเภทข้อมูล (Data classification)

หากฎเพื่อระบุประเภทของวัตถุจากคุณสมบัติของวัตถุต่างๆ เช่น ความสัมพันธ์ระหว่างผลตรวจร่างกาย กับการเกิดโรค ที่สัมพันธ์กัน โดยใช้ข้อมูลของผู้ป่วยและการวินิจฉัยของแพทย์ที่เคยวินิจฉัยไว้เพื่อช่วยนำมาวินิจฉัยโรคของผู้ป่วย เป็นต้น

2.4.3 การแบ่งกลุ่มข้อมูล (Data clustering)

แบ่งข้อมูลที่มีลักษณะคล้ายกันออกเป็นกลุ่ม เช่น กลุ่มผู้ป่วยที่เป็นโรคเดียวกันตามลักษณะอาการเพื่อนำไปใช้วิเคราะห์หาสาเหตุของโรคจากอาการที่คล้ายคลึงกันของผู้ป่วย เป็นต้น

2.5 ต้นไม้ตัดสินใจ (Decision Tree) [9]

ต้นไม้การตัดสินใจจะทำการจัดกลุ่ม (Classify) ชุดข้อมูลนำเข้าในแต่ละกรณี (Instance) แต่ละบัพ (node) ของต้นไม้การตัดสินใจคือตัวแปร (Attribute) ต่างๆของชุดข้อมูล เช่นหากต้องการตัดสินใจว่าจะไปเล่นกีฬาหรือไม่ก็จะมีตัวแปรต้นที่จะต้องพิจารณาคือ ทัศนียภาพ ลม ความชื้น อุณหภูมิ เป็นต้น และมีตัวแปรตาม ซึ่งเป็นผลลัพธ์จากต้นไม้คือการตัดสินใจว่าจะไปเล่นกีฬารึเปล่า ซึ่งแต่ละตัวแปรนั้นก็จะมีค่าของตัวเอง (Value) เกิดเป็นชุดของตัวแปร-ค่าของตัวแปร (Attribute-Value Pair) เช่น ทัศนียภาพเป็นตัวแปรก็อาจมีค่าได้เป็นฝนตก แดดออก หรือการตัดสินใจว่าจะไปเล่นกีฬารึเปล่านั้นก็อาจมีค่าได้เป็นใช่ กับ ไม่ใช่ เป็นต้น

การทำนายประเภทด้วยต้นไม้ตัดสินใจ จะเริ่มจากบัพราก โดยทดสอบค่าตัวแปรของบัพ แล้วจึงตามกิ่งของต้นไม้ที่กำหนดค่าเพื่อไปยังบัพลูกถัดไป การทดสอบนี้จะกระทำไปจนกระทั่งเจอบัพใบซึ่งจะแสดงผลการทำนาย เช่น ต้นไม้ตัดสินใจนี้ใช้ทำนายว่าจะเล่นกีฬาหรือไม่ โดยพิจารณาจากลักษณะอากาศของวันนั้น โดยวัตถุที่ต้องการทำนายประเภท ประกอบด้วยลักษณะหรือตัวแปร 3 ตัว ได้แก่ ทัศนียภาพ ความชื้น และ กระแสลม ดังนั้น ถ้ากำหนดวันวันหนึ่งมีคุณลักษณะแสดงเป็นเวกเตอร์ เช่น [สภาพอากาศ = แดดออก, ความชื้น = สูง] การทำนายว่าจะเล่นกีฬาหรือไม่ จะเริ่มจากบัพราก โดยทดสอบค่าตัวแปร

"สภาพอากาศ" ซึ่งมีค่าเท่ากับ "แดดออก" จึงไปทดสอบค่าตัวแปร "ความชื้น" ในบัพถัดไป ทำให้ได้ประเภทของวันนี้คือ "ไม่เล่นกีฬา" เป็นต้น

2.6 เทคนิค Ensemble

Ensemble technique [2] คือเทคนิคที่ใช้โมเดล classification จำนวนมากมาช่วยในการหาคำตอบ ซึ่งเป็นเทคนิคที่มีประสิทธิภาพสูง โดยเทคนิค Ensemble ใน Machine Learning ที่เป็นที่นิยมกันมากซึ่งมีอยู่ 2 วิธีด้วยกัน ได้แก่

2.6.1 Bagging หรือ Bootstrap Aggregation

Bagging เป็นพื้นฐานของอัลกอริทึมที่นิยมใช้กัน ได้แก่ Random Forest Classifier [6] ใน Scikit-learn library โดยคำว่า Bagging ย่อมาจาก “bootstrap aggregation” ซึ่งในทางสถิติ boostrap คือการสุ่มข้อมูลมาจากข้อมูลประชากร เพื่อใช้คำนวณค่าทางสถิติของประชากรกลุ่มเล็กๆที่เลือกสุ่มออกมา และ aggregation คือการนำ มารวมกัน ดังนั้น Bagging จึงหมายถึงการสุ่มตัวอย่างข้อมูลออกมาแล้วสร้างตัวจำแนกประเภทขึ้นมา สำหรับวิธีการสุ่มข้อมูลสำหรับใช้งานจะใช้วิธีสุ่มแบบแทนที่ (random with replacement) ซึ่งหมายถึงข้อมูลที่มีก็ยังคงอยู่เหมือนเดิม ไม่ได้ลดลงหลังจากการสุ่ม ซึ่งสามารถสุ่มข้อมูลหลายๆรอบเพื่อให้ได้ตัวจำแนกประเภทหลายๆตัว จากนั้น เมื่อต้องการทำนายก็จะใช้งานตัวจำแนกประเภททุกตัวที่สร้างขึ้นมาเพื่อทำนายชุดข้อมูลใหม่ที่พบ ซึ่งมีการทำนายอยู่หลายแบบ ได้แก่การเฉลี่ยหรือการโหวต (เป็นการใช้ชุดข้อมูลฝึกฝน (training data) ชุดเดียวกัน แต่สร้างโมเดลด้วยเทคนิคต่างกัน) แล้วแต่ว่าต้องการที่จะทำนายความน่าจะเป็นหรือทำนายประเภทข้อมูล หรืออาจกล่าวโดยสั้นว่าเป็นการสุ่มชุดข้อมูลฝึกฝนให้เป็นหลายชุด แต่สร้างโมเดลด้วยเทคนิคเดียวกันทั้งหมด และนอกจากสุ่มข้อมูลแล้วยังสามารถสุ่ม features ของข้อมูลได้เช่นกัน วิธีการสุ่มข้อมูลนี้ทำให้ได้โมเดลหลายๆ ตัวมาช่วยทำนาย ซึ่งสามารถลดแนวโน้มที่โมเดลจะ overfitting [1] ข้อมูลอีกชุดที่แตกต่างกัน เช่นข้อมูลมี 20 features (20 columns) เราก็สามารถสุ่มข้อมูลออกมาโดยใช้ features เพียงครึ่งเดียวก็สามารถทำได้

ซึ่งสังเกตุว่าถ้าทำนายผลของข้อมูลด้วย decision tree เพียง 1 โมเดล โอกาสที่จะเกิด overfitting มีสูงมากเนื่องจากโมเดลนี้ถูกทดสอบมาเพื่อทำนายข้อมูลที่ใส่เข้าไป แต่อาจจะทำนายไม่สำเร็จสำหรับข้อมูลอื่นๆได้ ดังนั้นการนำ decision tree มารวมกันเป็นแบบ Random Forest สามารถช่วยลดโอกาสที่จะเกิด overfitting ได้

โดย Overfitting เป็นอีกหนึ่งปัญหาพื้นฐานที่พบบ่อยมากในการพัฒนา Algorithm สำหรับ Machine Learning ทำให้เกิดเหตุการณ์ที่ โมเดลทำงาน (เช่น ทำนายข้อมูล) ได้ดีมากกับชุดข้อมูลฝึกฝน (Training data/ in-sample data) แต่เมื่อไหร่ก็ตามที่เรานำโมเดลนั้นมาทำงานกับชุดข้อมูลทดสอบ (testing data/out-

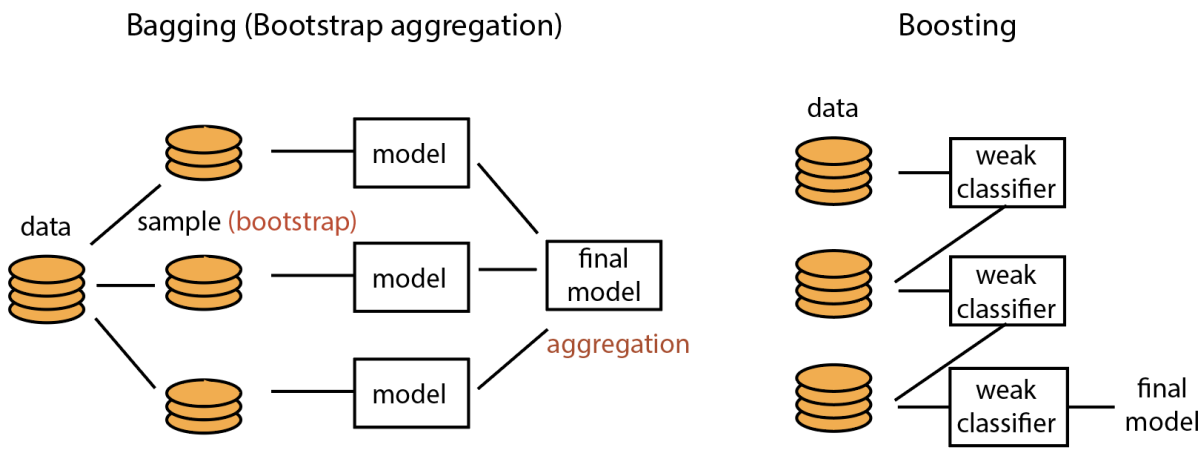
sample data) ซึ่งเป็นข้อมูลที่โมเดลไม่เคยเห็นมาก่อน โมเดลกลับทำงานแต่แย่มากเพราะสามารถรู้จำได้เพียงข้อมูลที่ใส่เข้าไป เท่านั้น

2.6.2 Boosting

Boosting เป็นพื้นฐานของ AdaBoost [2] หรือ Gradient Boosting ใน Library เช่น XGBoost และ LightGBM [6] ซึ่งเป็นการนำตัวจำแนกประเภทที่มีความแม่นยำต่ำ (Weak Classifier) มาทำนายข้อมูลที่มีอยู่ จากนั้นจะให้ตัวจำแนกประเภทที่มีความแม่นยำต่ำตัวใหม่มาแก้ไขข้อผิดพลาดที่เกิดขึ้น โดยผลรวมของตัวจำแนกประเภทจะทำให้เกิดเป็นตัวจำแนกประเภทตัวใหม่ขึ้นมา ซึ่งจะทำแบบนี้ไปเรื่อยๆจนได้โมเดลที่ดีที่สุด จากผลรวมของตัวจำแนกประเภท

ซึ่งภาพรวมการทำงานของ Boosting นั้นเปรียบเมือนการทำงานเป็นทีมโดยการเอาตัวจำแนกประเภทที่คุณภาพต่ำมากมารวมกันจนสามารถทำนายข้อมูลที่ซับซ้อนมากๆได้ โดยข้อเสียของการใช้เทคนิค Boosting ก็คือต้องทำงานหลายๆครั้งและเป็นลำดับจนกว่าจะได้โมเดลที่ตรงตามความต้องการ ต่างจากเทคนิค Bagging ที่สามาถสุ่มข้อมูลได้แล้วทำการฝึกฝนโมเดลได้พร้อมๆกัน แต่ข้อดีของ library เช่น XGBoost ที่สามารถเลือกชนิดของตัวจำแนกประเภทที่มีความแม่นยำต่ำ นี้ได้ ไม่ว่าจะเป็นแบบต้นไม้ (Tree) หรือแบบเส้นตรง (Linear) และในหลายๆครั้งนั้นเทคนิค Boosting สามารถทำนายข้อมูลที่มีความซับซ้อนมากๆได้มากกว่าการใช้เทคนิค Bagging

รูปที่ 2.8 : แสดงความแตกต่างระหว่างเทคนิค Bagging และ Bossting (Machine Learning) (ที่มา : [6])



2.6.3 Bagging vs Boosting

ทั้งสองเทคนิคเป็นการช่วยทำให้ได้โมเดลที่มีความแม่นยำและความเสถียรต่อการเจอข้อมูลใหม่ๆ แต่ทั้งสองวิธีก็มีจุดประสงค์แตกต่างกันไป โดยถ้าต้องการฝึกฝนโมเดลเดียวเช่น Decision tree แล้วยังเกิดการ Overfit ข้อมูลอยู่ (Variance มีค่าสูง) การเลือกใช้เทคนิค Bagging ค่อนข้างจะแก้ปัญหาได้มากที่สุด เนื่องจากการสร้างโมเดลหลายๆ โมเดลมาฝึกฝนชุดข้อมูลแบบสุ่มทำให้ไม่ overfit ข้อมูลมากจนเกินไป ส่วนถ้าทำการ Fit โมเดลแล้วยังไม่ได้ความแม่นยำเท่าที่ควรและต้องการเพิ่มความแม่นยำ การใช้เทคนิค Boosting สามารถสร้างตัวจำแนกประเภทที่มีความแม่นยำสูงได้

ซึ่งสามารถสรุปได้ว่า เทคนิค Bagging ช่วยแก้ปัญหาการเกิด overfitting และเทคนิค Boosting ช่วยแก้ ปัญหาความแม่นยำต่ำ (bias)

บทที่ 3

วิธีดำเนินงาน

3.1 วิธีดำเนินงานเพื่อบรรลุวัตถุประสงค์ข้อที่หนึ่ง

3.3.1 Find Minor and Major classes

เป็นฟังก์ชั่นสำหรับการทำ Class ของตัว Dataset ขึ้นมาใหม่ ซึ่งจะให้ผลลัพธ์เป็น ข้อมูลที่ Labels ขึ้นมาใหม่ โดยมี Syntax ดังนี้ new_labels = find_minor_and_MAJOR(labels) โดย

labels = เมทริกซ์ของ lables ดั้งเดิม ซึ่งมาจากตัว Dataset ตั้งแต่ตอนแรก

new_labels = เมทริกซ์ของ lables หลังจากผ่านฟังก์ชั่น ซึ่งมีข้อกำหนดไว้ดังนี้

- -1 class : Majority Class (สำหรับตัวอย่างจำนวนมาก)
- +1 class : Minority Class (สำหรับตัวอย่างจำนวนน้อย)

3.1.2 Correct class imbalance by a sampling method

เป็นฟังก์ชันที่ทำหน้าที่ปรับสมดุลการกระจายของข้อมูลโดยมี

Input(samples, labels, method)

sample, เป็นเมทริกซ์ที่ประกอบด้วยแถว Samples และหลัก Features

labels, เป็นเซ็ตของข้อมูลที่มีข้อมูลเป็น {-1 , +1} มีจำนวนเท่ากับ sample

โดยที่ labels(i) = sample(i , :)

method, เป็นตัวแปร String ที่ใช้เลือกเทคนิค Preprocessing ฟังก์ชัน Correct class imbalance ได้แก่ 'UnderSampling', 'OverSampling' และ 'Hybrid'

Output(new_sample, new_labels)

new_sample, เป็นเมทริกซ์หลังผ่านเทคนิค Preprocessing ที่ประกอบด้วยแถว Samples และหลัก Features

new_labels, เป็นเซ็ตของข้อมูลหลังผ่านเทคนิค Preprocessing ที่มีข้อมูลเป็น {-1 , +1} มีจำนวนเท่ากับ new_sample

3.1.3 Training an SVM Classifier

การเทรนโดยคำสั่งหลักคือ fitcsvm โดยปกติแล้วมีSyntax ดังนี้

```
SVMModel = fitcsvm(X,Y,'KernelFunction','rbf',...
 'Standardize',true,'ClassNames',{'negClass','posClass'});
```

อินพุตคือ:

X – เมทริกซ์ของข้อมูลตัวทำนายซึ่งแต่ละแถวนับเป็นหนึ่งการสังเกตและแต่ละคอลัมน์เป็นหนึ่งตัวทำนาย

Y - อาเรย์ของคลาสเลเบลที่มีแต่ละแถวที่สอดคล้องกับค่าของแถวที่เกี่ยวข้องใน X. Y สามารถเป็นหมวดหมู่อักขระหรืออาร์เรย์สตริงเวกเตอร์เชิงตรรกะหรือตัวเลขหรือเซลล์อาร์เรย์ของเวกเตอร์อักขระได้

KernelFunction - ค่าเริ่มต้นคือ 'เชิงเส้น' สำหรับการเรียนรู้สองระดับซึ่งแยกข้อมูลด้วยไฮเปอร์เพลน ค่า 'gaussian' (หรือ 'rbf') เป็นค่าเริ่มต้นสำหรับการเรียนรู้แบบหนึ่งคลาสและระบุให้ใช้เคอร์เนล Gaussian (หรือ ฟังก์ชันพื้นฐานของรัศมี) ขั้นตอนสำคัญในการฝึกอบรมตัวจำแนก SVM ให้ประสบความสำเร็จคือเลือกฟังก์ชันเคอร์เนลที่เหมาะสม

Standardize – ตั้งค่าสถานะระบุว่าซอฟต์แวร์ควรสร้างมาตรฐานให้กับผู้ทำนายก่อนฝึกอบรมตัวจำแนก

ClassNames - แยกความแตกต่างระหว่างคลาสที่เป็นลบและบวกหรือระบุคลาสที่จะรวมในข้อมูล คลาสลบคือองค์ประกอบแรก (หรือแถวของอาร์เรย์อักขระ) เช่น 'negClass' และคลาสบวกคือองค์ประกอบที่สอง (หรือแถวของอาร์เรย์อักขระ) เช่น 'posClass' ClassNames ต้องเป็นชนิดข้อมูลเดียวกับ Y เป็นวิธีปฏิบัติที่ดีในการระบุชื่อคลาสโดยเฉพาะอย่างยิ่งหากกำลังเปรียบเทียบประสิทธิภาพของตัวแยกประเภทที่แตกต่างกัน แบบจำลองที่ได้รับการฝึกอบรม (SVMModel) มีพารามิเตอร์ที่ปรับให้เหมาะสมที่สุดจากอัลกอริทึม SVM ช่วยให้สามารถจำแนกข้อมูลใหม่ได้

3.1.4 Classifying New Data with an SVM Classifier

การจำแนกข้อมูลใหม่โดยมีคำสั่งหลักคือ predict ดึงค่าจาก SVMModel มาใช้โดยปกติแล้วมี Syntax ดังนี้

```
[label,score] = predict(SVMModel,newX);
```

ผลลัพธ์เวกเตอร์, เลเบล, แสดงถึงการจำแนกประเภทของแต่ละแถวใน X.score เป็นเมทริกซ์ n-by-2 ของ soft score แต่ละแถวสอดคล้องกับแถวใน X ซึ่งเป็นการสังเกตใหม่ คอลัมน์แรกมีคะแนนสำหรับการสังเกตที่ถูกจำแนกประเภทใน Class ลบและคอลัมน์ที่สองประกอบด้วยการสังเกตคะแนนที่จัดอยู่ใน Class บวก

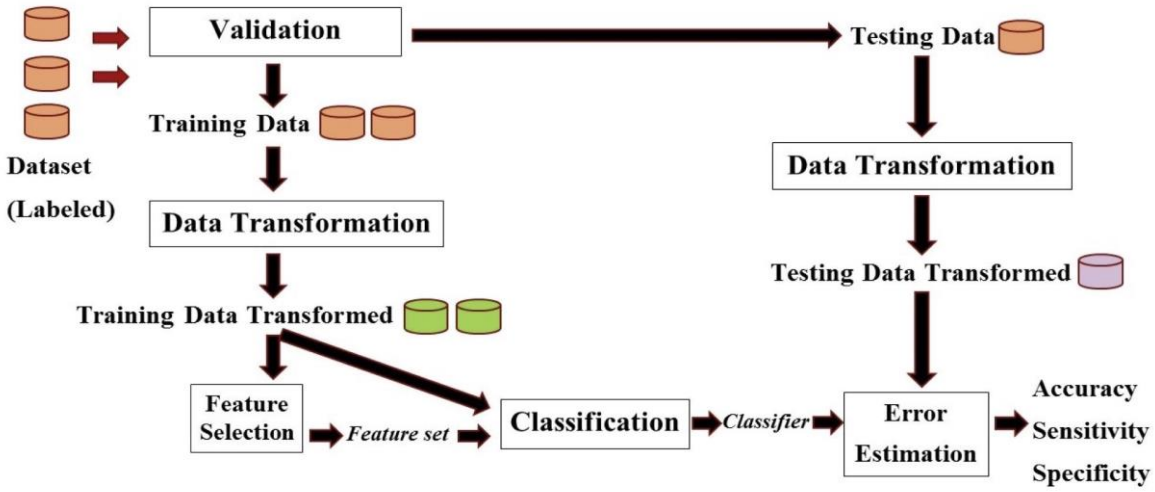
ในการประมาณความน่าจะเป็นด้านหลังมากกว่าคะแนน ก่อนอื่นให้ผ่านตัวจำแนก SVM (SVMModel) ที่ผ่านการฝึกอบรมมาให้กับ fitPosterior ซึ่งเหมาะกับฟังก์ชั่นการเปลี่ยนความน่าจะเป็นแบบคะแนนต่อหลัง (score-to-posterior-probability) Syntax คือ:

ScoreSVMModel = fitPosterior(SVMModel,X,Y);

คุณสมบัติScoreTransform ของตัวแยกประเภท ScoreSVMModel มีฟังก์ชันการแปลงที่เหมาะสมที่สุดผ่าน ScoreSVMModel เพื่อทำนายแทนที่จะส่งกลับคะแนนคะแนนอาร์กิวเมนต์เอาท์พุทมีความน่าจะเป็นหลังของการสังเกตที่ถูกจัดประเภทในคลาสเชิงลบ (คอลัมน์ที่ 1 ของคะแนน) หรือบวก (คอลัมน์ 2 ของคะแนน)

3.1.5 Tuning an SVM Classifier

ใช้อาร์กิวเมนต์คู่ค่าชื่อ 'OptimizeHyperparameters' ของ fitcsvm เพื่อค้นหาค่าพารามิเตอร์ที่ลดการสูญเสียการตรวจสอบความถูกต้องข้าม พารามิเตอร์ที่มีสิทธิ์คือ 'BoxConstraint', 'KernelFunction', 'KernelScale', 'PolynomialOrder' และ 'Standardize' ตัวอย่างเช่นดูที่การปรับ SVM Classifier ให้เหมาะสมที่สุด โดยใช้การBayesian Optimization หรือสามารถใช้ฟังก์ชั่น bayesopt ดังที่แสดงในออปติไมซ์ลอจิคัล SVM ลักษณนามข้ามการตรวจสอบความถูกต้องโดยใช้ bayesopt ฟังก์ชัน bayesopt ให้ความยืดหยุ่นในการปรับแต่ง การปรับแต่งให้เหมาะสมยิ่งขึ้น โดยสามารถใช้ฟังก์ชัน bayesopt เพื่อปรับพารามิเตอร์ให้เหมาะสม รวมถึงพารามิเตอร์ที่ไม่มีสิทธิ์ปรับให้เหมาะสมเมื่อใช้ฟังก์ชัน fitcsvm



รูปที่ 3.1 แสดงขั้นตอนการสร้างแบบจำลองจำแนกประเภทของ SVM

ซึ่ง ในส่วนของเทคนิคการจัดการกับความไม่สมดุลของข้อมูล สามารถใส่เทคนิคนี้ได้ 2 ตำแหน่ง คือ อยู่กับ Data Transformation และ อยู่ก่อน Classification ซึ่งจะมีกล่องใหม่ขึ้นมา เป็นกล่องที่ชื่อว่า “Class Imbalance Migitation” แทรกเข้ามา

จากการศึกษาจากงานวิจัยที่ใช้อ้างอิงในการทำ พบว่าขั้นตอนการจัดการกับความไม่สมดุลของข้อมูลจะเริ่มกระทำที่ตำแหน่งก่อน Classification เนื่องจากมีการใช้ Ensemble เทคนิคที่เป็นการรวมการจำแนกประเภทเข้าด้วยกันเช่นBagging และ Boosting เป็นต้น และในแต่ละการจำแนกประเภทก็มีการจัดการของข้อมูลก่อนนำมาเทรนต่างกันเช่น Bagging ก็จะทำในส่วนของ Preprocessing ที่ Data Transformation ด้วยที่จะ bootstrap ข้อมูลเป็นหลายส่วนก่อนแล้วนำไปเทรนในหลายโมเดลในเวลาเดียวกันเพื่อลดความไม่สมดุลในการ จำแนกประเภทได้ ส่วน Boosting ก็จะเน้นไปที่การรวมตัวจำแนกประเภทที่ไม่ดีแล้วปรับแก้ไปจนสามารถจัดการกับความไม่สมดุลในการจำแนกประเภท

3.1.6 Evaluate performance of a binary classifier

เป็นฟังก์ชั่นสำหรับการคำนวณค่า Accuracy, Sensitivity และ Specitivity[ก] ออกมาเป็น Output โดยมี Parameter (actual_lables, predict_lables) ซึ่งใช้การคำนวณจากข้อมูลจริงและข้อมูลจากการทำนายของตัวโมเดล ได้แก่ อัตราการทำนายถูกและผิด ของคลาส +1 (True positive rate, False positive rate) อัตราการทำนาย ถูกและผิดของคลาส -1 (True negative rate, False negative rate)

3.1.7 Mean and Standard Deviation

ใช้สำหรับการหาค่าเฉลี่ย และหาส่วนเบี่ยงเบนมาตรฐาน เพื่อหาการกระจัดกระจายของข้อมูลทำให้ ง่ายต่อการบอกว่าข้อมูลที่ได้ มีความเป็นกลุ่มก้อนแค่ไหน ซึ่งมีคำสั่งดังนี้

ค่าเฉลี่ย : mean(A) โดยผลลัพธ์จะเป็นการหาค่าเฉลี่ยของอาเรย์ A ทุกตัว โดยขนาดของ A จะต้องมีค่า ไม่เท่ากับ 1

ส่วนเบี่ยงเบนมาตรฐาน : std(A) ผลลัพธ์ที่ได้จะเป็นการหาส่วนเบี่ยงเบนมาตรฐานของอาเรย์ A โดย ขนาดของ A จะต้องมีค่าไม่เท่ากับ 1

3.2 วิธีดำเนินงานเพื่อบรรลุวัตถุประสงค์ข้อที่สอง

3.2.1 เลือกใช้ข้อมูลที่ต้องการจำแนกประเภท โดยใช้ข้อมูลที่ไม่สมดุลประเภท Binary classification {-1, +1} ซึ่ง Negative เป็น Majority และ Positive เป็น Minority

3.2.2 ทำการเลือกใช้วิธีที่เหมาะสมกับชุดข้อมูลที่เลือกใช้ โดยใช้วิธีดังนี้

- วิธีการแบบ Undersampling

  เป็นการปรับสมดุลการกระจายของข้อมูลด้วยการกำจัดตัวอย่างที่มีจำนวนมากออกไปบางส่วน

- วิธีการแบบ Oversampling

  เป็นการปรับสมดุลการกระจายของข้อมูลด้วยการเพิ่มตัวอย่างที่มีจำนวนน้อยจากตัวอย่างที่มีอยู่ขึ้นมาบางส่วน

- วิธีการแบบ Hybrid คือการรวมวิธีระหว่าง Undersampling และ Oversampling เข้าด้วยกัน

3.3 วิธีดำเนินงานเพื่อบรรลุวัตถุประสงค์ข้อที่สาม

3.3.1 เปรียบเทียบผลลัพธ์ของแต่ละเทคนิค โดยใช้เกณฑ์ที่กำหนด

- กรณีที่ 1 เปรียบเทียบโดยใช้ค่าเฉลี่ย (Mean)

  คือการสร้างตัวควบคุมตัวแปรสุ่มที่ใช้เมล็ดสามารถกำหนดค่าสุ่มได้ขึ้นมาในแต่ละครั้งของการเก็บค่าเพื่อให้ได้ผลลัพธ์แตกต่างกันในแต่ละค่าเมล็ดจากนั้นจึงนำผลลัพธ์มาหาค่าเฉลี่ย

- กรณีที่ 2 เปรียบเทียบโดยใช้ส่วนเบี่ยงเบนมาตรฐาน (Standard Deviation)

  คือการสร้างตัวควบคุมตัวแปรสุ่มที่เราสามารถกำหนดได้ขึ้นมาในแต่ละครั้งของการเก็บค่าเพื่อให้ได้ผลลัพธ์แตกต่างกันจากนั้นจึงนำมาหาส่วนเบี่ยงเบนมาตรฐาน

- กรณีที่ 3 เปรียบเทียบโดยวิธีทางสถิติ เรียกว่า t test on two population means

  คือการทดสอบความแตกต่างระหว่างค่ากลางของสองประชากรที่มีการกระจายแบบปกติและอิสระต่อกัน

3.3.2 วิเคราะห์ข้อได้เปรียบ/เสียเปรียบของผลลัพธ์ที่ได้ในแต่ละเทคนิคที่ใช้

3.3.3 ทำการสรุปผลเพื่อให้ได้แนวทางการเลือกใช้เทคนิคที่เหมาะสมที่สุด

บทที่ 4

ผลการดำเนินงาน

4.1 ผลการดำเนินงานเพื่อบรรลุวัตถุประสงค์ข้อที่หนึ่ง

จากการศึกษาวิธีการจำแนกประเภทข้อมูล พบว่าปัจจุบันมีวิธีการจำแนกประเภทข้อมูลที่นิยมใช้ ได้แก่ Logistic Regression, Support Vector Machine, Naïve Bayes Classifier, Decision Trees, Random Forest, Neural Network, K Nearest Neighbor และ Stochastic Gradient Descent ซึ่งในที่นี้จะเป็นการเลือกใช้วิธีการจำแนกประเภทข้อมูลแบบ Support Vector Machine (SVM) ซึ่งเป็นวิธีจำแนกประเภทที่เหมาะสมกับข้อมูลแบบ Binary Class ซึ่งเป็นประเภทข้อมูลที่เกี่ยวข้องกับวิจัยชิ้นนี้ และวิธีการจำแนกประเภทข้อมูลอีกวิธีหนึ่งที่น่าสนใจ คือวิธีการแบบ Dynamic Multi-hyper plane Partitioning (DMP) ซึ่งกำลังอยู่ในระหว่างการศึกษา

ในส่วนของวิธีการแก้ปัญหาชุดข้อมูลที่ไม่สมดุล ซึ่งจะแบ่งออกเป็น 2 หลักการ ได้แก่ การแก้ปัญหาแบบ Preprocessing และการแก้ปัญหาที่ตัว Algorithm ของตัวจำแนกประเภทข้อมูล ซึ่งทางผู้จัดทำได้ทำการทดลองแก้ปัญหาแบบ Preprocessing ก่อน เนื่องจากมีความซับซ้อนน้อยกว่า และสามารถพัฒนาต่อไปได้ โดยวิธีการที่เลือกใช้แก้ปัญหาหลักๆจะมีอยู่ 3 วิธี ได้แก่

- Under Sampling method
- Over Sampling method
- Hybrid method

และนอกจากนี้ ยังมีวิธีการแก้ปัญหาความไม่สมดุลของชุดข้อมูลที่อยู่ในช่วงของการศึกษาอยู่อีกทั้งหมด 3 วิธี ได้แก่

- Synthetic minority oversampling technique (SMOTE)
- Modified synthetic minority oversampling technique (MSMOTE)
- Selective preprocessing of imbalanced data (SPIDER)

ซึ่งในส่วนของการออกแบบโปรแกรม จะเป็นการใช้คำสั่งและฟังก์ชั่นหลักๆที่มีอยู่แล้วในตัวโปรแกรมแมทแลป ซึ่งทำให้ได้โมเดลของตัวจำแนกประเภทแบบ SVM ซึ่งตัวโมเดลจะประกอบไปด้วย 3 ส่วนหลักๆ ได้แก่

4.1.1 Classify and Evaluate function

เป็นฟังก์ชั่นที่ใช้สำหรับเรียนรู้ตัวข้อมูลทดสอบมีอินพุทเป็น Samples (Matrix ที่ประกอบด้วยตัว Samples ในแต่ละแถว และ Features ในแต่ละหลัก) และ Labels (Vector ที่มีความยาวเท่ากับตัว Samples ซึ่งเก็บค่า Class(+1,-1) ของแต่ละ Sample) เพื่อให้ได้ผลลัพธ์ออกมาเป็น Predicted Labels (Labels จากการทำนายของตัวจำแนกประเภท) โดยใช้ตัวจำแนกประเภทแบบ SVM และใช้หลักการของ K-fold Cross Validation ในการแบ่งข้อมูลหลักออกเป็นข้อมูลจำนวนมากสำหรับการฝึกฝน (Training Data) และข้อมูลจำนวนน้อยสำหรับการทดสอบ (Testing Data)

4.1.2 Performance Eval function

เป็นฟังก์ชั่นสำหรับการประเมินผลประสิทธิภาพของตัวจำแนกประเภท ซึ่งมีอินพุทเป็น Actual labels (Labels จริงของตัวข้อมูลที่มีอยู่แล้ว) และ Predicted Labels (Labels จากการทำนายของตัวจำแนกประเภท) ซึ่งฟังก์ชั่นจะมีหน้าที่ในการเปรียบเทียบสองค่านี้ โดยคิดจาก [ก] และคำนวณค่าออกมาเป็นผลลัพธ์ ซึ่งได้แก่ Accuracy (ความสามารถในการทำนายได้ตรงกับค่าจริง), Sensitivity (ความสามารถในการทำนาย Class +1 ได้ถูกต้อง) และ Specificity (ความสามารถในการทำนาย Class -1 ได้อย่างถูกต้อง)

4.1.3 Test Classify and Evaluate

เป็น Script สำหรับเรียกใช้งานฟังก์ชั่นทั้งหมด และทำการโหลดตัวชุดข้อมูลเข้ามาในโปรแกรม Matlab จากนั้นทำการเรียกใช้งานฟังก์ชั่นตามลำดับ ได้แก่ ฟังก์ชั่นการแก้ปัญหาความไม่สมดุลของชุดข้อมูล, Classify and Evaluate function และ Performance Eval function ตามลำดับ

4.2 ผลการดำเนินงานเพื่อบรรลุวัตถุประสงค์ข้อที่สอง

จากการศึกษาและทำการโปรแกรมที่ช่วยแก้ไขปัญหาความไม่สมดุลของชุดข้อมูลโดยใช้ภาษา Matlab ในการพัฒนา ทำให้ได้ฟังก์ชั่นที่ใช้สำหรับการแก้ไขความไม่สมดุลของชุดข้อมูลแบบ Preprocessing ด้วยวิธีการทั้ง 3 แบบที่ว่ามา ได้แก่ Over Sampling, Under Sampling และ Hybrid โดยโปรแกรมที่ได้จะเป็นฟังก์ชั่นที่มีชื่อว่า Correct Class imbalance Function ซึ่งมีอินพุทเป็น Samples (Matrix ที่ประกอบด้วยตัว Samples ในแต่ละแถว และ Features ในแต่ละหลัก) และ Labels (Vector ที่มีความยาวเท่ากับตัว Samples ซึ่งเก็บค่า Class(+1,-1) ของแต่ละ Sample) และ Method (วิธีการแก้ไขความไม่สมดุลของชุดข้อมูล เช่น Oversampling, Undersampling, Hybrid) ซึ่งฟังก์ชั่นจะทำหน้าที่จัดการตัวชุดข้อมูล ให้เกิดความสมดุลตามวิธีการแก้ปัญหา และให้ผลลัพธ์ออกมาเป็น New Samples (Samples และ Features หลังจากการแก้ปัญหา

ความไม่สมดุล) และ New Labels (Labels ของ Samples ที่ผ่านการแก้ไขปัญหาความไม่สมดุล) โดยนำฟังก์ชั่น Correct Class imbalance ตัวนี้ ไปใส่ไว้

4.3 ผลการดำเนินงานเพื่อบรรลุวัตถุประสงค์ข้อที่สาม

จากการทดลองด้วยชุดข้อมูลทั้ง 3 ชนิด ได้แก่ Colon, Parkinson และ Haberman ซึ่งมีรายละเอียดตามตารางด้านล่าง

Class ratio (อัตราส่วนระหว่าง Class ที่มีจำนวนมาก ต่อ Class ที่มีจำนวนน้อย)

ตารางที่ 4.1 ลักษณะของแต่ชุดข้อมูล ได้แก่ Feature, Sample, Class ratio และ Application

Dataset	Number of features	Number of samples (Class -1; Class +1)	Class ratio	Application
Parkinson [10]	22 features	195 (Subjects with Parkinson's 147; Healthy 48)	3.6025	Parkinson Prediction
Haberman [11]	3 features	306 (Normal 225; Dead 81)	2.778	Chance to survive after had surgery for breast cancer
Colon [12]	2000 genes	62 (Tumor 40; Normal 22)	1.8	Cancer Prediction

ซึ่งชุดข้อมูลแรกคือ Colon โดย Samples จะประกอบไปด้วย 62 Samples (Row) และ 2000 genes (Column) ซึ่ง Label ของแต่ละ Sample จะประกอบไปด้วย Class -1 (Tumor) จำนวน 40 Samples และ Class +1 (Normal) จำนวน 22 Samples โดยมีค่า Class ratio เท่ากับ 1.8 และมี Application สำหรับการทำนายการเกิดโรคมะเร็ง

ต่อมาคือชุดข้อมูล Parkinsonโดย Samples โดย Samples จะประกอบไปด้วย 195 Samples (Row) และ 22 features (Column) ซึ่ง Label ของแต่ละ Sample จะประกอบไปด้วย Class -1 (Subjects with Parkinson) จำนวน 147 Samples และ Class +1 (Healthy Subject) จำนวน 48 Sample โดยมีค่า Class ratio เท่ากับ 3.6025 และมี Application สำหรับการทำนายการเกิดโรค Parkinson จะประกอบไปด้วย 306 Samples (Row) และ 3 features (Column) ซึ่ง Label ของแต่ละ Sample จะประกอบไปด้วย Class -1 (Normal) จำนวน 225 Samples และ Class +1 (Died) จำนวน 81 Sample โดยมีค่า Class ratio เท่ากับ 2.778 และมี Application สำหรับการทำนายการเกิดโรค Parkinson

สุดท้ายคือชุดข้อมูล Haberman โดย Samples จะประกอบไปด้วย 306 Samples (Row) และ 3 features (Column) ซึ่ง Label ของแต่ละ Sample จะประกอบไปด้วย Class -1 (Normal) จำนวน 225 Samples และ Class +1 (Died) จำนวน 81 Sample โดยมีค่า Class ratio เท่ากับ 2.778 และมี Application สำหรับการทำนายโอกาสรอดชีวิตหลังรับการผ่าตัดเต้านม

ตารางที่ 4.2 ตารางผลการทดลอง

Dataset	วิธีแก้ Class imbalance	Accuracy	Sensitivity	Specificity	Seed
Parkinson	ไม่มีการแก้ไข	0.8205	0.932	0.4792	789
	Down Sampling	0.6042	0.5208	0.6875	
	Up Sampling	0.7313	0.7279	0.7347	
	Hybrid Sampling	0.7436	0.7216	0.7653	
	ไม่มีการแก้ไข	0.8205	0.932	0.4792	732
	Down Sampling	0.6146	0.5833	0.6458	
	Up Sampling	0.7415	0.7347	0.7483	
	Hybrid Sampling	0.7795	0.7629	0.7959	
	ไม่มีการแก้ไข	0.8205	0.932	0.4792	761
	Down Sampling	0.6875	0.5625	0.8125	
	Up Sampling	0.7381	0.7279	0.7483	
	Hybrid Sampling	0.7744	0.7732	0.7755	
	Mean ไม่แก้ไข	0.8205	0.932	0.4792	-
	Mean Down sampling	0.6354	0.5556	0.7153	
	Mean Up sampling	0.737	0.7302	0.7438	
	Mean Hybrid sampling	0.7658	0.7526	0.7789	
	Std deviation ไมแก้ไข	0	0	0	-
	Std deviation Down sampling	0.0454	0.0318	0.0867	
	Std deviation Up sampling	0.0052	0.0039	0.0079	
	Std deviation Hybrid	0.0194	0.0273	0.0156	
Haberman	ไม่มีการแก้ไข	0.7288	0.9778	0.037	789
	Down Sampling	0.5679	0.5062	0.6296	
	Up Sampling	0.4	0.4756	0.3244	
	Hybrid Sampling	0.6111	0.3922	0.8301	
	ไม่มีการแก้ไข	0.7288	0.9778	0.037	732
	Down Sampling	0.6235	0.3951	0.4198	
	Up Sampling	0.4222	0.4756	0.3689	
	Hybrid Sampling	0.598	0.4248	0.7712	
	ไม่มีการแก้ไข	0.7288	0.9778	0.037	761
	Down Sampling	0.5617	0.4198	0.7037	
	Up Sampling	0.4	0.4978	0.3022	
	Hybrid Sampling	0.634	0.4052	0.8627	
	Mean ไม่แก้ไข	0.7288	0.9778	0.037	-
	Mean Down sampling	0.5844	0.4403	0.7284	
	Mean Up sampling	0.4074	0.483	0.3319	
	Mean Hybrid sampling	0.6144	0.4074	0.8214	
	Std deviation ไม่แก้ไข	0	0	0	-
	Std deviation Down sampling	.0.40	0.0583	0.1132	
	Std deviation Up sampling	0.0128	0.0128	0.0339	
	Std deviation Hybrid	0.0182	0.0164	0.0464	

Dataset	วิธีแก้ Class imbalance	Accuracy	Sensitivity	Specificity	Seed
Colon	ไม่มีการแก้ไข	0.4677	0.5909	0.4	789
	Down Sampling	0.5909	0.4545	0.7273	
	Up Sampling	0.5375	0.525	0.55	
	Hybrid Sampling	0.4355	0.4516	0.4194	
	ไม่มีการแก้ไข	0.4677	0.5909	0.4	732
	Down Sampling	0.4545	0.2273	0.6818	
	Up Sampling	0.5875	0.65	0.525	
	Hybrid Sampling	0.5161	0.5484	0.4839	
	ไม่มีการแก้ไข	0.4677	0.5909	0.4	761
	Down Sampling	0.4773	0.4545	0.5	
	Up Sampling	0.6125	0.625	0.6	
	Hybrid Sampling	0.4839	0.5161	0.4516	
	Mean ไม่แก้ไข	0.4677	0.5909	0.4	-
	Mean Down sampling	0.5076	0.3788	0.6364	
	Mean Up sampling	0.5792	0.6	0.5583	
	Mean Hybrid sampling	0.4785	0.5054	0.4516	
	Std deviation ไมแก้ไข	0	0	0	-
	Std deviation Down sampling	0.0731	0.1312	0.1203	
	Std deviation Up sampling	0.0382	0.0661	0.0382	
	Std deviation Hybrid	0.0406	0.0493	0.0323	

ตารางนี้จะแสดงถึงผลการทดลองของชุดข้อมูล โดยเก็บผลผ่านค่าความถูกต้อง ความอ่อนไหว ความจำเพาะ ทั้งก่อนและหลังการแก้ไขความไม่สมดุลของชุดข้อมูล ซึ่งจะแสดงผลดังตารางด้านล่าง

ที่ชุดข้อมูล Colon เมื่อทำการทดลองทั้งสามครั้ง พบว่าค่าเฉลี่ยความถูกต้องสูงสุดจะอยู่ที่วิธีการ Over Sampling ซึ่งให้ผลลัพธ์อยู่ที่ 0.5792 เช่นเดียวกันกับความอ่อนไหว คือวิธีการ Over Sampling โดยให้ผลลัพธ์ที่ 0.6 และสุดท้ายคือค่าความจำเพาะ ซึ่งอยู่ที่วิธีการ Under Sampling โดยให้ผลลัพธ์อยู่ที่ 0.6364

ต่อมา ชุดข้อมูล Parkinson เมื่อทำการทดลองทั้งสามครั้ง พบว่าค่าเฉลี่ยความถูกต้องสูงสุดจะอยู่ที่วิธีการที่ไม่มีการแก้ไขความไม่สมดุล ซึ่งให้ผลลัพธ์อยู่ที่ 0.8205 เช่นเดียวกันกับความอ่อนไหว คือวิธีการที่ไม่มีการแก้ไขความไม่สมดุล โดยให้ผลลัพธ์ที่ 0.932 และสุดท้ายคือค่าความจำเพาะ ซึ่งอยู่ที่วิธีการ Hybrid Sampling โดยให้ผลลัพธ์อยู่ที่ 0.7789

และสุดท้าย ชุดข้อมูล Haberman เมื่อทำการทดลองทั้งสามครั้ง พบว่าค่าเฉลี่ยความถูกต้องสูงสุดจะอยู่ที่วิธีการที่ไม่มีการแก้ไขความไม่สมดุล ซึ่งให้ผลลัพธ์อยู่ที่ 0.7288 เช่นเดียวกันกับความอ่อนไหว คือวิธีการที่ไม่มีการแก้ไขความไม่สมดุล โดยให้ผลลัพธ์ที่ 0.9778 และสุดท้ายคือค่าความจำเพาะ ซึ่งอยู่ที่วิธีการ Hybrid Sampling โดยให้ผลลัพธ์อยู่ที่ 0.8214

บทที่ 5

สรุป

จากวัตถุประสงค์ข้างต้น คือการศึกษาวิธีการจำแนกประเภทข้อมูล วิธีการจัดการความไม่สมดุลของชุดข้อมูล การพัฒนาโปรแกรมสำหรับจัดการความไม่สมดุลของชุดข้อมูล และเปรียบเทียบผลลัพธ์ของแต่ละวิธีการผ่านการอ้างอิงจากความถูกต้อง ความอ่อนไหว และความจำเพาะ

เพื่อให้บรรลุวัตถุประสงค์เหล่านี้ จึงเกิดการศึกษา และเลือกใช้วิธีการจำแนกประเภทที่สนใจทั้ง 2 ตัว ได้แก่ Support Vector Machine และ Dynamic Multi-hyper plane Partitioning (กำลังศึกษา) และวิธีการแก้ไขความไม่สมดุลของชุดข้อมูล ได้แก่วิธีการ Over Sampling, Under Sampling และวิธีการแบบ Hybrid โดยขั้นตอนแรกคือทำการออกแบบโปรแกรมสำหรับการจำแนกประเภทข้อมูลชนิด Support Vector Machine ต่อมาคือการพัฒนาโปรแกรมสำหรับการแก้ปัญหาความไม่สมดุลของชุดข้อมูล โดยใช้วิธีแก้ปัญหาทั้ง 3 วิธี และทำการเก็บผลลัพธ์โดยใช้เกณฑ์ความถูกต้อง ความอ่อนไหว และความจำเพาะของแต่ละตัวมาทำการเปรียบเทียบกัน ซึ่งเมื่อพิจารณาดูแล้ว พบว่าวิธีการแก้ปัญหาแต่ละชนิดจะมีความเหมาะสมตามชนิดข้อมูลที่ใช้งาน เช่น ถ้าข้อมูลมีจำนวนน้อย วิธีการแบบ Over Sampling จะให้ผลลัพธ์ดีที่สุด แต่ถ้าข้อมูลมีจำนวนมาก วิธีการ Under Sampling หรือวิธีการแบบ Hybrid จะให้ผลลัพธ์ที่ดีกว่าวิธีการแบบ Over Sampling

เอกสารอ้างอิง

[1] Overfitting ปัญหายอดฮิตของ Machine Learning. สืบค้นเมื่อ 23 มีนาคม 2562, จาก https://qtmlresearch.com/2017/03/24/overfitting-%E0%B8%9B%E0%B8%B1%E0%B8%8D%E0%B8%AB%E0%B8%B2%E0%B8%A2%E0%B8%AD%E0%B8%94%E0%B8%AE%E0%B8%B4%E0%B8%95%E0%B8%82%E0%B8%AD%E0%B8%87-machine-learning/?fbclid=IwAR0Z9xa6j_XWiR0XISq3gR_hc5NkMfPEiNpy7q1Zfb9VA0XbUlgeRjnNAHs

[2] Mikel Galar, Alberto Fernandez, Edurne Barrenechea, Humberto Bustince, Member, IEEE, and Francisco Herrera, Member, IEEE. “A Review on Ensembles for the Class Imbalance Problem: Bagging-, Boosting-, and Hybrid-Based Approaches”, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART C: APPLICATIONS AND REVIEWS, VOL. 42, NO. 4, pp.463 - 481, 2012

[3] แนวคิด Support Vector Machine (SVM). สืบค้นเมื่อ 23 มีนาคม 2562, จาก http://www.mindphp.com/forums/viewtopic.php?t=12298

[4] Usine Logicielle.org. (2018). ความหมาย SOFTWARE คืออะไร. สืบค้นเมื่อ 8 เมษายน 2562, จาก https://www.usine-logicielle.org/%E0%B8%84%E0%B8%A7%E0%B8%B2%E0%B8%A1%E0%B8%AB%E0%B8%A1%E0%B8%B2%E0%B8%A2-software-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3/

[5] Tanawat Budkod. (2018). “DATA MINING คืออะไร. สืบค้นเมื่อ 17 เมษายน 2562, จาก http://www.glurgeek.com/education/data-mining-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3-%E0%B8%AD%E0%B8%A2%E0%B8%B2%E0%B8%81%E0%B8%A3%E0%B8%B9%E0%B9%89%E0%B8%95%E0%B9%89%E0%B8%AD%E0%B8%87%E0%B8%AD%E0%B9%88/

[6] TITIPATA. (2018). Bagging หรือ Boosting คืออะไร ทำงานอย่างไร. สืบค้นเมื่อ 17 เมษายน 2562, จาก https://tupleblog.github.io/bagging-boosting/

[7] Azhar Umar Darma. (2018). Is MATLAB software or a tool?. สืบค้นเมื่อ 8 เมษายน 2562, จาก https://www.quora.com/Is-MATLAB-a-software-or-a-tool

[8] วารุณี เทศะกรณ์, วิภา ธนาชาติเวทย์. (2554). วิธีการตรวจวินิจฉัย. ใน ศาสตราจารย์แพทย์หญิงพรรณีปิติสุทธิธรรม, รองศาสตราจารย์ดร.ชยันต์พิเชียรสุนทร, ตำราการวิจัยทางคลินิก Textbook of Clinical Research (441 - 466). กรุงเทพฯ: บริษัทอมรินทร์พริ้นติ้งแอนด์พับลิชชิ่งจำกัด(มหาชน).

[9] Witchapong Daroontham. (2018). รู้จัก Decision Tree, Random Forest, และ XGBoost!!!—PART 1. สืบค้นเมื่อ 17 เมษายน 2562, จาก https://medium.com/@witchapongdaroontham/%E0%B8%A3%E0%B8%B9%E0%B9%89%E0%B8%88%E0%B8%B1%E0%B8%81-decision-tree-random-forrest-%E0%B9%81%E0%B8%A5%E0%B8%B0-xgboost-part-1-cb49c4ac1315

[10] Tjen-Sien Lim, “The UC Irvine Machine Learning Repository,” [Online]. Available: https://archive.ics.uci.edu/ml/machine-learning-databases/haberman. [Accessed Nov. 3, 2019].

[11] J.Geralds, “Sega Ends Production of Dreamcast,” [Online]. Available: http://nli.vnunet.com/news/1116995. [Accessed Nov. 3, 2019].

[12] Max A. Little, “The UC Irvine Machine Learning Repository,” [Online]. Available: https://archive.ics.uci.edu/ml/machine-learning-databases/parkinsons. [Accessed Nov. 3, 2019].

ภาคผนวก

ก. เกณฑ์การเปรียบเทียบ

ตารางที่ ก.1 ความสัมพันธ์ระหว่างค่าจริงกับค่าที่ทำนาย

		ค่าจริง	
		+1	-1
ค่าที่ทำนาย	+1	True Positive (TP)	False Positive (FP)
	-1	False Negative (FN)	True Negative (TN)

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

$$\text{Sensitivity} = \frac{TP}{TP+FN}$$

$$\text{Specificity} = \frac{TN}{TN+FP}$$

เพราะฉะนั้น จะนิยามได้ว่า

ความถูกต้อง (Accuracy) คือ ความสามารถในการทำนายได้ผลลัพธ์ตรงกับค่าจริง

ความอ่อนไหว (Sensitivity) คือ ความสามารถในการทำนายค่า +1 ได้อย่างถูกต้อง

ความจำเพาะ (Specificity) คือ ความสามารถในการทำนายค่า -1 ได้อย่างถูกต้อง

ข. SVM (Support Vector Machines) on Matlab

สามารถใช้ Classification Learner app ที่เป็นเครื่องมือใน Matlab เพื่อสร้าง SVM Model โดยมีคำสั่งเกี่ยวข้องดังนี้

ตารางที่ ข.1 Function : Binary SVM

fitcsvm	ใช้เทรน Support vector machine สำหรับการจำแนกหนึ่งคลาสและแบบสองคลาส
fitSVMPosterior	ทำให้ความน่าจะเป็นหลังที่ได้จากการเทรนมีค่าที่พอดี
predict	ส่งค่าเวกเตอร์ของlabelคลาสที่ทำนายไว้สำหรับข้อมูลตัวทำนายในตาราง
templateSVM	เทมเพลต Support vector machine

ตารางที่ ข.2 Classes : Binary SVM

ClassificationSVM	Support vector machine สำหรับการจำแนกหนึ่งคลาสและแบบสองคลาส
CompactClassificationSVM	Support vector machine สำหรับการจำแนกหนึ่งคลาสและแบบสองคลาสแบบย่อ
ClassificationPartitionedModel	ชุดรูปแบบการจำแนกประเภทที่ผ่านการตรวจสอบความถูกต้องแล้ว