เทคนิคการค้นหาคลาสที่ค้นพบได้ยากสำหรับข้อมูลที่ขนาดแตกต่างกันมาก

นายกิตติพงศ์ ชมบุญ

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์
มหาวิทยาลัยเทคโนโลยีสุรนารี
ปีการศึกษา 2555

RARE CLASS DISCOVERY TECHNIQUES FOR HIGHLY IMBALANCED DATA

Kittipong Chomboon

A Thesis Submitted in Partial Fulfillment of the Requirements for the

Degree of Master of Engineering in Computer Engineering

Suranaree University of Technology

Academic Year 2012

เทคนิคการค้นหาคลาสที่ค้นพบได้ยากสำหรับข้อมูลที่ขนาดแตกต่างกันมาก

มหาวิทยาลัยเทคโนโลยีสุรนารี อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

คณะกรรมการสอบวิทยานิพนธ์

_____

(รศ. ดร.นิตยา เกิดประสพ)
ประธานกรรมการ

_____

(รศ. ดร.กิตติศักดิ์ เกิดประสพ)
กรรมการ (อาจารย์ที่ปรึกษาวิทยานิพนธ์)

_____

(อ. ดร.ธรา อั่งสกุล)
กรรมการ

_____

(ศ. ดร.ชูกิจ ลิมปีจำนงค์)
รองอธิการบดีฝ่ายวิชาการ

_____

(รศ. ร.อ. ดร.กนต์ธร ชำนิประศาสน์)
คณบดีสำนักวิชาวิศวกรรมศาสตร์

กิตติพงศ์ ชมบุญ : เทคนิคการค้นหาคลาสที่ค้นพบได้ยากสำหรับข้อมูลที่ขนาดแตกต่างกันมาก (RARE CLASS DISCOVERY TECHNIQUES FOR HIGHLY IMBALANCED DATA) อาจารย์ที่ปรึกษา : รองศาสตราจารย์ ดร.กิตติศักดิ์ เกิดประสพ, 76 หน้า.

การค้นหาคลาสที่ค้นพบได้ยากเป็นงานหนึ่งของการทำเหมืองข้อมูลที่น่าสนใจ เพราะสามารถช่วยเหลือในการผลิตของภาคอุตสาหกรรมด้วยการค้นหารูปแบบการผลิตที่นำไปสู่เหตุการณ์ที่ผลผลิตเสียหายซึ่งมักจดเป็นเหตุการณ์ที่เกิดขึ้นไม่บ่อยจึงเรียกว่าคลาสที่ค้นพบได้ยาก ประโยชน์ที่ได้คือทำให้ทราบถึงสิ่งที่ควรปรับปรุงหรือแนวทางบำรุงรักษาเครื่องจักร เพื่อให้ได้ผลผลิตสมบูรณ์มากขึ้น แต่ในการค้นหาคลาสที่ค้นพบได้ยากนั้นสามารถทำได้ยาก เนื่องจากคุณสมบัติเด่นของคลาสที่มีอยู่น้อยจะถูกคุณสมบัติเด่นของคลาสส่วนใหญ่บดบัง ซึ่งถ้าหากทำการจำแนกข้อมูลด้วยวิธีการปกติแล้วจะไม่สามารถค้นพบคลาสที่ค้นพบได้ยากเลย ดังนั้นงานวิจัยนี้จึงนำเสนอเทคนิคการค้นหาคลาสที่ค้นพบได้ยาก

ในงานวิจัยนี้ได้นำเสนอแนวทางการแก้ปัญหาในการค้นหาคลาสที่ค้นพบได้ยาก ด้วยการคัดเลือกคอลัมน์ด้วยค่าสหสัมพันธ์ และเพิ่มจำนวนคลาสที่ค้นพบได้ยากด้วยวิธีการสุ่มเกิน เพื่อให้คลาสที่ค้นพบได้ยากมีจำนวนใกล้เคียงกับคลาสส่วนใหญ่ทำให้สามารถสร้างรูปแบบเพื่อการจำแนกคลาสที่ค้นพบได้ยากได้

สาขาวิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อนักศึกษา_____

ปีการศึกษา 2555 ลายมือชื่ออาจารย์ที่ปรึกษา_____

KITTIPONG CHOMBOON : RARE CLASS DISCOVERY TECHNIQUES FOR HIGHLY IMBALANCED DATA. THESIS ADVISOR : ASSOC. PROF. KITTISAK KERDPRASOP, Ph.D., 76 PP.

DATA MINING/CLASSIFICATION/RARE CLASS/OVER-SAMPLING

Rare class discovery is one of interesting data mining tasks that can support manufacturing industries by discovering process patterns that can lead to fault products. Faulty process is rarely occurred; it is thus called rare class. The benefit of rare class discovery is the knowledge regarding to the improvement of process or the tool maintenance policy. Rare class discovery is however a difficult task because its rarely occurrences are always dominant by the much larger group of frequently occurred events.

This research therefore proposes techniques to solve the rare class discovery problem. Our main techniques are feature selection with the correlation analysis and then the increase of minority class, which is rare event, with the over-sampling method. With the proposed techniques, patterns of rare class can finally be discovered.

School of Computer Engineering Student's Signature_____

Academic Year 2012 Advisor's Signature_____

กิตติกรรมประกาศ

วิทยานิพนธ์นี้สำเร็จลุล่วงด้วยดี ผู้วิจัยขอกราบขอบพระคุณ บุคคล และกลุ่มบุคคลต่างๆที่ได้กรุณาให้คำปรึกษา แนะนำ ช่วยเหลืออย่างดียิ่ง ทั้งในด้านวิชาการ และด้านการดำเนินงานวิจัย ดังต่อไปนี้

รองศาสตราจารย์ ดร.นิตยา เกิดประสพ และรองศาสตราจารย์ ดร.กิตติศักดิ์ เกิดประสพ อาจารย์ที่ปรึกษาวิทยานิพนธ์ที่ให้คำปรึกษาในการทำงานวิจัย การจัดการรูปแบบ และช่วยตรวจทานความถูกต้องของวิทยานิพนธ์

ผู้ช่วยศาสตราจารย์ ดร.พิชโยทัย มหัทธนาภิวัฒน์ ผู้ช่วยศาสตราจารย์ ดร.คะชา ชาญศิลป์ ผู้ช่วยศาสตราจารย์ สมพันธ์ ชาญศิลป์ และ ผู้ช่วยศาสตราจารย์ ดร.ปรเมศวร์ ห่อแก้ว อาจารย์ประจำสาขาวิชาวิศวกรรมคอมพิวเตอร์ สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี ที่ให้ความรู้พื้นฐานในสาขาต่าง ๆ ของวิศวกรรมคอมพิวเตอร์

คุณกัลญา พับโพธิ์ เลขานุการสาขาวิชาวิศวกรรมคอมพิวเตอร์ ที่ให้ความช่วยเหลือในการประสานงานด้านเอกสารระหว่างศึกษา คุณไพชยนต์ คงไชย ที่ช่วยตรวจวิทยานิพนธ์ฉบับร่าง

นอกจากนี้ขอขอบคุณครู อาจารย์ทั้งในอดีตและปัจจุบันที่ให้ความรู้แก่ผู้วิจัยจนประสบความสำเร็จในชีวิต

ท้ายที่สุด ขอกราบขอบพระคุณ บิดา มารดา ที่ให้กำเนิด อบรม เลี้ยงดูและส่งเสริมการศึกษาเป็นอย่างดี ทำให้ผู้วิจัยมีความพร้อมด้านความรู้ และมีกำลังใจในการทำวิจัย

กิตติพงศ์ ชมบุญ

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

ตารางที่ **หน้า**

สารบัญตาราง (ต่อ)

ตารางที่ **หน้า**

สารบัญรูป

รูปที่ **หน้า**

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มาของปัญหาการวิจัย

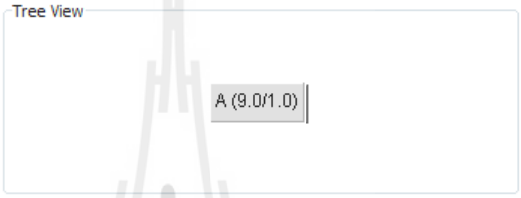
ในปัจจุบันเทคโนโลยีคอมพิวเตอร์มีความก้าวหน้าเป็นอย่างมาก ทำให้วัสดุอุปกรณ์ที่ใช้ในการจัดเก็บข้อมูลมีความทันสมัย การจัดเก็บข้อมูลสามารถทำได้ง่ายและมีประสิทธิภาพสูงมากขึ้น ส่งผลให้ปริมาณของข้อมูลเพิ่มมากขึ้นเช่นเดียวกัน ซึ่งทำให้ยากต่อการวิเคราะห์ข้อมูลเนื่องจากว่าปริมาณของข้อมูลมีการเพิ่มขึ้นตลอดเวลา จึงทำให้การทำเหมืองข้อมูลได้รับความสนใจมากขึ้น เนื่องจากการทำเหมืองข้อมูลนั้นเหมาะสมกับการวิเคราะห์ข้อมูลที่มีขนาดใหญ่

การทำเหมืองข้อมูลคือการค้นหารูปแบบ (Pattern) ของข้อมูลโดยใช้ข้อมูลขนาดใหญ่ในการดำเนินการ ซึ่งรูปแบบของข้อมูลที่ได้นั้นเรียกว่าความรู้ (Knowledge) ความรู้ที่ได้จากการทำเหมืองข้อมูลจะมีประโยชน์ช่วยในการตัดสินใจ หรือใช้ในการพยากรณ์ปรากฏการณ์ในอนาคต ซึ่งเป็นประโยชน์ที่ทุก ๆ องค์กร ไม่ว่าจะเป็นหน่วยงานราชการ หรือว่าหน่วยงานเอกชน จะได้ประโยชน์จากการทำเหมืองข้อมูลเป็นอย่างมาก โดยเฉพาะในด้านธุรกิจสามารถเพิ่มความได้เปรียบในการวางแผนและดำเนินการ

ในปัจจุบันปัญหาที่สำคัญในการทำเหมืองข้อมูล คือการค้นหารูปแบบจากข้อมูลที่มีลักษณะการกระจายของข้อมูลที่ผิดปกติ โดยอาจจะเกิดจากข้อมูลที่มีขนาดของคลาสเป้าหมายที่แตกต่างกันมาก (Chawla, 2005) ตัวอย่างดังรูปที่ 1.1 จะสามารถสังเกตได้ว่าคอลัมน์ที่เป็นเป้าหมายที่ชื่อ “Class” นั้นมีข้อมูลแบ่งออกเป็นคลาส “A” และ “B” แต่ว่ามีจำนวนที่แตกต่างกันโดยที่มีคลาส “A” จำนวน 8 เรคคอร์ด แต่มีคลาส “B” เพียงแค่ 1 เรคคอร์ด ซึ่งเมื่อนำข้อมูลที่คลาสไม่สมดุลนี้มาทำการทำเหมืองข้อมูลด้วยวิธีอุปนัยโครงสร้างต้นไม้แล้วไม่สามารถจำแนกคลาส “B” ได้เลย (ดังรูปที่ 1.2) โมเดลในรูปแบบโครงสร้างต้นไม้นี้จะพยากรณ์ข้อมูลเป็นคลาส A เสมอ งานวิจัยนี้จึงสนใจที่จะทำการศึกษาเทคนิคต่าง ๆ เพื่อให้สามารถจำแนกคลาสที่ค้นพบได้ยาก (คือคลาส “B” ตามตัวอย่างในรูปที่ 1.1) จากข้อมูลที่ขนาดของคลาสแตกต่างกันมากได้

A1	A2	A3	A4	Class
a	b	c	a	A
b	a	c	b	A
a	b	c	a	A
a	a	c	b	A
a	a	c	a	A
a	b	c	b	A
b	a	c	a	A
a	b	c	a	A
a	b	c	a	B

รูปที่ 1.1 แสดงข้อมูลที่มีขนาดของคลาสเป้าหมาย (Class) ที่แตกต่างกันมาก



รูปที่ 1.2 โมเดลลักษณะต้นไม้ตัดสินใจที่สร้างจากข้อมูลตามรูปที่ 1.1

1.2 วัตถุประสงค์ของการวิจัย

ในการดำเนินงานวิจัยนี้มีจุดประสงค์คือ

1. เพื่อศึกษาวิธีการต่าง ๆ ที่จะช่วยให้สามารถค้นหารูปแบบของข้อมูลที่ค้นพบได้ยากจากข้อมูลที่มีการกระจายของข้อมูลแบบเบ้ (Skew) หรือเรียกว่ามีขนาดของข้อมูลแตกต่างกันมาก

2. เพื่อสร้างอัลกอริทึมสำหรับค้นหารูปแบบของข้อมูลที่ค้นพบได้ยากจากข้อมูลที่มีขนาดของคลาสแตกต่างกันมาก

3. เพื่อพัฒนาโปรแกรมที่มีประสิทธิภาพในการค้นหารูปแบบของข้อมูลที่ค้นพบได้ยากจากข้อมูลที่มีขนาดแตกต่างกันมาก

1.3 ขอบเขตของการวิจัย

ในการดำเนินงานได้กำหนดขอบเขตของการวิจัยไว้ดังนี้

1. ศึกษาค้นคว้าและออกแบบวิธีการค้นหารูปแบบข้อมูลที่พบได้ยากอย่างน้อย 2 วิธีการเพื่อใช้ในการเปรียบเทียบประสิทธิภาพการค้นหารูปแบบข้อมูลที่ค้นพบได้ยาก

2. ข้อมูลที่ใช้ในการทดสอบจะเป็นทั้งข้อมูลสังเคราะห์ที่สร้างจากโปรแกรมและข้อมูลจริงจากแหล่งข้อมูลมาตรฐานเช่น UCI Machine Learning Repository โดยข้อมูลเป็นชนิดตัวเลข (Numerical Data)

3. ประเภทของการทำเหมืองข้อมูลที่ใช้ในงานวิจัยนี้จะเป็นการจำแนกข้อมูล (Data Classification)

4. งานวิจัยนี้ใช้ภาษาอาร์ซึ่งเป็นภาษาเชิงฟังก์ชันและเป็นโอเพนซอร์สซอฟต์แวร์ในการเขียนโปรแกรมเพื่อทดสอบประสิทธิภาพของการออกแบบอัลกอริทึม

1.4 ประโยชน์ที่ได้รับ

ผลสำเร็จของการวิจัยนี้ก่อให้เกิดประโยชน์ดังนี้

1. สามารถพัฒนาอัลกอริทึมที่ค้นพบรูปแบบของข้อมูลที่พบได้ยากจากข้อมูลที่มีขนาดของคลาสต่างกันมาก

2. สามารถสร้างโปรแกรมที่มีประสิทธิภาพในการค้นหารูปแบบของข้อมูลที่พบได้ยาก

3. อัลกอริทึมและโปรแกรมที่สร้างขึ้นสามารถใช้เป็นแนวทางในการพัฒนาวิธีการค้นหารูปแบบของข้อมูลที่พบได้ยากในงานประยุกต์เฉพาะด้านได้ต่อไปในอนาคต

บทที่ 2
ปริทัศน์วรรณกรรมและงานวิจัยที่เกี่ยวข้อง

บทนี้กล่าวถึงปริทัศน์วรรณกรรมและงานวิจัยที่เกี่ยวข้องกับงานวิจัยของวิทยานิพนธ์นี้ โดยเนื้อหาประกอบด้วย การทำเหมืองข้อมูล (Data Mining) คลาสที่ค้นพบได้ยาก (Rare Class) การเขียนโปรแกรมด้วยภาษาอาร์ (R Language Programming) และงานวิจัยที่เกี่ยวข้อง

2.1 การทำเหมืองข้อมูล (Data Mining)

การทำเหมืองข้อมูล (J. Han, 2006) คือ การขุดค้นความรู้จากข้อมูลซึ่งสามารถหาได้จากการหาลักษณะพิเศษของข้อมูล หรือรูปแบบของข้อมูล ด้วยวิธีการทางด้านคณิตศาสตร์ หรือทางด้านสถิติ เพื่อนำความรู้จากข้อมูลไปใช้ประโยชน์ในด้านต่าง ๆ ซึ่งการทำเหมืองข้อมูลสามารถทำได้หลายแบบขึ้นอยู่กับวัตถุประสงค์ของการนำความรู้ที่ได้ไปใช้งาน เช่น การหาความสัมพันธ์ของข้อมูล (Association Rule) การจำแนกประเภทข้อมูล (Data Classification) และการแบ่งกลุ่มข้อมูล (Data Clustering) ความรู้ที่ได้หลังจากการทำเหมืองข้อมูลสามารถนำไปใช้ประโยชน์ในการช่วยในการตัดสินใจ หรือใช้ในการพยากรณ์เหตุการณ์ที่จะเกิดขึ้นในอนาคตได้ ดังนั้นการทำเหมืองข้อมูลจึงเป็นเรื่องที่น่าสนใจในปัจจุบัน

2.1.1 วิวัฒนาการของการทำเหมืองข้อมูล

พัฒนาการของการทำงานกับข้อมูลด้วยวิธีการอัตโนมัติในแต่ละช่วงทศวรรษ (Kurt, 2012) สรุปได้ดังนี้

ในช่วงทศวรรษที่ 1960 มีการจัดเก็บข้อมูล (Data Collection) ด้วยอุปกรณ์ที่เหมาะสมเพื่อป้องกันไม่ให้ข้อมูลสูญหาย

ในช่วงทศวรรษที่ 1970 มีการพัฒนารูปแบบการจัดเก็บข้อมูลโดยจัดเก็บข้อมูลในรูปแบบของตารางในฐานข้อมูลเชิงสัมพันธ์ (Relational Database System)

ในช่วงทศวรรษที่ 1980 มีการพัฒนาการเข้าถึงข้อมูล (Data Access) โดยการนำข้อมูลที่จัดเก็บมาสร้างความสัมพันธ์ระหว่างกัน เพื่อนำไปใช้ในการวิเคราะห์และช่วยในการตัดสินใจ

ในช่วงทศวรรษที่ 1990 มีการจัดเก็บข้อมูลเป็นคลังข้อมูลขนาดใหญ่ (Data Warehouse) ซึ่งครอบคลุมการใช้งานทั้งองค์กรเพื่อช่วยในการตัดสินใจ (Decision Support)

ในช่วงทศวรรษที่ 2000 มีการพัฒนาการทำเหมืองข้อมูล (Data Mining) โดยนำข้อมูลจากฐานข้อมูลและคลังข้อมูลมาวิเคราะห์ และสร้างแบบจำลองด้วยเทคนิคด้านสถิติเป็นหลัก

2.1.2 กระบวนการทำเหมืองข้อมูล

การทำเหมืองข้อมูลแบบที่ง่ายที่สุดสามารถแบ่งออกได้เป็น 3 ขั้นตอนดังต่อไปนี้

ขั้นตอนที่ 1 คือขั้นตอนก่อนการทำเหมืองข้อมูล (Pre-processing) ในขั้นตอนนี้จะเป็นขั้นตอนการจัดการเกี่ยวกับข้อมูลก่อนที่จะนำข้อมูลเข้าสู่กระบวนการทำเหมืองข้อมูล โดยจะเป็นการทำความเข้าใจเกี่ยวกับข้อมูล การเลือกเป้าหมาย การเลือกข้อมูลที่ต้องใช้งาน การจัดการข้อมูลเกี่ยวกับข้อมูลที่ผิดเพี้ยน (Noise) และข้อมูลที่ขาดหายไป (Missing Data) เมื่อผ่านกระบวนการเหล่านี้แล้วจึงดำเนินการทำเหมืองข้อมูลต่อไป

ขั้นตอนที่ 2 คือขั้นตอนการทำเหมืองข้อมูล (Mining) โดยนำข้อมูลที่ผ่านกระบวนการจากขั้นตอนที่ 1 มาดำเนินการทำเหมืองข้อมูลด้วยเทคนิคต่าง ๆ เช่น การหาความสัมพันธ์ของข้อมูล การจำแนกข้อมูล และการแบ่งกลุ่มข้อมูล เป็นต้น

ขั้นตอนที่ 3 คือขั้นตอนการตรวจสอบผลลัพธ์ (Result Validation) เป็นขั้นตอนสุดท้ายคือการนำผลลัพธ์ที่ได้จากการทำเหมืองข้อมูลมาทดสอบประสิทธิภาพด้วยข้อมูลทดสอบก่อนที่จะนำผลลัพธ์จากการทำเหมืองข้อมูลไปใช้งาน

2.1.3 ประเภทของการทำเหมืองข้อมูล

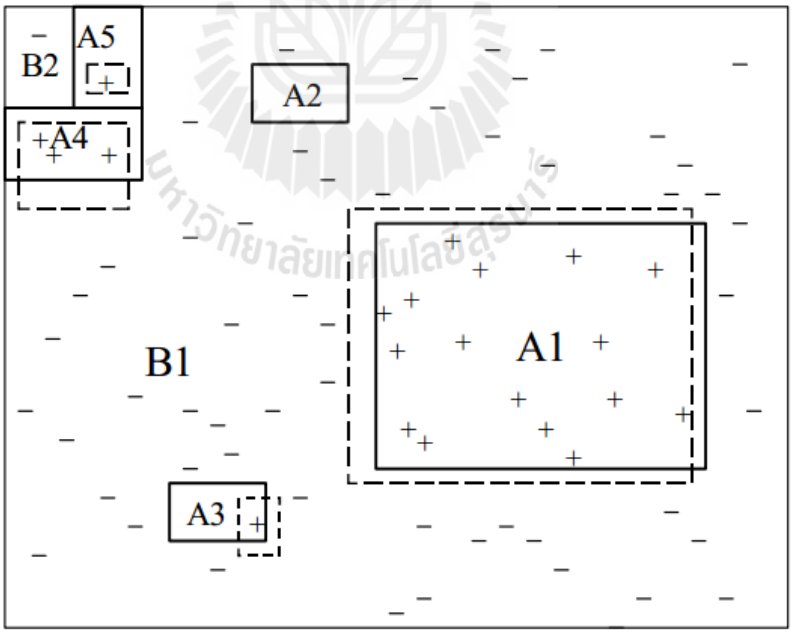
1. การหากฎความสัมพันธ์ของข้อมูล (Association Rule) คือการค้นหากฎความสัมพันธ์ของข้อมูลจากฐานข้อมูลที่มีขนาดใหญ่ เพื่อนำผลลัพธ์ไปใช้ในการวิเคราะห์ความเป็นไปได้ของปรากฏการณ์ต่าง ๆ หรือใช้ในการทำนายเหตุการณ์ที่กำลังจะเกิดขึ้นในอนาคตได้

2. การจำแนกข้อมูล (Data Classification) คือการจำแนกประเภทหรือคลาสของข้อมูลจากข้อมูลขนาดใหญ่ที่มีข้อมูลหลายคลาสปะปนกัน โดยจะทำการจำแนกข้อมูลออกอย่างชัดเจน ด้วยการสร้างรูปแบบของข้อมูล เพื่อใช้ในการจำแนกหรือคาดเดาประเภทของข้อมูลที่จะเกิดขึ้นในอนาคต

3. การแบ่งกลุ่มข้อมูล (Data Clustering) คือการแบ่งกลุ่มข้อมูลจากข้อมูลขนาดใหญ่ให้เป็นกลุ่มข้อมูลย่อย โดยจะพิจารณาจากลักษณะข้อมูลที่ใกล้เคียงกันให้อยู่ในกลุ่มเดียวกัน โดยอาศัยความรู้ด้านสถิติ และคณิตศาสตร์เข้ามาช่วยในการพิจารณาลักษณะของข้อมูล

2.2 คลาสที่ค้นได้พบยาก (Rare Class)

คลาสที่ค้นพบได้ยากหมายถึงกลุ่มของข้อมูลที่ผ่านการทำเหมืองข้อมูลแล้วพบว่าโมเดลหรือรูปแบบข้อมูลไม่สามารถจำแนกประเภทของข้อมูลกลุ่มนี้ได้ เนื่องจากมีจำนวนข้อมูลในคลาสน้อยกว่าข้อมูลในคลาสอื่น ๆ ตัวอย่างเช่น ผลจากการทำเหมืองข้อมูลแล้วพบว่า สามารถทำนายคลาส “A” ได้ถูกต้องถึง 95% แต่ทำนายคลาส “B” ได้ถูกต้องเพียง 5% จะถือว่าคลาส “B” เป็นคลาสของข้อมูลที่ค้นพบได้ยาก โดยปกติแล้วคลาสของข้อมูลที่ค้นพบได้ยากมักจะปรากฏในข้อมูลที่มีขนาดใหญ่มาก ๆ และรูปแบบของข้อมูลไม่สามารถแยกแยะได้อย่างเด่นชัด โดยจะสามารถพิจารณาได้จากการสร้างความสัมพันธ์ หรือการจำแนกข้อมูล โดยจะสังเกตจากจำนวนข้อมูลในคลาสที่มีอยู่น้อยเมื่อเทียบกับจำนวนข้อมูลทั้งหมด ตัวอย่างดังรูปที่ 2.1 ซึ่งเป็นข้อมูลที่มี 2 คลาสคือ + และ - โดย A ซึ่งเป็นกลุ่มข้อมูลที่น้อยกว่ากลุ่ม B โดยข้อมูล A จะแทนด้วยเครื่องหมาย “+” และข้อมูล B ซึ่งเป็นข้อมูลปกติจะแทนด้วยเครื่องหมาย “-” โดยเป้าหมายที่แท้จริงจะถูกล้อมกรอบด้วยเส้นทึบ ส่วนเป้าหมายที่เกิดจากจากการจำแนกข้อมูลจะถูกล้อมรอบด้วยเส้นประ โดยสามารถสังเกตจากรูปที่ 2.1 ได้ว่ากลุ่มข้อมูล A3, A4 และ A5 เป็นกลุ่มข้อมูลที่ค้นพบได้ยากจากการจำแนกข้อมูล ส่วนกลุ่มข้อมูล A2 นั้นไม่สามารถจำแนกได้จากการจำแนกข้อมูล



รูปที่ 2.1 แสดงตัวอย่างข้อมูลที่ค้นพบได้ยากจากการจำแนกข้อมูล (Weiss, 2004)

2.3 การวัดประสิทธิภาพด้วยเมตริกซ์

การประเมินประสิทธิภาพการจำแนกของโมเดลและแสดงผลด้วยเมตริกซ์ เป็นส่วนสำคัญในขั้นตอนสุดท้ายของการทำเหมืองข้อมูลเนื่องจากการวัดประสิทธิภาพของการจำแนกข้อมูล (Classifier) จะบอกถึงความน่าเชื่อถือของโมเดล รูปแบบของเมตริกซ์วัดประสิทธิภาพ (Confusion Matrix) แสดงได้ ดังรูปที่ 2.2

	Actual Positive	Actual Negative
Predict Positive	TP	FP
Predict Negative	FN	TN

รูปที่ 2.2 แสดงเมตริกซ์วัดประสิทธิภาพ (Confusion Matrix)

เมตริกซ์วัดประสิทธิภาพเป็นผลสรุปของการประเมินความสามารถในการจำแนกของโมเดลจำแนกข้อมูลโดยทดสอบด้วยชุดทดสอบ จากรูปที่ 2.2 TP หรือ True Positive คือ จำนวนข้อมูลทดสอบที่เป็นคลาส Positive และโมเดลจำแนกได้ถูกต้องว่าเป็น Positive ค่า FP หรือ False Positive คือจำนวนข้อมูลทดสอบที่ไม่ใช่คลาส Positive แต่โมเดลทำนายผิดว่าเป็นคลาส Positive ค่า TN หรือ True Negative คือ จำนวนข้อมูลทดสอบที่เป็นคลาส Negative และโมเดลทำนายได้ถูกต้องว่าเป็นคลาส Negative ค่า FN หรือ False Negative คือจำนวนข้อมูลทดสอบที่เป็นคลาส Positive แต่โมเดลทำนายผิดว่าเป็นคลาส Negative

ค่า TP, FP, TN และ FN สามารถนำมาใช้คำนวณมาตรวัดต่าง ๆ เพื่อประเมินประสิทธิภาพการจำแนกข้อมูลของโมเดล มาตรวัดที่นิยมใช้โดยทั่วไปประกอบด้วยค่า Accuracy, Precision และ Recall โดยแต่ละมาตรวัดคำนวณได้ดังนี้

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

Precision $= \frac{TP}{TP + FP}$ ซึ่ง Precision จะเป็นการวัดความแม่นยำของการทำนายคลาส Positive คำนวณได้จากการหาอัตราส่วนของการทำนายว่าถูกต้องเมื่อเทียบกับข้อมูลที่ถูกต้องจริง (TP) หารด้วยส่วนที่เป็นจริงทั้งหมดของข้อมูลจริง (TP+FP)

Recall $= \frac{TP}{TP + FN}$ โดย Recall จะเป็นการวัดความสามารถในการค้นหาข้อมูลที่เป็นคลาส Positive คำนวณได้จากการหาอัตราส่วนของการทำนายว่าถูกต้องเมื่อเทียบกับข้อมูลที่ถูกต้องจริง (TP) หารด้วยค่าที่ทำนายว่าถูกต้องทั้งหมด (TP+FN)

Rare Rate คืออัตราการทำนายถูกของคลาสที่ค้นพบได้ยากเมื่อเทียบกับคลาสที่ค้นพบได้ยากทั้งหมด หรืออาจเรียกได้ว่าเป็นค่า Recall ของคลาสที่ค้นพบได้ยาก ยกตัวอย่างเช่น ชุดข้อมูลมีข้อมูลคลาสที่ค้นพบได้ยากทั้งหมด 50 ข้อมูลแต่โมเดลสามารถทำนายได้ถูกต้องจำนวน 30 ข้อมูลค่า Rare Rate สามารถคำนวนได้จาก 30/50 มีค่าเท่ากับ 0.6 หรือ 60%

2.4 การเขียนโปรแกรมด้วยภาษาอาร์ (R Language Programming)

ภาษาอาร์เป็นภาษาการโปรแกรมเชิงฟังก์ชัน (Functional Language) การทำงานจะกระทำผ่านฟังก์ชัน ซึ่งจะทำการคืนค่าให้กับทุกคำสั่งที่สั่งให้ดำเนินการ และเป็นภาษาที่นิยมใช้ในด้านการทำเหมืองข้อมูลเนื่องจากเป็นโปรแกรมที่ถูกพัฒนาเพื่อใช้งานด้านสถิติ ซึ่งถือเป็นพื้นฐานสำคัญของการทำเหมืองข้อมูล นอกจากนั้นยังเป็นโปรแกรมที่ถูกออกแบบมาให้ง่ายต่อการใช้งานเนื่องจากโปรแกรมภาษาอาร์ได้ถูกออกแบบมาเพื่อให้กลุ่มเป้าหมายหลักคือนักสถิติสามารถเขียนโปรแกรมได้ ดังนั้นงานวิจัยนี้จึงเลือกภาษาอาร์ในการเขียนโปรแกรมเพื่อทำเหมืองข้อมูล โดยในงานวิจัยนี้ได้นำโปรแกรมที่มีชื่อว่าอาร์สตูดิโอ (RStudio) มาใช้ในการเขียนโปรแกรมภาษาอาร์สำหรับค้นหาคลาสที่ค้นพบได้ยากในข้อมูลที่มีขนาดแตกต่างกันมาก

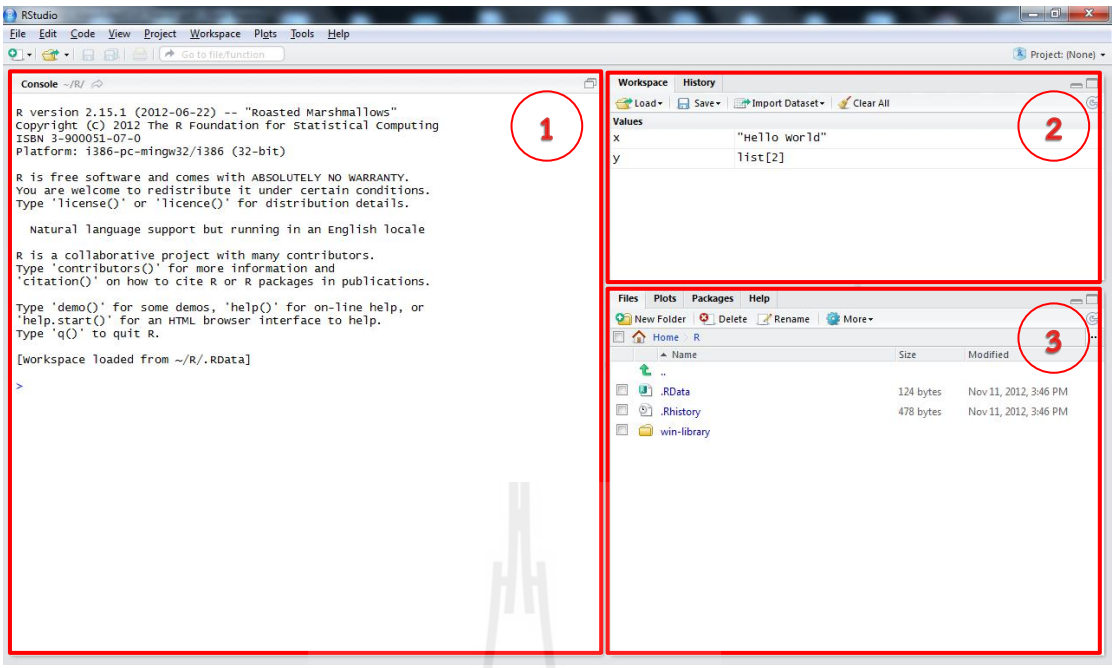
2.4.1 โปรแกรมอาร์สตูดิโอ (RStudio)

โปรแกรมอาร์สตูดิโอเมื่อทำการติดตั้งเรียบร้อยแล้ว และเรียกใช้งานจะปรากฏจอภาพแสดงได้ดังรูปที่ 2.3 โดยจะสังเกตว่าหน้าต่างสำหรับผู้ใช้งานจะถูกแบ่งออกเป็น 3 ส่วนหลัก ๆ ดังนี้

ส่วนที่ 1 เป็นส่วนสำหรับพิมพ์คำสั่งต่าง ๆ (Console) เพื่อเป็นการควบคุมโปรแกรมให้ทำงานตามที่ต้องการโดยผู้ใช้สามารถพิมพ์คำสั่งตามหลักไวยากรณ์ของภาษาอาร์ในหน้าต่างส่วนนี้

ส่วนที่ 2 เป็นส่วนที่แสดงตัวแปรที่ถูกสร้างขึ้นและประวัติการใช้งานคำสั่งของโปรแกรม นอกจากนี้ผู้ใช้ยังสามารถสั่งบันทึกเพื่อเก็บตัวแปรและค่าของตัวแปรสำหรับใช้ภายหลังได้

ส่วนที่ 3 เป็นส่วนที่เกี่ยวกับแฟ้มข้อมูล (File) การดูผลลัพธ์ที่แสดงผลออกเป็นแผนภาพ (Plots) การจัดการเกี่ยวกับแพ็คเกจ (Packages) โดยผู้ใช้สามารถติดตั้งแพ็คเกจ ยกเลิกการติดตั้งแพ็คเกจ หรือเปิดใช้แพ็คเกจได้ที่หน้าต่างส่วนที่ 3 นี้ นอกจากนั้นในส่วนนี้ยังเป็นส่วนที่แสดงผลเกี่ยวกับรายละเอียดและคำอธิบายการใช้งานคำสั่งอีกด้วย (Help)



รูปที่ 2.3 แสดงส่วนประกอบหลักของโปรแกรมอาร์สตูดิโอ

2.4.2 คำสั่งพื้นฐานสำหรับภาษาอาร์

คำสั่งพื้นฐานสำหรับภาษาอาร์ถูกสร้างและรวบรวมไว้แล้วในไลบรารี (Library) โดยผู้ใช้สามารถพิมพ์คำสั่งลงไปในส่วนที่ 1 ของโปรแกรมอาร์สตูดิโอ (Console) โดยในหัวข้อนี้จะนำเสนอคำสั่งที่จำเป็นสำหรับการใช้งานสำหรับการวิจัยนี้เท่านั้น เนื่องจากคำสั่งของภาษาอาร์มีจำนวนมากขึ้นอยู่กับการเลือกใช้งานแพ็คเกจด้วย

คำสั่งเกี่ยวกับแพ็คเกจ

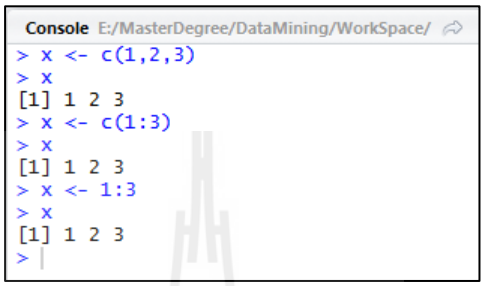
1. คำสั่งติดตั้งแพ็คเกจ ตัวอย่างการใช้คำสั่งเพื่อติดตั้งแพ็คเกจ party สามารถใช้คำสั่งได้ดังนี้ “install.packages(“party”)”

2. คำสั่งยกเลิกการติดตั้งแพ็คเกจ ตัวอย่างการใช้คำสั่งเพื่อยกเลิกการติดตั้งแพ็คเกจ party สามารถใช้คำสั่งได้ดังนี้ “remove.packages(“party”)”

3. คำสั่งเพื่อใช้งานแพ็คเกจที่ได้ทำการติดตั้งเรียบร้อยแล้ว ตัวอย่างการใช้คำสั่งเพื่อใช้งานแพ็คเกจ party สามารถใช้คำสั่งได้ดังนี้ “library(“party”)”

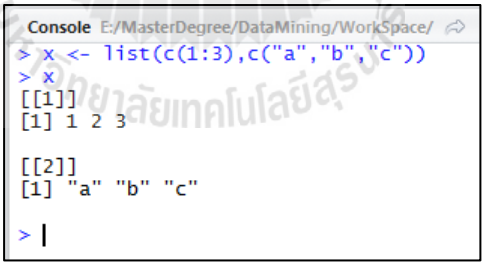
คำสั่งเกี่ยวกับการสร้างข้อมูลชนิดต่าง ๆ

1. คำสั่งสร้างข้อมูลชนิดเวกเตอร์ (Vector) การสร้างข้อมูลชนิดเวกเตอร์นั้นข้อมูลที่อยู่ภายในเวกเตอร์จะต้องเป็นข้อมูลชนิดเดียวกัน เช่น ข้อมูลตัวเลข ข้อมูลตัวอักษร ตัวอย่างการใช้คำสั่งเพื่อสร้างข้อมูลชนิดเวกเตอร์ที่มีสมาชิก 3 ค่าคือ 1, 2, 3 ให้กับตัวแปร x สามารถใช้คำสั่งได้ดังรูปที่ 2.4 โดยมีผลลัพธ์เป็นเช่นเดียวกันทั้ง 3 คำสั่ง



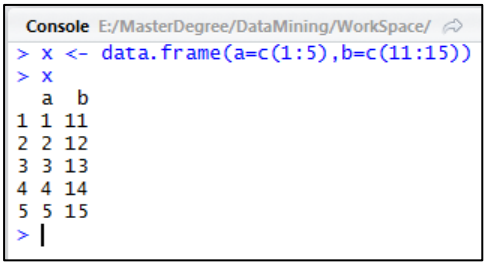
รูปที่ 2.4 แสดงตัวอย่างการสร้างข้อมูลชนิดเวกเตอร์

2. คำสั่งสร้างข้อมูลชนิดลิสต์ (List) โดยข้อมูลในลิสต์ไม่จำเป็นต้องเป็นข้อมูลชนิดเดียวกัน ตัวอย่างการใช้คำสั่งเพื่อสร้างข้อมูลชนิดลิสต์ที่สมาชิกภายในเป็นเวกเตอร์ 1, 2, 3 และเวกเตอร์ “a”, “b”, “c” ให้กับตัวแปร x สามารถใช้คำสั่งได้ดังรูปที่ 2.5



รูปที่ 2.5 แสดงตัวอย่างการสร้างข้อมูลชนิดลิสต์

3. คำสั่งสร้างข้อมูลชนิดดาต้าเฟรม (Data Frame) โดยข้อมูลชนิดดาต้าเฟรมนั้นจะคล้ายข้อมูลชนิดลิสต์เพียงแต่มีข้อกำหนดว่าแต่ละคอลัมน์ต้องมีความยาว (Length) ที่เท่ากัน ตัวอย่างการใช้คำสั่งเพื่อสร้างข้อมูลชนิดดาต้าเฟรมที่ประกอบด้วย 2 คอลัมน์คือ a และ b ให้กับตัวแปร x สามารถใช้คำสั่งได้ดังรูปที่ 2.6

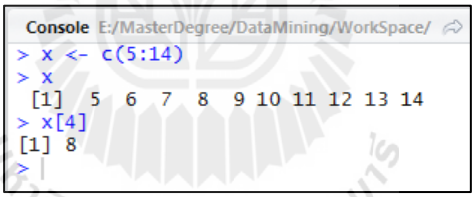


รูปที่ 2.6 แสดงตัวอย่างการสร้างข้อมูลชนิดดาต้าเฟรม

การเข้าถึงข้อมูลด้วยตำแหน่งของข้อมูล

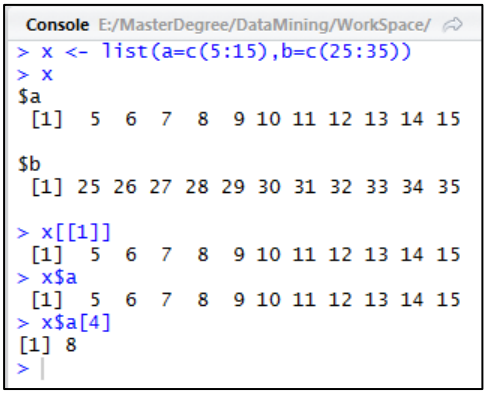
การเข้าถึงข้อมูลทำได้โดยการระบุตำแหน่งของข้อมูลที่ต้องการ โดยข้อมูลทั้ง 3 ชนิดสามารถเข้าถึงข้อมูลด้วยตำแหน่งของข้อมูลดังนี้

1. ข้อมูลชนิดเวกเตอร์ (Vector) สามารถเข้าถึงตำแหน่งข้อมูลด้วยการระบุตำแหน่งที่ต้องการในเครื่องหมาย “[]” ซึ่งเครื่องหมายนี้จะอยู่หลังตัวแปรที่ต้องการระบุตำแหน่งของข้อมูล ตัวอย่างดังรูปที่ 2.7 เป็นการเข้าถึงข้อมูลตัวที่ 4 ในเวกเตอร์ โดยข้อมูลตัวแรกจะนับเป็นตำแหน่งที่หนึ่ง



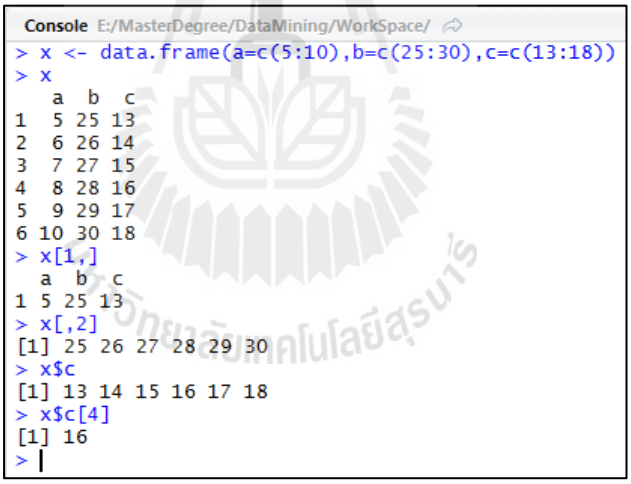
รูปที่ 2.7 แสดงตัวอย่างการเข้าถึงข้อมูลชนิดเวกเตอร์ด้วยตำแหน่งของข้อมูล

2. ข้อมูลชนิดลิสต์ (List) สามารถเข้าถึงตำแหน่งข้อมูลด้วยการระบุตำแหน่งที่ต้องการในเครื่องหมาย “[[]]” ซึ่งเครื่องหมายนี้จะอยู่หลังตัวแปรที่ต้องการระบุตำแหน่งของข้อมูล หรือสามารถใช้เครื่องหมาย “$” แล้วตามด้วยชื่อของตัวแปรในลิสต์ ตัวอย่างดังรูปที่ 2.8



รูปที่ 2.8 แสดงตัวอย่างการเข้าถึงข้อมูลชนิดลิสต์ด้วยตำแหน่งของข้อมูลหรือชื่อตัวแปรในลิสต์

3. ข้อมูลชนิดดาต้าเฟรม (Data Frame) สามารถเข้าถึงข้อมูลด้วยการระบุตำแหน่งที่ต้องการในเครื่องหมาย “[a,b]” โดย a จะแทนแถวที่ต้องการเลือกและ b คือคอลัมน์ที่ต้องการเลือก หรืออาจจะใช้เครื่องหมาย “$” แล้วตามด้วยชื่อคอลัมน์ก็ได้เช่นกัน ตัวอย่างดังรูปที่ 2.9



รูปที่ 2.9 แสดงตัวอย่างการเข้าถึงข้อมูลชนิดดาต้าเฟรมด้วยตำแหน่งของข้อมูลหรือชื่อคอลัมน์

2.5 ข้อมูลที่ผิดปกติ (Outlier)

ข้อมูลที่ผิดปกติในงานวิจัยนี้ได้ใช้การตรวจจับข้อมูลที่ผิดปกติด้วยการอาศัยหลักการทางด้านสถิติด้วยควอร์ไทล์ (Quartile) ซึ่งการใช้เทคนิคนี้จำเป็นที่จะต้องหาค่าควอร์ไทล์ที่ 1 หรือ q1 และค่าควอร์ไทล์ที่ 3 หรือ q3 เพื่อนำมาหาค่าอินเตอร์ควอร์ไทล์ (Interquartile) โดยค่าอินเตอร์ควอร์ไทล์นั้นสามารถหาได้โดยนำ q3 – q1 เมื่อได้ค่าอินเตอร์ควอร์ไทล์เรียบร้อยแล้วนำมาคูณกับ 1.5 กำหนดให้ค่าที่ได้คือ x หลังจากนั้นทำการหาค่าผิดปกติซึ่งจะอยู่นอกช่วงระหว่าง q1-x และ q3+x

ตารางที่ 2.1 แสดงข้อมูลตัวอย่างที่ 1

9	3	1	2	4

การหาค่าควอร์ไทล์ที่ 1 และ 3 จากข้อมูลตารางที่ 2.1 สามารถหาได้ดังนี้

1. ทำการเรียงข้อมูลจากน้อยไปมากแสดงได้ดังตารางที่ 2.2

ตารางที่ 2.2 แสดงข้อมูลตัวอย่างที่ 2 ที่ผ่านการเรียงข้อมูล

1	2	3	4	9

2. สมติให้ค่า n คือจำนวนข้อมูล

ค่า q ของควอร์ไทล์ที่ 1 สามารถหาได้จาก n/4

ค่า q ของควอร์ไทล์ที่ 3 สามารถหาได้จาก n*3/4

3. ถ้าหากค่า q เป็นจำนวนเต็มค่าควอร์ไทล์ที่ได้คือค่าเฉลี่ยของค่าตำแหน่งที่ q และ q+1 แต่ถ้าหากค่า q ไม่เป็นจำนวนเต็มทำการปัดขึ้นและค่าควอไดท์ที่ได้คือตำแหน่งที่ q

จากตารางที่ 2.2 ควอร์ไทล์ที่ 1 มีค่า q เท่ากับ 5/4 = 1.25 ซึ่งไม่เป็นจำนวนเต็มจะทำการปัดขึ้นดังนั้นจะมีค่าเท่ากับ 2 ดังนั้นค่าของควอร์ไทล์ที่ 1 มีค่าเท่ากับข้อมูลตำแหน่งที่ 2 มีค่าเท่ากับ 2 ส่วน ควอร์ไทล์ที่ 3 มีค่า q เท่ากับ 5*3/4 = 3.75 ซึ่งไม่เป็นจำนวนเต็มจะทำการปัดขึ้นดังนั้นจะมีค่าเท่ากับ 4 ดังนั้นค่าของควอร์ไทล์ที่ 3 มีค่าเท่ากับข้อมูลตำแหน่งที่ 4 มีค่าเท่ากับ 4 เมื่อทำการหาค่า q1 และ q3 เรียบร้อยแล้วหาค่า อินเตอร์ควอร์ไทล์ได้จาก q3 – q1 มีค่าเท่ากับ 4-2 = 2 แล้วทำการหาข้อมูลผิดที่ปกติซึ่งจะอยู่นอกช่วง q1-2 ซึ่งก็คือ -1 และ q3+2 ซึ่งก็คือ 6 ดังนั้นค่าที่อยู่นอกช่วง -1 ถึง 6 จะถือเป็นข้อมูลที่ผิดปกติดังนั้นในข้อมูลนี้มีข้อมูลที่ผิดปกติคือ 9

2.6 ค่าสหสัมพันธ์ (Correlation)

ค่าสหสัมพันธ์คือค่าความสัมพันธ์ระหว่างข้อมูล 2 ข้อมูลซึ่งข้อมูล 2 ข้อมูลนี้จำเป็นต้องเป็นข้อมูลชนิดตัวเลขเท่านั้นซึ่งค่าสหสัมพันธ์สามารถเป็นได้ 2 แบบก็คือ แบบบวก (Positive) ซึ่งหมายความว่าเมื่อข้อมูลที่ 1 มีค่าเพิ่มขึ้นข้อมูลที่ 2 ก็จะมีค่าเพิ่มขึ้นด้วย ส่วนแบบที่ 2 คือแบบลบ (Negative) ซึ่งหมายความว่าเมื่อข้อมูลที่ 1 มีค่าเพิ่มขึ้น ข้อมูลที่ 2 จะมีค่าลดลง ตัวอย่างดังรูปที่ 2.10 ที่แสดงค่าความสัมพันธ์ของค่าในแนวแกน x และแกน y ค่าสหสัมพันธ์สามารถหาได้จากสมการที่ 2.1 ซึ่งมีรายละเอียดตัวแปรดังต่อไปนี้

r คือค่าสหสัมพันธ์ที่ได้หาได้จากสมการ

n คือจำนวนข้อมูลทั้งหมด

x_i คือค่าตำแหน่งที่ i ของข้อมูลที่ 1 ซึ่ง i มีค่าตั้งแต่ 1 ถึง n

y_i คือค่าตำแหน่งที่ i ของข้อมูลที่ 2 ซึ่ง i มีค่าตั้งแต่ 1 ถึง n

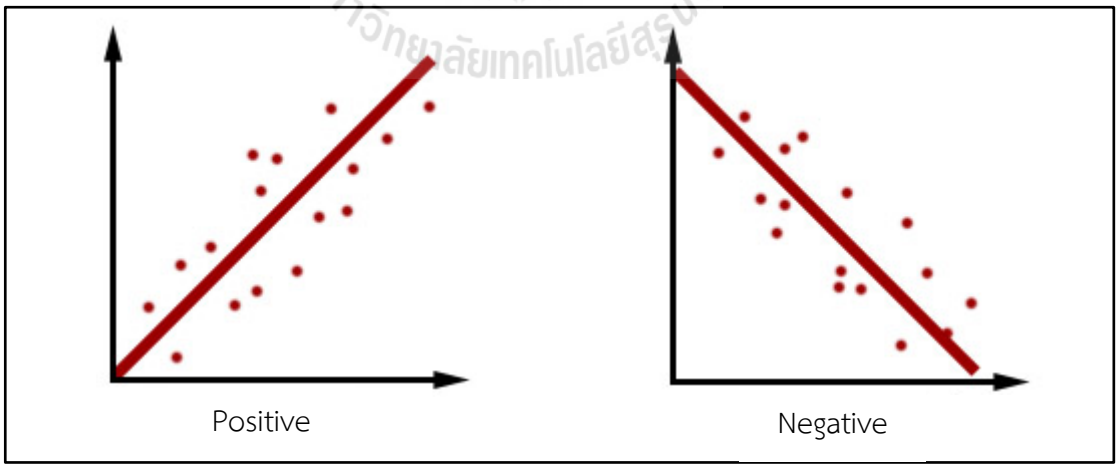
$\overline{x}$ คือค่าเฉลี่ยของข้อมูลที่ 1 ซึ่งมีจำนวนข้อมูล n ตัว

$\overline{y}$ คือค่าเฉลี่ยของข้อมูลที่ 2 ซึ่งมีจำนวนข้อมูล n ตัว

S_x คือค่าเบี่ยงเบนมาตรฐานของข้อมูลที่ 1

S_y คือค่าเบี่ยงเบนมาตรฐานของข้อมูลที่ 2

$$r = \frac{1}{n-1}\sum_{i=1}^{n}\left(\frac{x_i - \overline{x}}{S_x}\right)\left(\frac{y_i - \overline{y}}{S_y}\right) \qquad 2.1$$



รูปที่ 2.10 แสดงรูปแบบของค่าสหสัมพันธ์

จากรูปที่ 2.11 ที่แสดงตัวอย่างการหาค่าความสัมพันธ์ระหว่างข้อมูลคะแนนวิชาคณิตศาสตร์ (x) กับข้อมูลคะแนนวิชาฟิสิกส์ (y) สามารถสรุปได้ว่ามีค่าความสัมพันธ์ แบบบวก ซึ่งสามารถสรุปได้จากค่าสหสัมพันธ์ซึ่งมีค่า 0.851 ซึ่งมีค่าเป็นบวก ถ้าหากค่าสหสัมพันธ์ที่หาได้เป็นลบจะมีความสัมพันธ์แบบลบ

Data		
Name	score Math(x)	score Physic (y)
student A	70	75
student B	73	60
student D	85	86
student E	74	68
student F	50	48
student G	75	80
student H	79	90

	x	y	$\frac{x_i-\bar{x}}{S_x}$	$\frac{y_i-\bar{y}}{S_y}$	$\left(\frac{x_i-\bar{x}}{S_x}\right)\left(\frac{y_i-\bar{y}}{S_y}\right)$	
	70	75	-0.21	0.17	-0.04	
	73	60	0.07	-0.84	-0.05	
	85	86	1.16	0.91	1.06	
	74	68	0.16	-0.30	-0.05	
	50	48	-2.04	-1.64	3.34	
	75	80	0.25	0.51	0.13	
	79	90	0.61	1.18	0.72	
$\bar{x}$	72.29	S_x	10.9501		sum =	5.11
$\bar{y}$	72.43	S_y	14.87568			

$$r = \left(\frac{1}{7-1}\right)(5.11) \quad \therefore r = 0.851$$

รูปที่ 2.11 แสดงตัวอย่างการหาค่าสหสัมพันธ์

2.7 การสุ่มเกิน (Over-Sampling)

ในงานวิจัยนี้นำวิธีการสุ่มเกินมาใช้กับการเพิ่มจำนวนคลาสเพื่อให้มีจำนวนคลาสที่ใกล้เคียงกัน ซึ่งในงานวิจัยนี้ได้ใช้วิธีการสุ่มเกินแบบเท่าตัว (Duplicate) ซึ่งจะเป็นการเพิ่มข้อมูลที่มีจำนวนคลาสน้อยกว่าครึ่งหนึ่งของคลาสที่มีมากที่สุด ตัวอย่างเช่นมีข้อมูลดังตารางที่ 2.3 ที่มีคลาส u จำนวน 5 แถว และมีคลาส s จำนวน 2 แถวจึงต้องทำการเพิ่มจำนวนคลาส s ให้ใกล้เคียงคลาส u โดยการเพิ่มคลาส s จำนวน 4 แถวดังตารางที่ 2.4

ตารางที่ 2.3 แสดงข้อมูลตัวอย่างก่อนการสุ่มเกิน

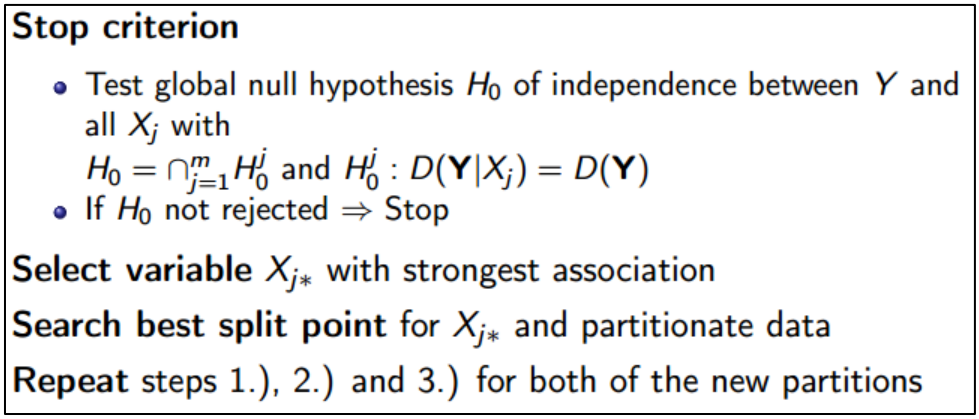
a	b	c	d	Class
1	4	10	35	s
2	5	11	22	u
8	7	12	21	u
1	6	22	23	u
1	4	10	25	u
7	8	12	24	u
1	13	12	13	s

ตารางที่ 2.4 แสดงข้อมูลตัวอย่างหลังการสุ่มเกิน

a	b	c	d	Class
1	4	10	35	s
1	4	10	35	s
2	5	11	22	u
8	7	12	21	u
1	6	22	23	u
1	4	10	25	u
7	8	12	24	u
1	13	12	13	s
1	13	12	13	s

2.8 โครงสร้างต้นไม้แบบมีเงื่อนไข (Conditional Tree)

โครงสร้างต้นไม้แบบมีเงื่อนไขนั้นเป็นการสร้างโมเดลจากข้อมูลที่มีชนิดข้อมูลเป็นตัวเลข (Numerical) โดยการนำเป้าหมายมาทั้งหมดมาสร้างเป็นสมมุติฐาน แล้วทำการทดสอบด้วยการหาค่า p-value โดยที่ถ้าหากสมมุติฐานที่ตั้งสามารถยอมรับได้จะทำการหยุดการทดสอบด้วยค่า p-value หลังจากนั้นนำข้อมูลที่ผ่านการทดสอบด้วยค่า p-value ที่มีค่าน้อยที่สุดมาทำการแบ่งข้อมูลแบบไบนารีทรี ซึ่งสามารถดูได้จากอัลกอริทึมดังรูปที่ 2.12



รูปที่ 2.12 อัลกอริทึมของโครงสร้างต้นไม้แบบมีเงื่อนไข (Molnar, 2012)

ในภาษาอาร์นั้นสามารถเรียกใช้โครงสร้างต้นไม้แบบมีเงื่อนไขได้จากการติดตั้งแพ็คเกจ Party และเลือกใช้ฟังก์ชัน ctree() โดยมีส่วนประกอบของฟังก์ชันดังนี้ ctree(A, B) โดยที่ A จะแทนด้วยรูปแบบที่ระบุคอลัมน์เป้าหมายและคั่นด้วยเครื่องหมาย “~” แล้วจึงตามด้วยคอลัมน์ที่ใช้ในการสร้างโมเดลแต่ถ้าหากต้องการใช้ทุกคอลัมน์ในการสร้างโมเดลสามารถแทนได้ด้วยเครื่องหมาย “.” ตัวอย่างเช่น หากต้องการใช้คอลัมน์ Species และใช้ทุกคอลัมน์ในการสร้างโมเดลสามารถระบุได้ดังนี้ ctree(Species ~. , B) ส่วน B จะเป็นการระบุชุดข้อมูลที่ต้องการตัวอย่างเช่น ถ้าหากต้องการใช้ชุดข้อมูล iris ในการสร้างโมเดลสามารถเรียกคำสั่งได้ดังนี้ ctree(Species ~. , data=iris)

2.9 งานวิจัยที่เกี่ยวข้อง

งานวิจัยที่เกี่ยวข้องกับการค้นหาข้อมูลในคลาสที่ค้นพบได้ยาก เป็นการนำเสนอวิธีการหรือเทคนิคที่นักวิจัยต่าง ๆ ใช้เพื่อค้นหาข้อมูลที่ค้นพบได้ยาก และวิธีที่จะเพิ่มประสิทธิภาพในการค้นพบข้อมูลที่ค้นพบได้ยากให้มีประสิทธิภาพมากขึ้น โดยผู้วิจัยได้ศึกษางานวิจัยที่ได้รับการตีพิมพ์และสามารถสรุปได้ดังนี้

Hamad Alhammady และ Kotagiri Ramamohanarao (2004) ได้ทำการศึกษาเกี่ยวกับการจำแนกข้อมูลที่ค้นพบได้ยาก (Rare Class) โดยใช้โครงสร้างต้นไม้ตัดสินใจ (Decision Tree) และได้นำเสนออัลกอริทึมชื่อ EPDT (Emerging Pattern and Decision Tree) โดยทำการเปรียบเทียบกับอัลกอริทึมอื่นได้แก่ C4.5, PNrule, Metacost, Over-sampling และ EPRC ซึ่งผลจากการทดสอบด้วยข้อมูล 5 ชุดข้อมูลได้แก่ Home insurance, Network intrusion, Disease, Hypothyroid และ Sick-euthyroid โดยใช้มาตรวัด F-measure ซึ่งเป็นการนำค่า Precision และ Recall มาคำนวณร่วมกันด้วย

สูตรการคำนวณ $\frac{2 * Precision * Recall}{Precision + Recall}$ ปรากฏว่าในทุกชุดข้อมูลที่ใช้ทดสอบอัลกอริทึม EPDT ให้ความแม่นยำในการจำแนกมากที่สุด

He He และ Ali Ghodsi (2010) ได้ทำการศึกษาเกี่ยวกับการจำแนกข้อมูลที่ค้นพบได้ยากจากข้อมูลที่มีขนาดต่างกันมาก (Imbalanced Datasets) โดยใช้อัลกอริทึมที่ได้มีการพัฒนามาจาก SVM (Support Vector Machine) ซึ่งได้ทำการทดสอบด้วยชุดข้อมูลทั้งหมด 7 ชุดข้อมูลและใช้มาตรวัด 3 ชนิดได้แก่ F-measure, G-mean และ AUR-ROC โดยทำการเปรียบเทียบอัลกอริทึมที่ทำการพัฒนาเปรียบเทียบกับอัลกอริทึม Under-Sampling ผลการทดสอบปรากฏว่า อัลกอริทึมที่พัฒนามาจาก SVM ให้ผลลัพธ์ที่ดีกว่าอัลกอริทึม SVM ปกติ และโดยรวมแล้วมีประสิทธิภาพดีกว่าอัลกอริทึม Under-Sampling

Junjie Wu, Hui Xiong, Peng Wu และ Jian Chen (2007) ได้ทำการศึกษาเกี่ยวกับการจำแนกข้อมูลที่ค้นพบได้ยากจากข้อมูลที่ขนาดต่างกันมาก ด้วยการใช้เทคนิค COG (Classification using local clustering) ในการช่วยเพิ่มประสิทธิภาพของอัลกอริทึมต่าง ๆ ที่มีลักษณะเป็นอัลกอริทึมที่ใช้ข้อมูลในคอลัมน์อื่นช่วยในการจำแนก (Supervised learning algorithm) เช่น SVM ซึ่งได้ทำการทดลองกับข้อมูลที่เป็นข้อมูลที่ขนาดต่างกันมากจำนวนทั้งหมด 8 ข้อมูลที่ได้นำมาจาก UCI ผลการทดสอบการเพิ่มประสิทธิภาพการจำแนกด้วยเทคนิค COG ปรากฏว่าสามารถเพิ่มประสิทธิภาพได้เป็นอย่างดี

Kate McCarthy, Bibi Zabar และ Gary Weiss (2005) ได้ทำการศึกษาเกี่ยวกับการทดสอบความสามารถในการจำแนกข้อมูลที่ค้นพบได้ยาก ด้วยอัลกอริทึม Cost-Sensitive และ เทคนิคการสุ่มสองเทคนิคได้แก่ Over-Sampling และ Under-Sampling ซึ่งในงานวิจัยของ McCarthy และคณะได้ใช้ข้อมูลที่มีอัตราส่วนของจำนวนข้อมูลที่ค้นพบได้ยากต่อจำนวนข้อมูลปกติน้อยที่สุดอยู่ที่ 4% และมากที่สุดอยู่ที่ 50% เพื่อทำการเปรียบเทียบประสิทธิภาพ ผลจากการทดลองสามารถสรุปได้ว่าผลที่ได้ไม่ชัดเจนพอที่จะสรุปว่าอัลกอริทึมใดมีประสิทธิภาพดีที่สุด แต่พบว่าโดยรวมแล้วอัลกอริทึม Over-Sampling มีประสิทธิภาพดีกว่า Under-Sampling และยังให้คำแนะนำในการวิจัยว่าถ้าหากทำการทดสอบประสิทธิภาพของอัลกอริทึม Over-Sampling และ Under-Sampling ไม่ควรนำไปเปรียบเทียบกับอัลกอริทึม Cost-Sensitive เนื่องจากประสิทธิภาพในการจำแนกข้อมูลใกล้เคียงกัน

จากการศึกษางานวิจัยที่เกี่ยวข้องกับการค้นหาข้อมูลที่ค้นพบได้ยากพบว่า งานวิจัยส่วนใหญ่จะเป็นการนำอัลกอริทึมที่มีอยู่แล้วมาพัฒนาประสิทธิภาพในการจำแนกข้อมูลที่ค้นพบได้ยากและทำการทดสอบอัลกอริทึมด้วยข้อมูลจากแหล่งข้อมูลที่น่าเชื่อถือ แต่งานวิจัยส่วนใหญ่จะเน้น

การวัดประสิทธิภาพด้วยมาตรวัดโดยรวมเช่นค่า Accuracy ค่า F-measure ในงานวิจัยของวิทยานิพนธ์ฉบับนี้เป็นการพัฒนาวิธีการจำแนกข้อมูลที่ค้นพบได้ยากด้วยเทคนิคที่หลากหลายมากขึ้น เช่น การค้นหาข้อมูลผิดปกติ การคัดเลือกคอลัมน์ด้วยค่าสหสัมพันธ์ และการสุ่มเกิน งานวิจัยนี้นอกจากจะประเมินประสิทธิภาพของวิธีการที่พัฒนาขึ้นด้วยมาตรวัด Accuracy แล้วยังเน้นที่การวัดค่า Rare Rate ของคลาสที่ค้นพบได้ยาก โดยงานวิจัยนี้ได้ทำการเปรียบเทียบกับงานวิจัยอื่น ๆ ซึ่งสามารถสรุปได้ดังตารางที่ 2.5

ตารางที่ 2.5 สรุปการเปรียบเทียบงานวิจัยที่เกี่ยวข้องกับเทคนิคการค้นหาคลาสที่ค้นพบได้ยาก สำหรับข้อมูลที่มีขนาดแตกต่างกันมาก

กระบวนการทำงาน	งานวิจัยที่เกี่ยวข้อง*				
	ก	ข	ค	ง	จ
อัลกอริทึมที่ใช้					
EPDT (Emerging Pattern and Decision Tree)	✓				
SVM (Support Vector Machine)		✓	✓		
Decision Tree	✓				✓
Cost-Sensitive				✓	
Over-Sampling				✓	✓
Under-Sampling		✓		✓	
COG (Classification using local clustering)			✓		
มาตรวัดที่ใช้					
F-measure	✓	✓			
G-mean		✓			
AUR-ROC		✓			
Precision & Recall	✓				✓
Confusion Matrix			✓	✓	✓
วัตถุประสงค์การวิจัย					
เพื่อทดสอบประสิทธิภาพ	✓	✓	✓	✓	✓
เพื่อเสนอแนวคิดใหม่	✓	✓	✓	✓	✓
เพื่อทดสอบความถูกต้อง	✓	✓	✓	✓	✓

* ก หมายถึง งานวิจัยของ Hamad Alhammady และ Kotagiri Ramamohanarao (2004)

ข หมายถึง งานวิจัยของ He He และ Ali Ghodsi (2010)

ค หมายถึง งานวิจัยของ Junjie Wu, Hui Xiong, Peng Wu และ Jian Chen (2007)

ง หมายถึง งานวิจัยของ Kate McCarthy, Bibi Zabar และ Gary Weiss (2005)

จ หมายถึง งานวิจัยเรื่อง เทคนิคการค้นหาคลาสที่ค้นพบได้ยากสำหรับข้อมูลที่ขนาดแตกต่างกันมาก (งานวิจัยของวิทยานิพนธ์ฉบับนี้)

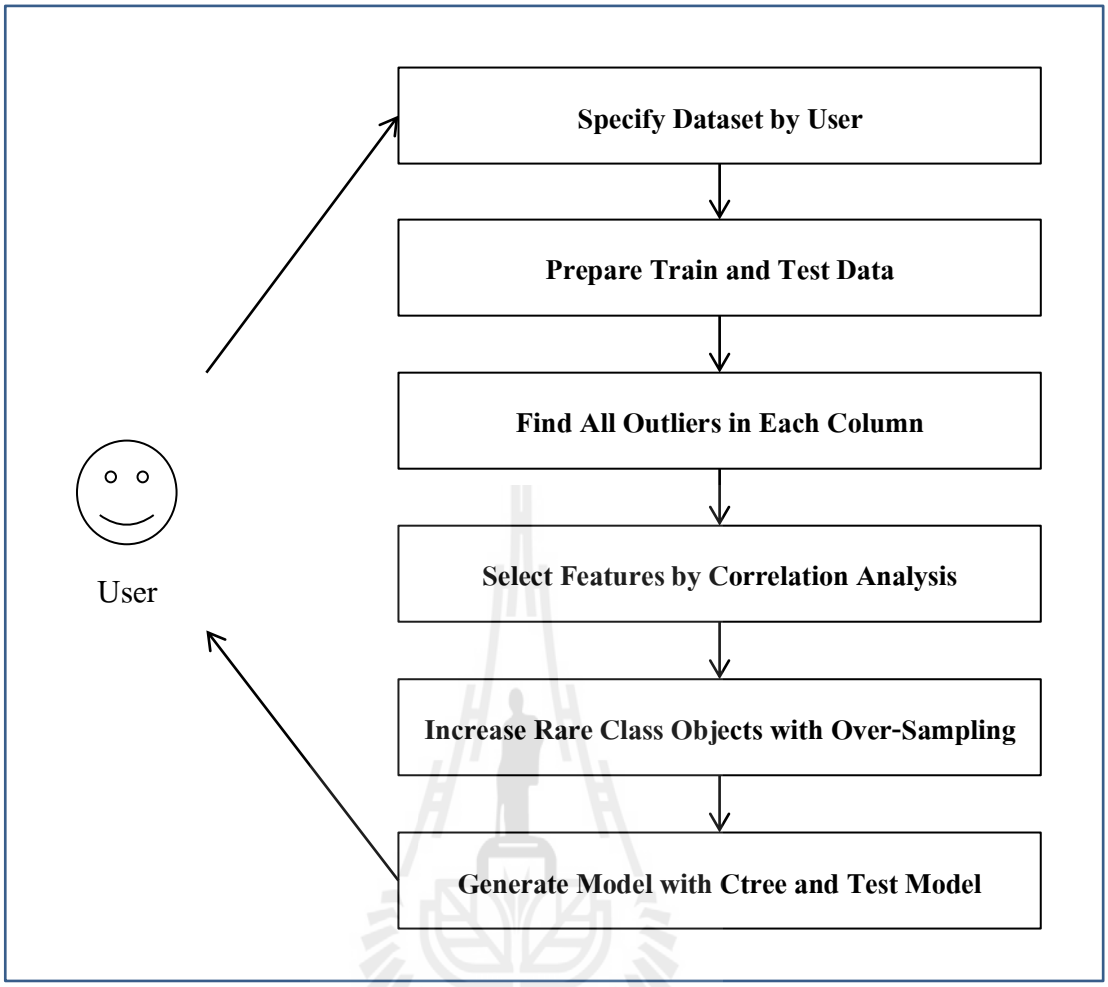
บทที่ 3
วิธีดำเนินการวิจัย

ในส่วนของบทที่ 3 นี้จะกล่าวถึงกรอบแนวคิดของการวิจัย และขั้นตอนการออกแบบอัลกอริทึมที่ใช้ในการค้นหาโมเดลของคลาสที่ค้นพบได้ยาก ด้วยวิธีคัดเลือกคอลัมน์โดยใช้ค่าสหสัมพันธ์ (Correlation) และการเพิ่มจำนวนคลาสด้วยเทคนิคการสุ่มเกิน (Over-Sampling) ซึ่งจะพัฒนาด้วยภาษาอาร์ (R Language) โดยมีรายละเอียดดังนี้

3.1 กรอบแนวคิดของการวิจัย

กรอบแนวคิดหลักของการวิจัยนี้คือ การปรับปรุงกระบวนการทำงานเพื่อค้นหาโมเดลของคลาสที่ค้นพบได้ยาก (Rare Class) จากข้อมูลที่มีขนาดของคลาสที่แตกต่างกันมากหรือเรียกว่าข้อมูลไม่สมดุล (Imbalanced Data) ซึ่งได้ออกแบบแนวทางการวิจัยคือ การค้นหาโมเดลของคลาสที่ค้นพบได้ยากด้วยข้อมูลที่ผิดปกติผนวกกับการเลือกคอลัมน์ด้วยค่าสหสัมพันธ์ กรอบของงานวิจัยจึงเป็นการผสมผสานเทคนิคต่าง ๆ ดังต่อไปนี้ การค้นหาข้อมูลที่ผิดปกติ (Outlier Detection) การเลือกคอลัมน์ด้วยค่าสหสัมพันธ์ (Correlation Selection) และการเพิ่มจำนวนคลาสที่ค้นพบได้ยากด้วยการเพิ่มแบบเท่าตัว โดยงานวิจัยนี้ได้ใช้โครงสร้างต้นไม้ตัดสินใจเป็นเครื่องมือในการจำแนกข้อมูล

รูปที่ 3.1 แสดงกรอบแนวคิดของการวิจัยที่ต้องการค้นหาโมเดลของคลาสที่ค้นพบได้ยากด้วยเทคนิคการวิเคราะห์หาข้อมูลที่ผิดปกติร่วมกับเทคนิคการเลือกคอลัมน์ด้วยการคำนวนค่าสหสัมพันธ์ จากนั้นทำให้ข้อมูลในคลาสที่มีปริมาณน้อยมากหรือเรียกว่าคลาสที่ค้นพบได้ยากปรากฏรูปแบบชัดเจนขึ้นด้วยเทคนิคการสุ่มเกิน



รูปที่ 3.1 กรอบแนวคิดการค้นหาโมเดลของคลาสที่ค้นพบได้ยากด้วยข้อมูลที่ผิดปกติผนวกกับการเลือกคอลัมน์ด้วยค่าสหสัมพันธ์และการสุ่มเกิน

จากกรอบแนวคิดการค้นหาโมเดลของคลาสที่ค้นพบได้ยากจะประกอบด้วย 5 ส่วน ดังต่อไปนี้

1) ระบุชุดข้อมูลโดยผู้ใช้ (Specify Dataset by User) ผู้ใช้มีหน้าที่นำชุดข้อมูลเข้าและทำการกำหนดชื่อให้ชุดข้อมูลที่นำเข้าเพื่อใช้ในการค้นหาโมเดลของคลาสที่ค้นพบได้ยาก และทำการระบุค่าสหสัมพันธ์ที่น้อยที่สุดเพื่อที่จะนำไปใช้ในส่วนของการคัดเลือกคอลัมน์ด้วยค่าสหสัมพันธ์ โดยชุดข้อมูลที่จะนำไปใช้ในการค้นหาโมเดลของคลาสที่ค้นพบได้ยากนั้นจะต้องเป็นข้อมูลประเภทตัวเลข (Numerical Data) เท่านั้น และผู้ใช้ต้องกำหนดค่าสหสัมพันธ์ที่น้อยที่สุดที่เพื่อใช้ในการคัดเลือกคอลัมน์ด้วยค่าสหสัมพันธ์

2) เตรียมข้อมูลฝึกสอนและข้อมูลทดสอบ (Prepare Train and Test Data) เป็นขั้นตอนการเตรียมข้อมูลให้อยู่ในสถานะพร้อมใช้งานคือการกำจัดข้อมูลที่ไม่สมบูรณ์หรือข้อมูลที่ขาดหาย (Missing Data) โดยการทำการแทนค่าที่ไม่สมบูรณ์หรือข้อมูลที่ขาดหายด้วยค่ากึ่งกลาง (Median) แล้วทำการแบ่งข้อมูลที่สมบูรณ์แล้วออกเป็น 2 ส่วนด้วยการสุ่มในอัตราส่วน 70% และ 30% ซึ่งจะใช้ส่วนที่แบ่งออกมา 70% เป็นข้อมูลฝึกสอน (Train Data) ส่วนที่เหลืออีก 30% จะใช้เป็นข้อมูลสำหรับทดสอบ (Test Data)

3) ค้นหาข้อมูลที่ผิดปกติทั้งหมดของทุก ๆ คอลัมน์ (Find All Outliers in Each Column) เป็นขั้นตอนในการหาข้อมูลที่ผิดปกติของทุก ๆ คอลัมน์ของชุดข้อมูลฝึกสอนด้วย Box Plot ผลลัพธ์ที่ได้คือแถวที่มีข้อมูลที่ผิดปกติจากคอลัมน์ทั้งหมด

4) คัดเลือกคอลัมน์ด้วยการวิเคราะห์ค่าสหสัมพันธ์ (Select Features by Correlation Analysis) ขั้นตอนนี้เป็นขั้นตอนการเลือกคอลัมน์ที่มีค่าสหสัมพันธ์ที่มากกว่าค่าสหสัมพันธ์ที่ผู้ใช้กำหนดนำมาจัดเป็นเป็นกลุ่มเดียวกันแล้วทำการคัดเลือกตัวแทนกลุ่มออกมาเพียงคอลัมน์เดียว ส่วนคอลัมน์ที่มีค่าสหสัมพันธ์ระหว่างคอลัมน์น้อยกว่าที่ผู้ใช้กำหนดจะถูกคัดเลือกเพื่อนำไปใช้ในการเพิ่มจำนวนคลาสที่หายากต่อไป

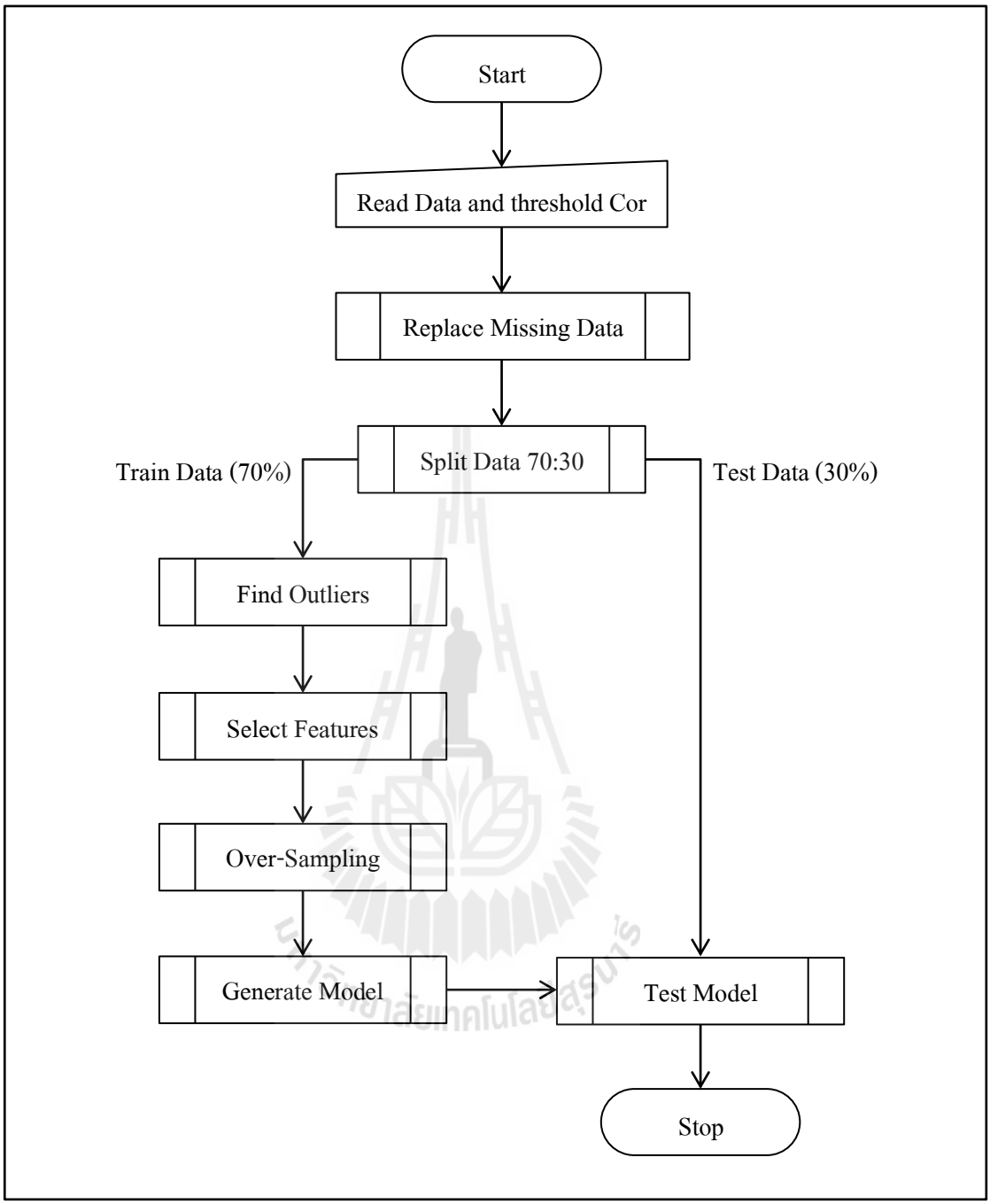
5) เพิ่มจำนวนข้อมูลในคลาสที่ค้นพบได้ยากด้วยการสุ่มเกิน (Increase Rare Class Objects with Over-Sampling) ในขั้นตอนนี้จะนำข้อมูลฝึกสอนที่ผ่านการคัดเลือกจากขั้นตอนก่อนหน้านี้มาทำการเพิ่มจำนวนข้อมูลในทุกคลาสที่มีจำนวนข้อมูลน้อยกว่าคลาสที่เป็นข้อมูลกลุ่มใหญ่ โดยจะทำการเพิ่มครั้งละเท่าตัวซึ่งอาจจะเป็น สองเท่า สามเท่า หรือมากกว่านั้นแต่เมื่อเพิ่มแล้วจะต้องมีจำนวนข้อมูลน้อยกว่าหรือเท่ากับจำนวนข้อมูลในคลาสที่เป็นข้อมูลกลุ่มใหญ่

6) สร้างโมเดลด้วยโครงสร้างต้นไม้และทดสอบโมเดล (Generate Model with Ctree and Test Model) ในขั้นตอนนี้จะนำข้อมูลชุดฝึกสอนที่ได้จากขั้นตอนที่ 5 มาทำการสร้างโมเดลด้วยโครงสร้างต้นไม้ แล้วทำการทดสอบโมเดลที่ได้ด้วยชุดข้อมูลทดสอบ

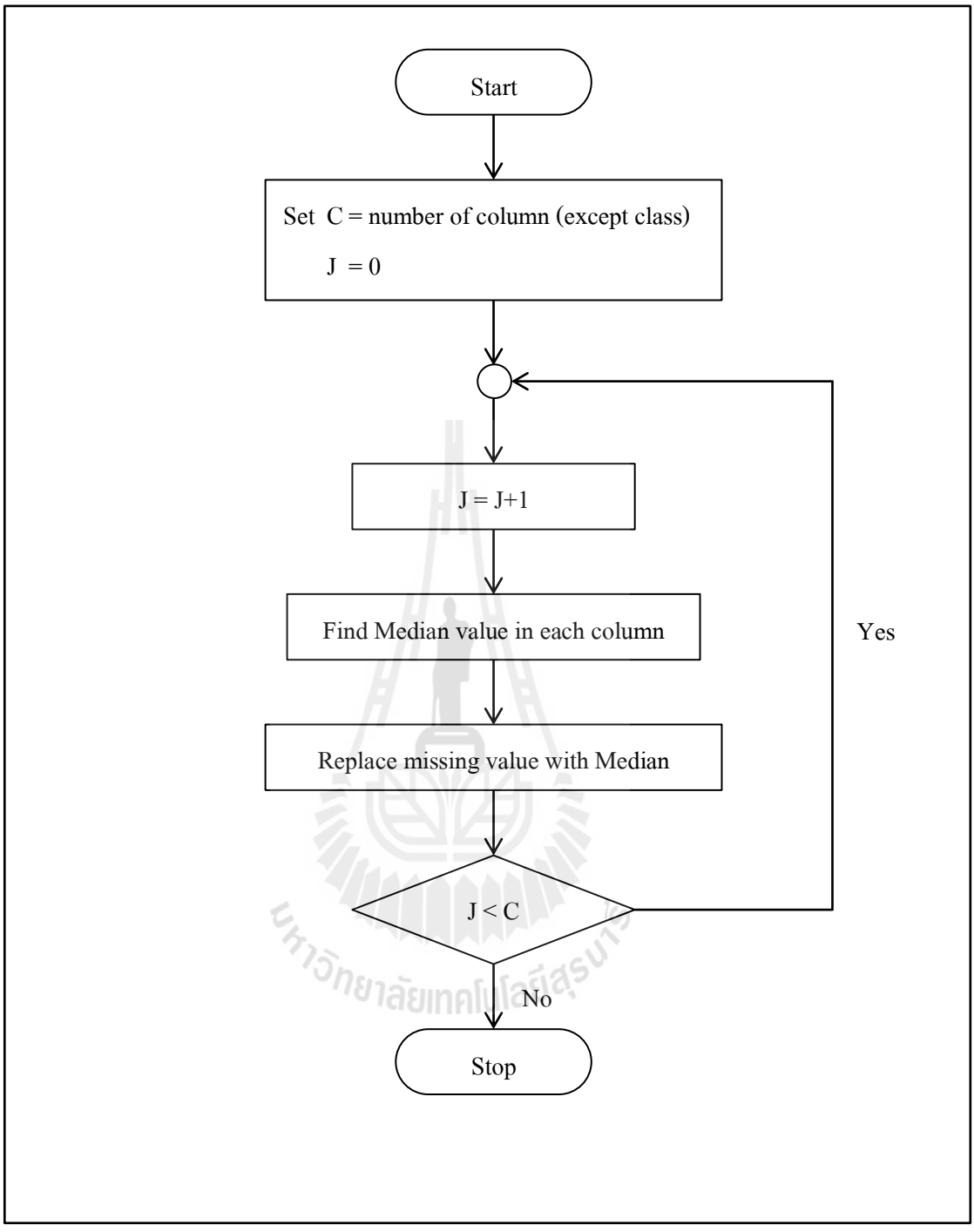
3.2 การออกแบบอัลกอริทึม

ผู้วิจัยได้พัฒนาอัลกอริทึมจากกรอบแนวคิดในหัวข้อที่ 3.1 ให้ชื่อว่า RCD (Rare Class Discovery) ซึ่งในการทดสอบการทำงานของอัลกอริทึมนั้นจะทำการวัดค่าความถูกต้อง (Accuracy) จากโมเดลของการทำนายคลาสที่พบได้ยาก

ในการออกแบบอัลกอริทึม RCD เพื่อใช้ในการค้นหาคลาสที่ค้นพบได้ยากจากข้อมูลที่มีจำนวนข้อมูลในแต่ละคลาสแตกต่างกันมากนั้น จากกรอบแนวคิดของผู้วิจัยจะแสดงเป็นผังงาน (Flow Chart) ได้ดังรูปที่ 3.2



รูปที่ 3.2 Flow chart แสดงขั้นตอนการทำงานของอัลกอริทึม RCD (Rare Class Discovery)



รูปที่ 3.3 Flow Chart แสดงขั้นตอนการทำงานของกระบวนการ Replace Missing Data

```
Process Replace Missing Data

//Input : Data with missing values.
//Output : Filled Data.

(1)     C = count_column(Data)
(2)     for(J=1; J<=C; J++){
(3)             median = find_median_in_column_J(Data)
(4)             for each value v ∈ value in column J {
(5)                     if (v = N/A or NaN or Null){
(6)                             v = median
(7)                     }
(8)             }
(9)     }
(10)    return Data
```

รูปที่ 3.4 กระบวนการ Replace Missing Data

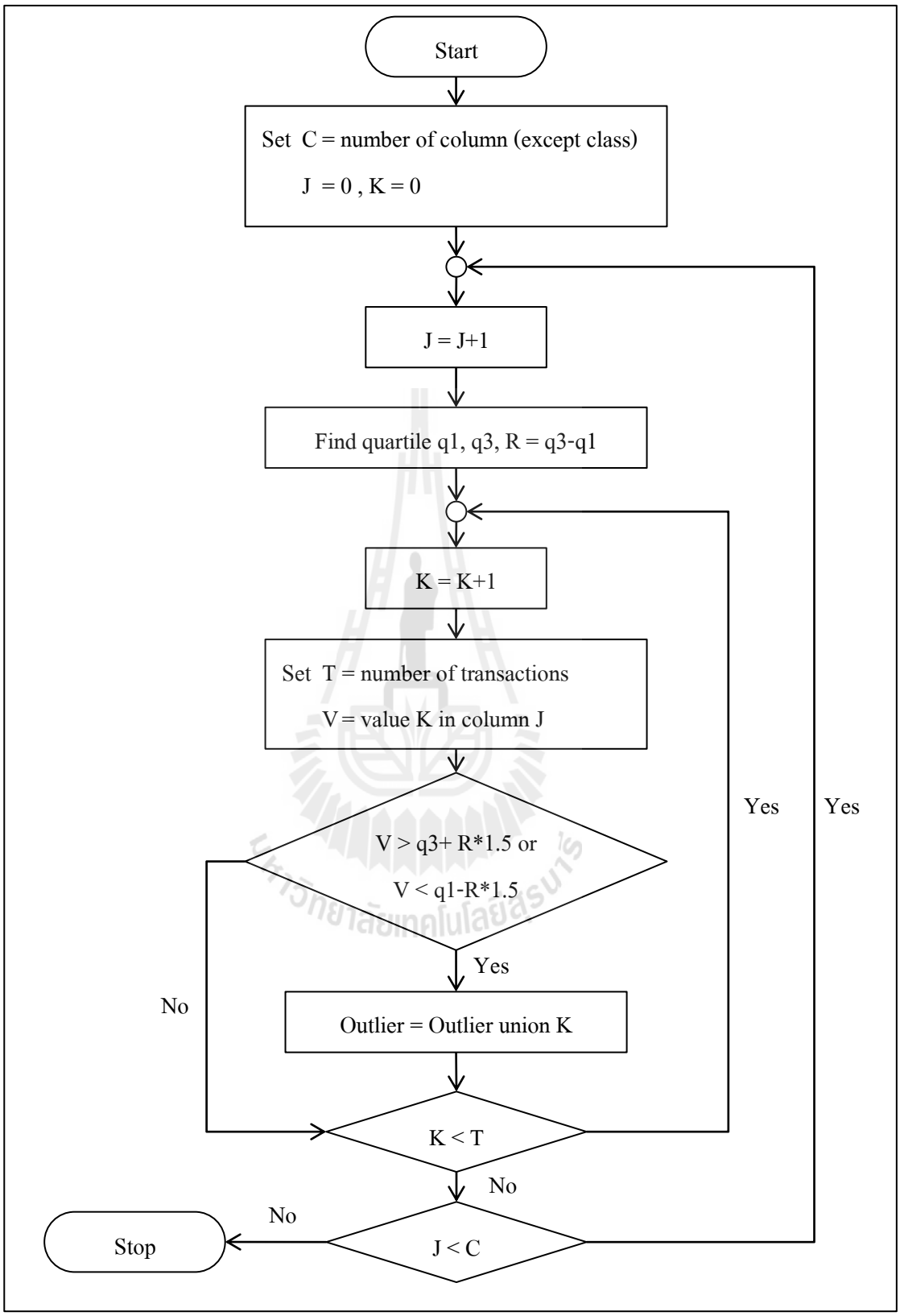
จากรูปที่ 3.3 และรูปที่ 3.4 แสดงขั้นตอนและกระบวนการแทนค่าที่ไม่สมบูรณ์หรือข้อมูลที่ขาดหายด้วยค่ากึ่งกลาง จากขั้นตอนดังกล่าวเมื่อกระทำกับข้อมูลตัวอย่างจากตาราง 3.1 สามารถอธิบายขั้นตอนต่าง ๆ ได้ดังนี้ อันดับแรกทำการนับจำนวนคอลัมน์ทั้งหมดจากชุดข้อมูลโดยไม่นับคอลัมน์คลาส ในที่นี้นับได้ทั้งหมด 4 คอลัมน์ หลังจากนั้นทำการหาค่ากึ่งกลาง (Median) ทีละคอลัมน์เริ่มจากคอลัมน์แรกคือคอลัมน์ a ซึ่งจากข้อมูลตัวอย่างคอลัมน์ a มีค่ากึ่งกลางคือ 1 แล้วทำการค้นหาค่าที่ไม่สมบูรณ์หรือข้อมูลที่ขาดหายแล้วทำการแทนที่ด้วย 1 เมื่อทำครบทุกคอลัมน์แล้วจะได้ข้อมูลที่สมบูรณ์ดังตารางที่ 3.2

ตารางที่ 3.1 แสดงข้อมูลตัวอย่างเพื่อใช้ประกอบการอธิบายขั้นตอนและกระบวนการต่าง ๆ ของอัลกอริทึม RCD

Row	a	b	c	d	Class
1	1	4	10	35	s
2	2	5	11	22	u
3	8	7	12	21	u
4	1	6	22	23	u
5	1	4	10	25	u
6	7	8		24	u
7	1	5	12	23	u
8	0	6	24	24	u
9		8	13	26	u
10	1	13	12	13	s

ตารางที่ 3.2 แสดงข้อมูลตัวอย่างที่ผ่านกระบวนการแทนค่าที่ไม่สมบูรณ์หรือข้อมูลที่ขาดหายด้วยค่ากึ่งกลาง

Row	a	b	c	d	Class
1	1	4	10	35	s
2	2	5	11	22	u
3	8	7	12	21	u
4	1	6	22	23	u
5	1	4	10	25	u
6	7	8	12	24	u
7	1	5	12	23	u
8	0	6	24	24	u
9	1	8	13	26	u
10	1	13	12	13	s



รูปที่ 3.5 Flow Chart แสดงขั้นตอนการทำงานของกระบวนการ Find Outliers

```
Process Find Outliers

//Input : Train Data.
//Output : Outlier.

(1)     C = count_column(Data)
(2)     for(J=1; J<=C; J++){
(3)             q1 = fine_quartile1_in_column_J(Train Data)
(4)             q3 = fine_quartile3_in_column_J(Train Data)
(5)             R = q3-q1
(6)             for each value v ∈ value in column J {
(7)                     if (v < q1-R*1.5 or v > q3+R*1.5){
(8)                             Outlier = Outlier union instance(v)
(9)                     }
(10)            }
(11)    }
(12)    return Outlier
```

รูปที่ 3.6 กระบวนการ Find Outliers

จากรูปที่ 3.5 และรูปที่ 3.6 แสดงขั้นตอนและกระบวนการค้นหาข้อมูลที่ผิดปกติทั้งหมดของทุก ๆ คอลัมน์สามารถอธิบายขั้นตอนต่าง ๆ ได้ดังต่อไปนี้ ขั้นตอนแรกทำการนับจำนวนคอลัมน์ทั้งหมดจากชุดข้อมูลยกเว้นคอลัมน์คลาส ในที่นี้ใช้ชุดข้อมูลตัวอย่างจากตารางที่ 3.2 จะได้จำนวนคอลัมน์ 4 คอลัมน์ หลังจากนั้นทำการค้นหาข้อมูลที่ผิดปกติทั้งหมดของทุกคอลัมน์ด้วยเทคนิค การใช้ค่าควอร์ไทล์ในการค้นหาข้อมูลที่ผิดปกติ โดยการหาข้อมูลที่อยู่นอกช่วงของอินเตอร์ควอร์ไทล์ 1.5 เท่า ตัวอย่างเช่นคอลัมน์ a จากตาราง 3.2 สามารถทำการหาค่าช่วงของอินเตอร์ควอร์ไทล์ได้ดังนี้ อันดับแรกทำการเรียงข้อมูลดังตารางที่ 3.3 และทำการหาค่าควอร์ไทล์ที่ 1 จากสูตรคือจำนวนข้อมูลทั้งหมดหารด้วย 4 คือ 10/4 เท่ากับ 2.5 ถ้าเป็นเลขที่ไม่ลงตัวให้ทำการปัดขึ้นให้เป็นจำนวนเต็มซึ่งมีค่าเท่ากับ 3 ดังนั้นค่าควอร์ไทล์ที่ 1 มีค่าเท่ากับค่าในตารางที่ 3.3 ตำแหน่งที่ 3 ซึ่งมีค่าเท่ากับ 1 และการหาค่าควอร์ไทล์ที่ 3 จากสูตรคือจำนวนข้อมูลทั้งหมดคูณด้วย 3/4 คือ (10*3)/4 เท่ากับ 7.5 ถ้าเป็นเลขที่ไม่ลงตัวให้ทำการปัดขึ้นให้เป็นจำนวนเต็มเช่นกันซึ่งมีค่าเท่ากับ 8 ดังนั้น

ค่าควอร์ไทล์ที่ 3 มีค่าเท่ากับค่าในตารางที่ 3.3 ตำแหน่งที่ 8 มีค่าเท่ากับ 2 ซึ่งการหาค่าอินเตอร์ควอร์ไทล์คือการนำค่าควอร์ไทล์ที่ 3 ลบกับค่าควอร์ไทล์ที่ 1 ซึ่งมีค่าเท่ากับ 2-1 เท่ากับ 1 ส่วนการหาข้อมูลที่ผิดปกติคือนำค่าอินเตอร์ควอร์ไทล์คูณด้วย 1.5 คือ 1*1.5 ซึ่งมีค่าเท่ากับ 1.5 แล้วนำไปคำนวนหาช่วงเพื่อระบุข้อมูลที่ผิดปกติคือนำค่าที่ได้ไปลบกับค่าควอร์ไทล์ที่ 1 คือ 1-1.5 มีค่าเท่ากับ -0.5 และนำค่าที่ได้ไปบวกกับค่าควอร์ไทล์ที่ 3 คือ 2+1.5 มีค่าเท่ากับ 3.5 ดังนั้นข้อมูลที่ไม่อยู่ในช่วง -0.5 ถึง 3.5 จะถือว่าเป็นข้อมูลที่ผิดปกติ ดังนั้นข้อมูลที่ผิดปกติในคอลัมน์ a มี 2 ค่าคือ 7 และ 8 ดังตารางที่ 3.4 แล้วทำการหาข้อมูลที่ผิดปกติทุกคอลัมน์จะได้ดังตารางที่ 3.5 เมื่อทำการค้นหาข้อมูลที่ผิดปกติครบทุกคอลัมน์แล้วทำการคัดเลือกเฉพาะแถวที่มีข้อมูลที่ผิดปกติมาสร้างเป็นชุดข้อมูลใหม่ดังตารางที่ 3.6

ตารางที่ 3.3 แสดงค่าควอร์ไทล์ที่ 1 และค่าควอร์ไทล์ที่ 3 ของคอลัมน์ a

a									
0	1	1	1	1	1	1	2	7	8

ตารางที่ 3.4 แสดงข้อมูลที่ผิดปกติของคอลัมน์ a

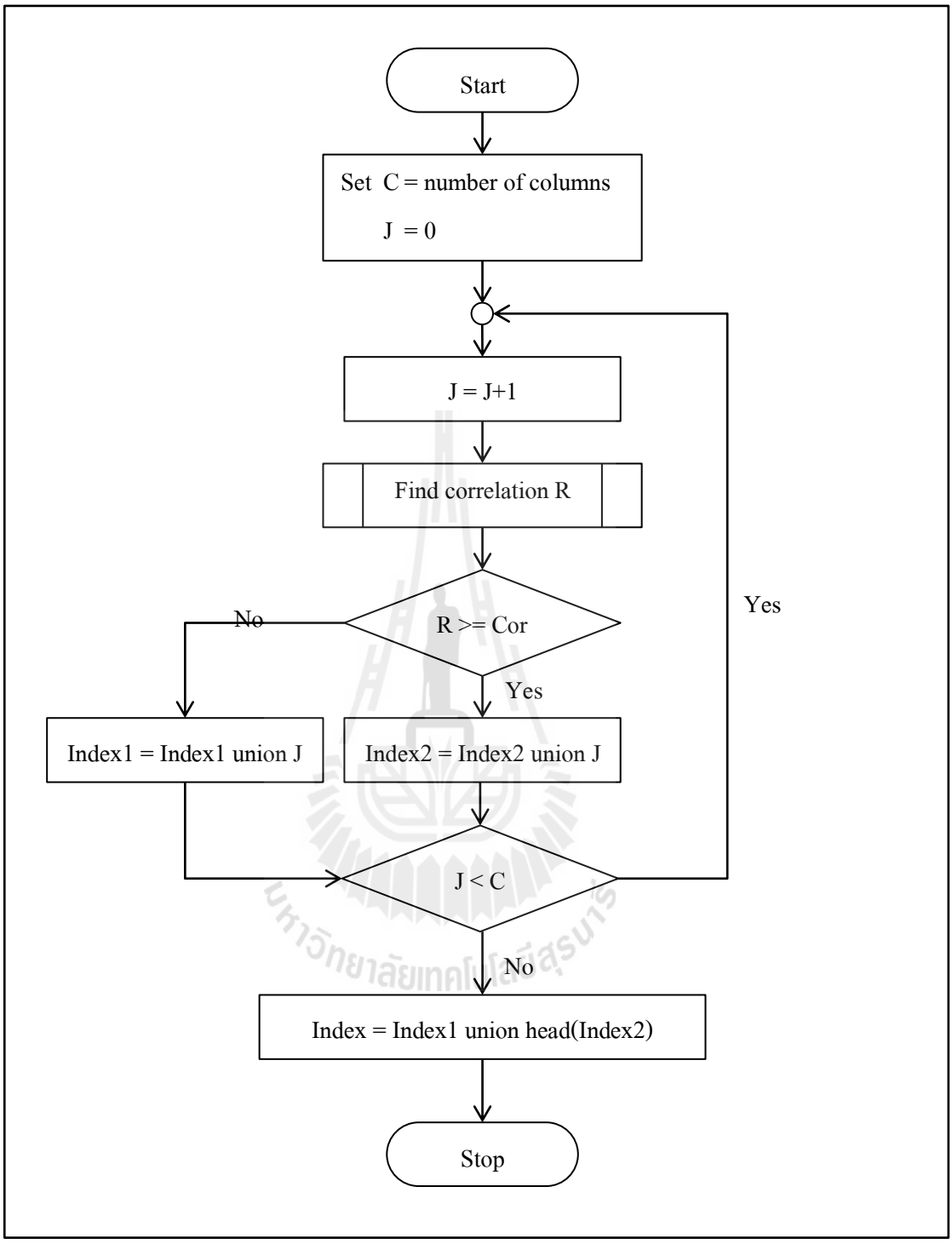
a									
0	1	1	1	1	1	1	2	7	8

ตารางที่ 3.5 แสดงข้อมูลที่ผิดปกติของทุกคอลัมน์

Row	a	b	c	d	Class
1	1	4	10	35	s
2	2	5	11	22	u
3	8	7	12	21	u
4	1	6	22	23	u
5	1	4	10	25	u
6	7	8	12	24	u
7	1	5	12	23	u
8	0	6	24	24	u
9	1	8	13	26	u
10	1	13	12	13	s

ตารางที่ 3.6 แสดงชุดข้อมูลใหม่ที่ได้จากข้อมูลที่ผิดปกติ

Row	a	b	c	d	Class
1	1	4	10	35	s
3	8	7	12	21	u
4	1	6	22	23	u
6	7	8	12	24	u
8	0	6	24	24	u
10	1	13	12	13	s



รูปที่ 3.7 Flow Chart แสดงขั้นตอนการทำงานของกระบวนการ Select Features

```
Process Select Features

//Input : Train Data, Cor.
//Output : Train Data.

(1)     C = count_column(Train Data)
(2)     for(J=1; J<=C; J++){
(3)             R = find_correlation_in_column_J(Train Data)
(5)             if (R >= Cor){
(6)                     Index2 = Index2 union J
(7)             }else{
(8)                     Index1 = Index1 union J
(9)             }
(10)    }
(11)    Index = Index1 union head(Index2)
(12)    Data = select_column_Index(Train Data)
(13)    return Train Data
```

รูปที่ 3.8 กระบวนการ Select Features ด้วยค่าสหสัมพันธ์

จากรูปที่ 3.7 และรูปที่ 3.8 แสดงขั้นตอนและกระบวนการเลือกคอลัมน์ด้วยค่าสหสัมพันธ์ สามารถอธิบายขั้นตอนต่าง ๆ ได้ดังต่อไปนี้ ในขั้นตอนนี้จะทำการหาค่าสหสัมพันธ์ระหว่างคอลัมน์ทุกคอลัมน์ แล้วนำเกณฑ์ค่าสหสัมพันธ์ขั้นต่ำที่ได้จากผู้ใช้ไปทำการเปรียบเทียบเพื่อแบ่งกลุ่ม โดยถ้าหากค่าสหสัมพันธ์มากกว่าค่าที่ผู้ใช้กำหนดจะรวมเป็นกลุ่มเดียวกัน แล้วจึงเลือกตัวแทนกลุ่มออกมา 1 คอลัมน์ แต่ถ้าหากค่าสหสัมพันธ์น้อยกว่าค่าที่ผู้ใช้กำหนดจะนำมาทุกคอลัมน์ ดังตารางที่ 3.7 และ 3.8 โดยในที่นี้สมมุติให้ค่าสหสัมพันธ์ที่ผู้ใช้กำหนดคือ 0.70 (ค่าสหสัมพันธ์ 0.70 เป็นค่าที่สมมุติเพื่อให้สามารถเข้าใจกระบวนการทำงานเท่านั้น) และใช้ข้อมูลจากตาราง 3.2 ในการหาค่าสหสัมพันธ์ และใช้ข้อมูลจากตาราง 3.6 ในการสร้างชุดข้อมูลใหม่

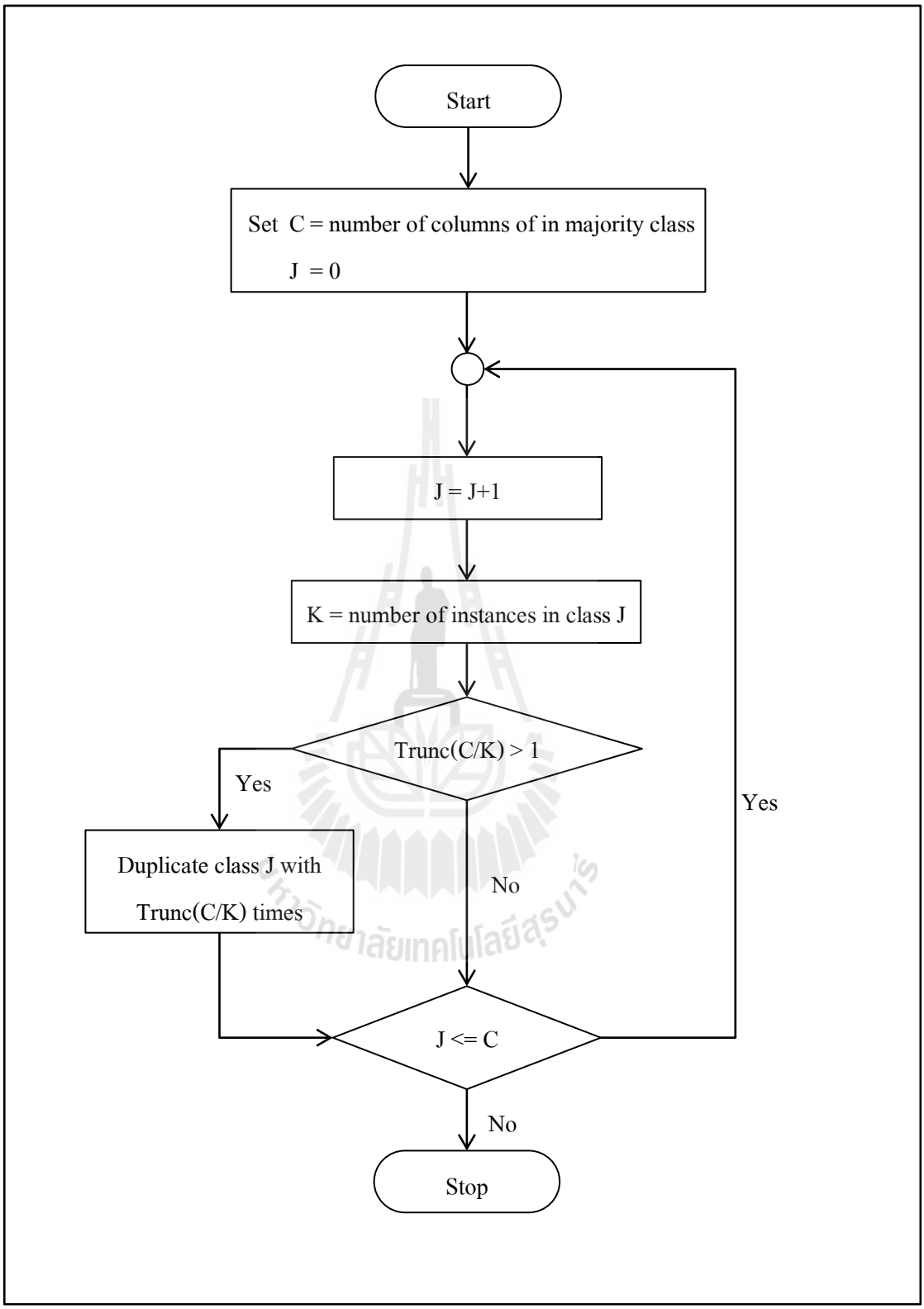
ตารางที่ 3.7 แสดงค่าสหสัมพันธ์ระหว่างคอลัมน์จากชุดข้อมูลในตาราง 3.2

Correlation	a	b	c	d
A	1	0.13	0.35	0.13
B	0.15	1	0.55	0.73
C	0.35	0.55	1	0.53
D	0.13	0.73	0.53	1

ตารางที่ 3.8 แสดงชุดข้อมูลใหม่ที่ได้จากข้อมูลที่ผิดปกติและการคัดเลือกคอลัมน์ด้วยค่าสหสัมพันธ์

Row	a	b	c	Class
1	1	4	10	s
3	8	7	12	u
4	1	6	22	u
6	7	8	12	u
8	0	6	24	u
10	1	13	12	s

หลังจากทำการคัดเลือกคอลัมน์เรียบร้อยแล้ว ขั้นตอนต่อไปเป็นการสุ่มเกินซึ่งขั้นตอนและกระบวนการต่าง ๆ สามารถอธิบายได้ด้วย Flow Chart และอัลกอริทึม ในรูปที่ 3.9 และ 3.10



รูปที่ 3.9 Flow Chart แสดงขั้นตอนการทำงานของกระบวนการ Over-Sampling

```
Process Over-Sampling

//Input : Train Data
//Output : Over-Sampling class Train Data.

    (1)     C = max_class(Train Data)
    (2)     for(J=1; J<=C; J++){
    (3)             K = count_class_J(Train Data)
    (5)             if (trunk(C/K) > 1){
    (6)                     next
    (7)             }else{
    (8)                     duplicate_class_J(TrunC(C/K))
    (9)             }
    (10)    }
    (11)    return Train Data
```

รูปที่ 3.10 กระบวนการ Over-Sampling

จากรูปที่ 3.9 และรูปที่ 3.10 เป็นขั้นตอนและกระบวนการในการสุ่มเกินของข้อมูลในคลาสที่มีจำนวนน้อยกว่าข้อมูลกลุ่มใหญ่ สามารถอธิบายขั้นตอนต่าง ๆ พร้อมตัวอย่างได้ดังนี้ ขั้นแรกจะทำการค้นหาคลาสที่มีจำนวนมากที่สุด และทำการเพิ่มคลาสที่มีจำนวนน้อยกว่าหรือเท่ากับครึ่งหนึ่งของคลาสที่มีจำนวนมากที่สุดด้วยการเพิ่มทีละเท่าตัว ตัวอย่างเช่นข้อมูลจากตารางที่ 3.8 มีคลาสที่มีจำนวนมากที่สุดคือคาส u จำนวน 4 แถว และมีจำนวนคลาส s จำนวน 2 แถว จะเห็นได้ว่าคลาส s น้อยกว่าหรือเท่ากับครึ่งหนึ่งของคลาส u และทำการเพิ่มคลาส s เพิ่มขึ้นเป็น 2 เท่าได้ดังตาราง 3.9

ตารางที่ 3.9 แสดงชุดข้อมูลใหม่ที่ได้จากกระบวนการสุ่มเกินของคลาส

Row	a	b	c	Class
1	1	4	10	s
1(new)	1	4	10	s
3	8	7	12	u
4	1	6	22	u
6	7	8	12	u
8	0	6	24	u
10	1	13	12	s
10(new)	1	13	24	s

3.3 การพัฒนาและการใช้งานโปรแกรม

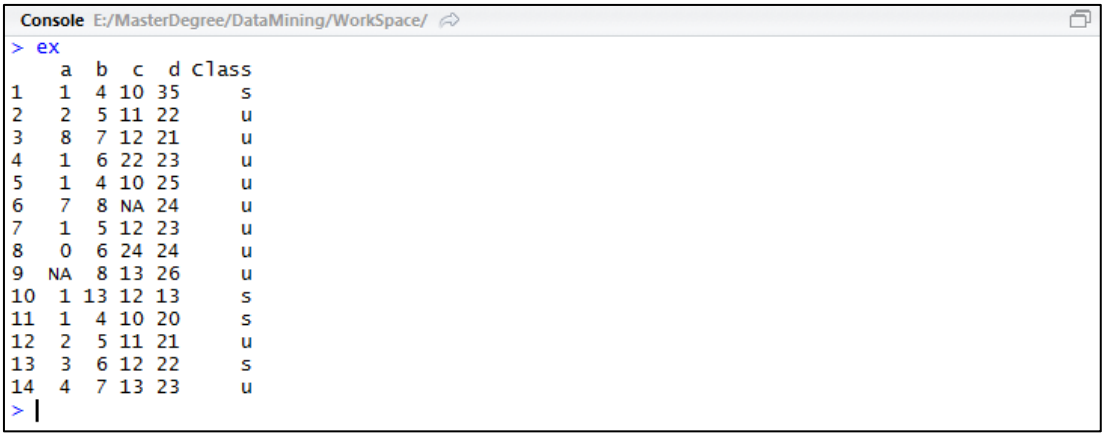
เนื้อหาในหัวข้อนี้จะเป็นการนำขั้นตอนและกระบวนการต่าง ๆ ที่ได้ออกแบบไว้ในหัวข้อ 3.2 มาพัฒนาโดยใช้ภาษาอาร์ในการพัฒนาโปรแกรมโดยแบ่งออกเป็นขั้นตอนต่าง ๆ ดังนี้

3.3.1 การเตรียมข้อมูลและนำเข้าข้อมูล

ข้อมูลที่สามารถใช้ในงานวิจัยนี้ได้จำเป็นต้องเป็นข้อมูลประเภทตัวเลข (Numerical Data) เท่านั้นและทำการนำเข้าข้อมูลในรูปแบบ csv (comma-separated value) ตัวอย่างเช่นการนำเข้าข้อมูลที่บันทึกอยู่ในไฟล์ example.csv (รูปที่ 3.11) สามารถนำเข้าข้อมูลได้โดยใช้คำสั่ง ex <- read.csv(“example.csv”) ข้อมูลที่นำเข้าและบันทึกอยู่ในตัวแปร ex แสดงได้ดังรูปที่ 3.12

```
a,b,c,d,Class
1,4,10,35,s
2,5,11,22,u
8,7,12,21,u
1,6,22,23,u
1,4,10,25,u
7,8,,24,u
1,5,12,23,u
0,6,24,24,u
,8,13,26,u
1,13,12,13,s
1,4,10,20,s
2,5,11,21,u
3,6,12,22,s
4,7,13,23,u
```

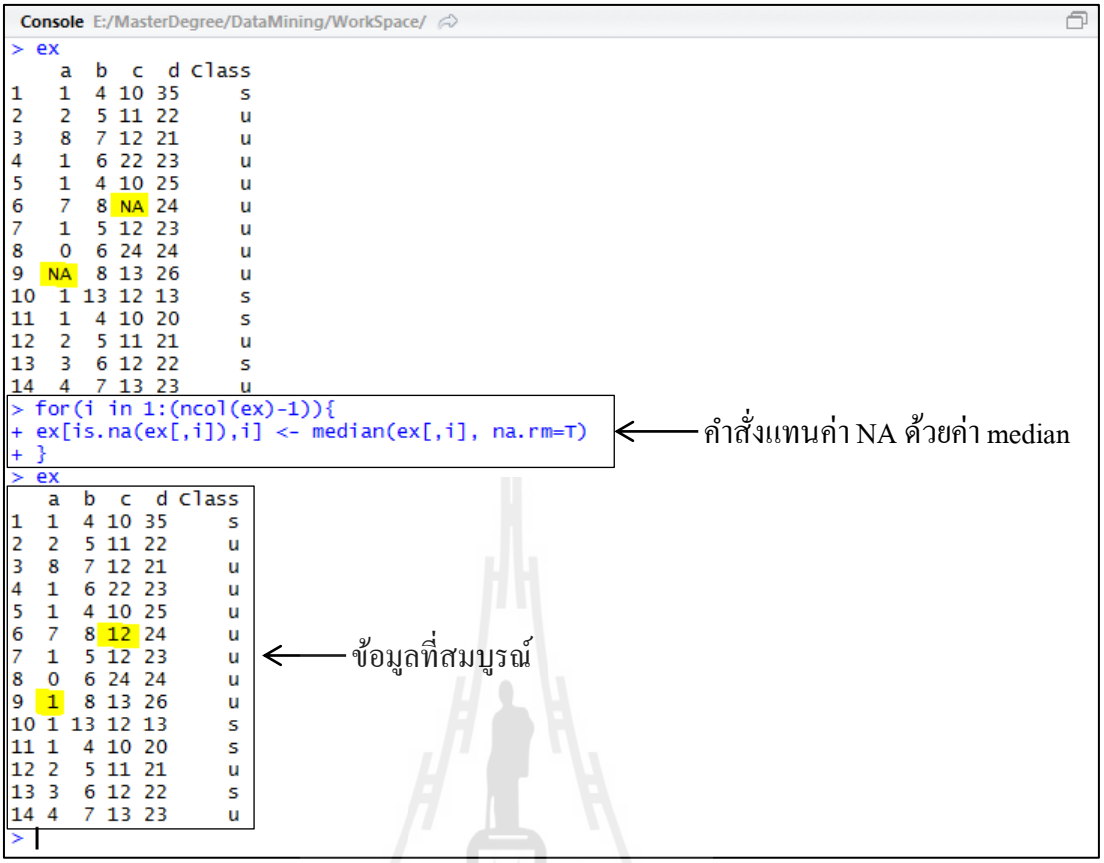
รูปที่ 3.11 แสดงข้อมูลในไฟล์ example.csv ที่มีข้อมูลไม่สมบูรณ์ในแถวที่ 6 และ 9



รูปที่ 3.12 แสดงตัวอย่างการนำเข้าข้อมูลในภาษาอาร์ด้วยไฟล์ example.csv

3.3.2 การแทนค่าที่ไม่สมบูรณ์หรือข้อมูลที่ขาดหายด้วยค่ากึ่งกลาง

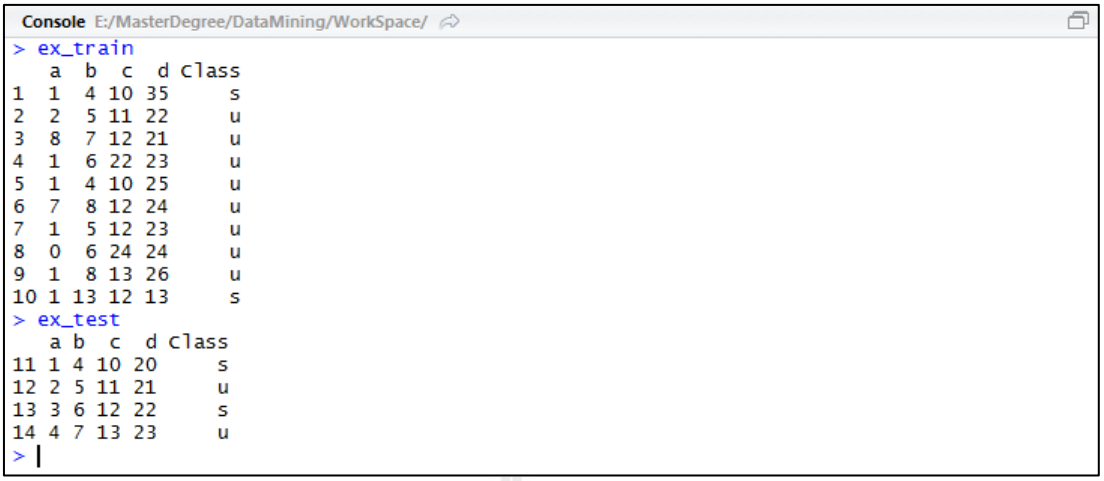
ในภาษาอาร์นั้นสามารถหาค่ากึ่งกลางได้โดยใช้คำสั่ง median() ซึ่งจะได้ค่ากึ่งกลางของข้อมูลและสามารถหาค่าที่ไม่สมบูรณ์หรือข้อมูลที่ขาดหายด้วยคำสั่ง is.na() แสดงตัวอย่างโดยการแทนค่าที่ไม่สมบูรณ์หรือข้อมูลที่ขาดหาย (แทนด้วย NA ในรูปที่ 3.12) ด้วยค่ากึ่งกลางจะได้ผลลัพธ์เป็นชุดข้อมูลใหม่ดังรูปที่ 3.13



รูปที่ 3.13 แสดงการแทนค่าที่ไม่สมบูรณ์หรือข้อมูลที่ขาดหายด้วยค่ากึ่งกลาง

3.3.3 แบ่งข้อมูลเป็นชุดข้อมูลฝึกสอนและชุดข้อมูลทดสอบ (Split Data)

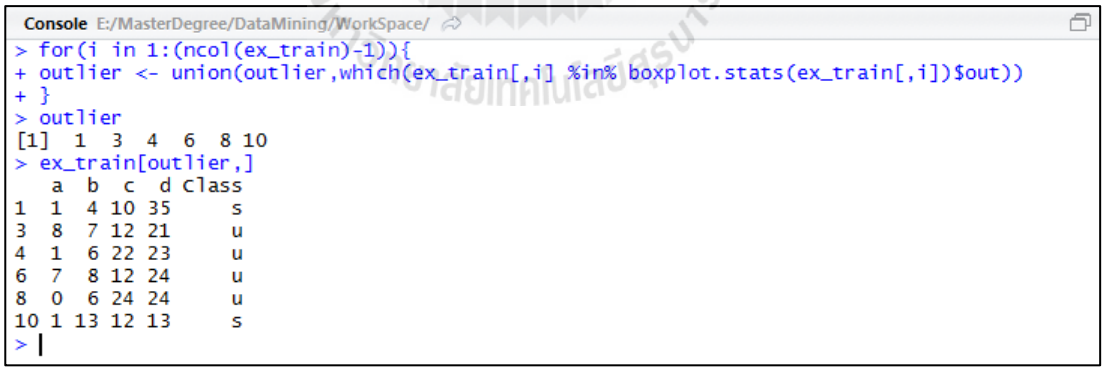
หลังจากแทนค่าที่ไม่สมบูรณ์หรือข้อมูลที่ขาดหายด้วยค่ากึ่งกลางแล้วทำการแบ่งข้อมูลเป็นชุดข้อมูลฝึกสอนในอัตราส่วน 70% และชุดข้อมูลทดสอบ 30% โดยให้ชื่อชุดข้อมูลว่า ex_train และ ex_test ดังรูปที่ 3.14



รูปที่ 3.14 แสดงผลลัพธ์ที่ได้จากการแบ่งข้อมูลเป็นชุดข้อมูลฝึกสอนและชุดข้อมูลทดสอบ

3.3.4 การค้นหาข้อมูลที่ผิดปกติทั้งหมดของทุก ๆ คอลัมน์ (Find Outliers)

หลังจากทำการแบ่งข้อมูลเป็นชุดข้อมูลฝึกสอนและชุดข้อมูลทดสอบเรียบร้อยแล้ว จะนำชุดข้อมูลฝึกสอนมาทำการค้นหาข้อมูลที่ผิดปกติทั้งหมดของทุก ๆ คอลัมน์ซึ่งในแต่ละคอลัมน์สามารถหาข้อมูลที่ผิดปกติได้โดยใช้คำสั่ง boxplot.stats()$out โดยคำสั่งนี้จะทำการคืนค่ากลับมาเป็นตำแหน่งของแถว (Row id) ตัวอย่างดังรูปที่ 3.15 ที่ปรากฏ Outlier ในแถวที่ 1, 3, 4, 8, และ 10



รูปที่ 3.15 แสดงการใช้คำสั่งค้นหาข้อมูลที่ผิดปกติทั้งหมดของทุก ๆ คอลัมน์

3.3.5 การเลือกคอลัมน์ด้วยค่าสหสัมพันธ์ (Select Features)

ในขั้นตอนนี้จะเป็นการนำชุดข้อมูลฝึกสอน (ex_train) มาทำการเลือกคอลัมน์ด้วยค่าสหสัมพันธ์ซึ่งในภาษาอาร์สามารถใช้คำสั่ง cor() ในการหาค่าสหสัมพันธ์ และทำตามกระบวนการตามรูปที่ 3.7 และ 3.8 โดยใช้ภาษาอาร์ได้ดังรูปที่ 3.16 โดยสร้างเป็นฟังก์ชันชื่อว่า feature_selection() และผลลัพธ์ของการคัดเลือกคอลัมน์แสดงดังรูปที่ 3.17 และเมื่อทำการเลือกคอลัมน์ด้วยค่าสหสัมพันธ์ผนวกกับขั้นตอนการค้นหาข้อมูลที่ผิดปกติทั้งหมดของทุก ๆ คอลัมน์จะได้ดังรูปที่ 3.18

```
function(data, max = 0.5){
    co <- abs(cor(data[,-ncol(data)]))
    co <- co[-nrow(co),-ncol(co)]
    name.co <- colnames(co)
    co.ind <- vector()
    while(ncol(co) >1){
        group <- which(co[,1] >= max)
        group.na <- names(group)[1]
        co.n <- ncol(co)
        if(co.n >2){
            co.ind <- append(co.ind,which(name.co == group.na))
        }

        if((co.n - length(group) == 1)){
            group.new <- which(co[,1] < max)
            co.ind <- append(co.ind,which(names(co[,1]) == names(group.new)))
            break
        }
        co <- co[-group,-group]
        if((co.n - length(group) == 2)){
            co.ind <- append(co.ind,which(name.co == names(co[,1])[1]))
            co.ind <- append(co.ind,which(name.co == names(co[,1])[2]))
            break
        }
    }
    co.ind <- append(co.ind,ncol(data))
    co.ind
}
```

รูปที่ 3.16 แสดงการเขียนฟังก์ชัน feature_selection() ในภาษาอาร์

```
Console E:/MasterDegree/DataMining/WorkSpace/
> feature_selection(ex_train,0.7)
[1] 1 2 3 5
> ex_train[,feature_selection(ex_train,0.7)]
   a  b  c Class
1  1  4 10     s
2  2  5 11     u
3  8  7 12     u
4  1  6 22     u
5  1  4 10     u
6  7  8 12     u
7  1  5 12     u
8  0  6 24     u
9  1  8 13     u
10 1 13 12     s
>
```

รูปที่ 3.17 แสดงผลลัพธ์ของการเลือกคอลัมน์ด้วยค่าสหสัมพันธ์

```
Console E:/MasterDegree/DataMining/WorkSpace/
> ex_train[outlier, feature_selection(ex_train,0.7)]
   a  b  c Class
1  1  4 10     s
3  8  7 12     u
4  1  6 22     u
6  7  8 12     u
8  0  6 24     u
10 1 13 12     s
>
```

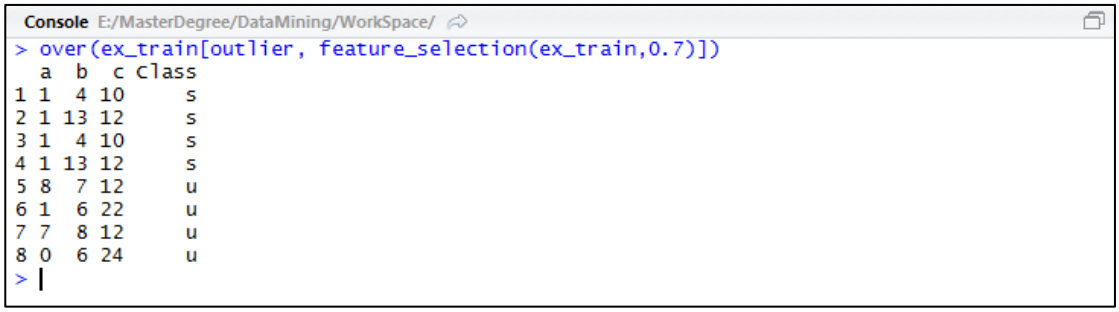
รูปที่ 3.18 แสดงผลลัพธ์จากการเลือกคอลัมน์ด้วยค่าสหสัมพันธ์ผนวกกับการค้นหาข้อมูลที่ผิดปกติทั้งหมดของทุก ๆ คอลัมน์

3.3.6 การสุ่มเกินของคลาสที่มีจำนวนน้อยกว่า (Over-Sampling)

ขั้นตอนนี้จะเป็นการทำการโปรแกรมด้วยภาษาอาร์เพื่อเพิ่มจำนวนคลาสที่มีจำนวนน้อยกว่า ด้วยขั้นตอนและกระบวนการจากรูปที่ 3.9 และ 3.10 โดยจะใช้ชื่อฟังก์ชันว่า over() รายละเอียดของคำสั่งในฟังก์ชันแสดงดังรูปที่ 3.19 และผลลัพธ์ที่ได้จากฟังก์ชัน over() แสดงได้ดังรูปที่ 3.20

```
function(data){
    over.table <- table(data[,ncol(data)])
    over.max.val <- max(over.table)
    over.max.ind <- which(over.table == over.max.val)
    over.out.ind <- vector()
    for(over.i in 1:length(over.table)){
        if(over.table[over.i] == 0){
            next
        }
        over.cal <- trunc(over.max.val / over.table[over.i])
        if(over.cal >= 2){
            for(over.j in 1:over.cal){
                over.out.ind <- append(over.out.ind, row.names(data[data[,ncol(data)] ==
                levels(factor(data[,ncol(data)]))[over.i],]))
            }
        }else{
            over.out.ind <- append(over.out.ind, row.names(data[data[,ncol(data)] ==
            levels(factor(data[,ncol(data)]))[over.i],]))
        }
    }
    over.out.data <- data[over.out.ind,]
    if(nrow(over.out.data) != 0){
        row.names(over.out.data) <- c(1:nrow(over.out.data))
    }
    over.out.data
}
```

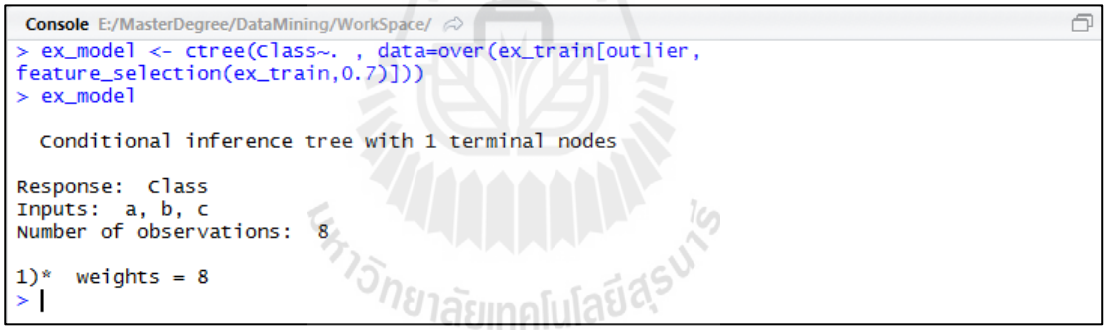
รูปที่ 3.19 แสดงฟังก์ชัน over() ในภาษาอาร์



รูปที่ 3.20 แสดงผลลัพธ์ของการสุ่มเกินของข้อมูลในคลาสที่มีจำนวนน้อยกว่า (คลาส s)

3.3.7 การสร้างโมเดล (Generate Model)

ขั้นตอนนี้จะเป็นการสร้างโมเดลจากข้อมูลที่มีการเพิ่มจำนวนข้อมูลในคลาสแล้ว ซึ่งในตัวอย่างนี้จะเป็นการนำข้อมูลจากรูปที่ 3.20 มาทำการสร้างโมเดลด้วยโครงสร้างต้นไม้ ซึ่งสามารถใช้คำสั่ง ctree() ในการสร้างโมเดลจากข้อมูลโดยเก็บโมเดลไว้ที่ตัวแปรชื่อ ex_model แสดงได้ดังรูปที่ 3.21 ซึ่งโมเดลที่ได้เป็นโมเดลที่ทำนายเป็นคลาส s เพียงอย่างเดียว



รูปที่ 3.21 แสดงโมเดลที่ได้จากข้อมูลที่ผ่านการเพิ่มจำนวนคลาสที่มีจำนวนน้อยกว่า

3.3.8 การทดสอบโมเดล

หลังจากที่ผ่านกระบวนการต่าง ๆ จนกระทั่งได้โมเดลจากชุดข้อมูลฝึกสอนแล้วจะทำการทดสอบโมเดลที่ได้ด้วยชุดข้อมูลทดสอบที่ได้แบ่งไว้ตั้งแต่ต้นซึ่งก็คือข้อมูล ex_test โดยในการทดสอบโมเดลสามารถใช้คำสั่ง predict() และใช้คำสั่ง table() ในการแสดง confusion matrix ดังรูปที่ 3.22 ซึ่งโมเดลที่ได้ทำนายข้อมูลทั้งหมดเป็นคลาส s โดยความถูกต้องของโมเดล 50% เมื่อทำการทดสอบกับข้อมูลชุดทดสอบ

```
Console E:/MasterDegree/DataMining/WorkSpace/
> predict(ex_model, newdata=ex_test)
[1] s s s s
Levels: s u
> table(predict(ex_model, newdata=ex_test),ex_test[,5])

   s u
  s 2 2
  u 0 0
>
```

รูปที่ 3.22 แสดงผลการทดสอบโมเดลที่ได้จากชุดข้อมูลฝึกสอนแล้วทดสอบด้วยชุดข้อมูลทดสอบ

3.4 เครื่องมือและอุปกรณ์ที่ใช้สำหรับการวิจัย

เครื่องมือและอุปกรณ์ที่ใช้สำหรับการวิจัยคือเครื่องคอมพิวเตอร์ส่วนบุคคลซึ่งมีรายละเอียดด้านประสิทธิภาพดังต่อไปนี้

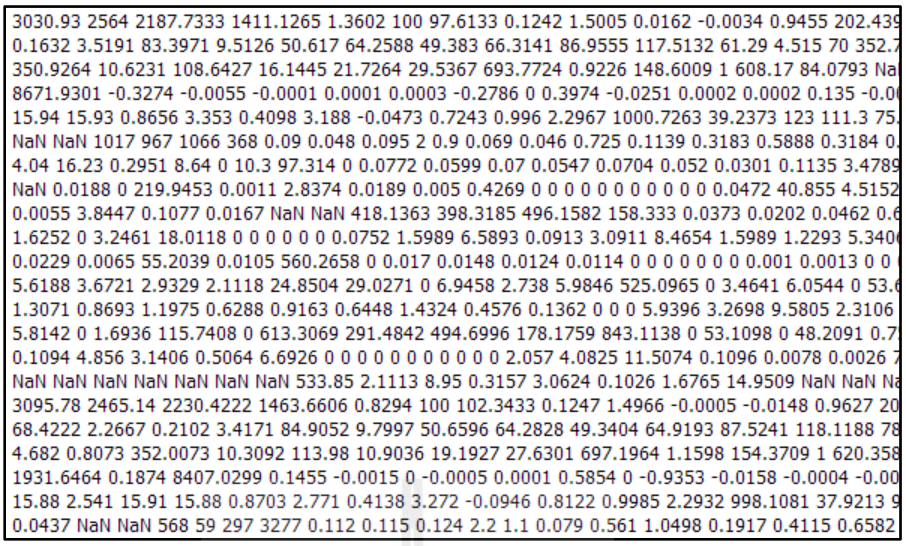
- หน่วยประมวลผลกลาง : AMD Phenom™ II X4 955 3.20 GHz
- หน่วยความจำสำรอง : 1 TB 7200 RPM
- หน่วยความจำหลัก : 4.00 GB DDR3
- ระบบปฏิบัติการ : Microsoft Windows 7 Ultimate 64-bit Operating System
- เครื่องมือในการพัฒนาโปรแกรม : RStudio, Microsoft Excel

บทที่ 4
การทดสอบและอภิปรายผล

ในบทที่ 4 นี้จะเป็นการทดสอบและอภิปรายผลในการใช้อัลกอริทึม RCD เพื่อค้นหาคลาสที่ค้นพบได้ยากสำหรับข้อมูลที่มีขนาดแตกต่างกันมาก โดยแบ่งออกเป็นส่วนย่อย ๆ คือ การเตรียมข้อมูลสำหรับการทดสอบ การออกแบบวิธีทดสอบ ผลการทดสอบ และอภิปรายผล

4.1 การเตรียมข้อมูลสำหรับการทดสอบ

ในการวิจัยนี้ได้นำข้อมูลจริงที่มีขนาดของคลาสที่แตกต่างกัน โดยนำข้อมูลจาก UCI Machine Learning Repository ซึ่งเป็นฐานข้อมูลที่ใช้ในการวิจัยด้านการทำเหมืองข้อมูลอย่างแพร่หลายโดยเลือกใช้ข้อมูล GLASS และ SECOM โดยตัวอย่างในการอธิบายขั้นตอนการทดสอบประสิทธิภาพจะใช้ข้อมูล SECOM เป็นตัวอย่างในการอธิบาย ในส่วนของผลการทดสอบจะแสดงผลของทั้งข้อมูล SECOM และ GLASS ข้อมูล SECOM เป็นข้อมูลกระบวนการผลิตสารกึ่งตัวนำโดยมีคลาสเป้าหมายคือ ผลการผลิตสารกึ่งตัวนำที่สามารถใช้งานได้ (Pass แทนด้วย -1 ซึ่งหมายถึง Negative) และสารกึ่งตัวนำที่ไม่สามารถใช้งานได้ (Fail แทนด้วย 1 ซึ่งหมายถึง Positive) โดยข้อมูล SECOM มีข้อมูลทั้งหมด 1567 เรคคอร์ด และมีเรคคอร์ดที่เป็นกลุ่มของสารกึ่งตัวนำที่สามารถใช้งานได้อยู่ 1463 เรคคอร์ด และมีกลุ่มข้อมูลของการผลิตสารกึ่งตัวนำที่ไม่สามารถใช้งานได้อยู่ 104 เรคคอร์ด โดยสารกึ่งตัวนำที่สามารถใช้งานได้คิดเป็นร้อยละ 93.36 ส่วนสารกึ่งตัวนำที่ไม่สามารถใช้งานได้คิดเป็นร้อยละ 6.64 จากข้อมูลทั้งหมด ซึ่งสามารถสังเกตได้ว่าข้อมูลสารกึ่งตัวนำที่สามารถใช้งานได้และข้อมูลสารกึ่งตัวนำที่ไม่สามารถใช้งานได้มีจำนวนข้อมูลที่แตกต่างกันมาก ตัวอย่างของข้อมูล SECOM ซึ่งแบ่งข้อมูลด้วยช่องว่างแสดงดังรูปที่ 4.1


3030.93 2564 2187.7333 1411.1265 1.3602 100 97.6133 0.1242 1.5005 0.0162 -0.0034 0.9455 202.43
0.1632 3.5191 83.3971 9.5126 50.617 64.2588 49.383 66.3141 86.9555 117.5132 61.29 4.515 70 352.
350.9264 10.6231 108.6427 16.1445 21.7264 29.5367 693.7724 0.9226 148.6009 1 608.17 84.0793 Na
8671.9301 -0.3274 -0.0055 -0.0001 0.0001 0.0003 -0.2786 0 0.3974 -0.0251 0.0002 0.0002 0.135 -0.0
15.94 15.93 0.8656 3.353 0.4098 3.188 -0.0473 0.7243 0.996 2.2967 1000.7263 39.2373 123 111.3 75.
NaN NaN 1017 967 1066 368 0.09 0.048 0.095 2 0.9 0.069 0.046 0.725 0.1139 0.3183 0.5888 0.3184 0.
4.04 16.23 0.2951 8.64 0 10.3 97.314 0 0.0772 0.0599 0.07 0.0547 0.0704 0.052 0.0301 0.1135 3.4789
NaN 0.0188 0 219.9453 0.0011 2.8374 0.0189 0.005 0.4269 0 0 0 0 0 0 0 0 0 0 0.0472 40.855 4.5152
0.0055 3.8447 0.1077 0.0167 NaN NaN 418.1363 398.3185 496.1582 158.333 0.0373 0.0202 0.0462 0.6
1.6252 0 3.2461 18.0118 0 0 0 0 0 0 0.0752 1.5989 6.5893 0.0913 3.0911 8.4654 1.5989 1.2293 5.340
0.0229 0.0065 55.2039 0.0105 560.2658 0 0.017 0.0148 0.0124 0.0114 0 0 0 0 0 0 0 0.001 0.0013 0 0
5.6188 3.6721 2.9329 2.1118 24.8504 29.0271 0 6.9458 2.738 5.9846 525.0965 0 3.4641 6.0544 0 53.
1.3071 0.8693 1.1975 0.6288 0.9163 0.6448 1.4324 0.4576 0.1362 0 0 0 5.9396 3.2698 9.5805 2.3106
5.8142 0 1.6936 115.7408 0 613.3069 291.4842 494.6996 178.1759 843.1138 0 53.1098 0 48.2091 0.7
0.1094 4.856 3.1406 0.5064 6.6926 0 0 0 0 0 0 0 0 0 0 0 2.057 4.0825 11.5074 0.1096 0.0078 0.0026 7
NaN NaN NaN NaN NaN NaN NaN 533.85 2.1113 8.95 0.3157 3.0624 0.1026 1.6765 14.9509 NaN NaN Na
3095.78 2465.14 2230.4222 1463.6606 0.8294 100 102.3433 0.1247 1.4966 -0.0005 -0.0148 0.9627 20
68.4222 2.2667 0.2102 3.4171 84.9052 9.7997 50.6596 64.2828 49.3404 64.9193 87.5241 118.1188 78
4.682 0.8073 352.0073 10.3092 113.98 10.9036 19.1927 27.6301 697.1964 1.1598 154.3709 1 620.358
1931.6464 0.1874 8407.0299 0.1455 -0.0015 0 -0.0005 0.0001 0.5854 0 -0.9353 -0.0158 -0.0004 -0.00
15.88 2.541 15.91 15.88 0.8703 2.771 0.4138 3.272 -0.0946 0.8122 0.9985 2.2932 998.1081 37.9213 9
0.0437 NaN NaN 568 59 297 3277 0.112 0.115 0.124 2.2 1.1 0.079 0.561 1.0498 0.1917 0.4115 0.6582

รูปที่ 4.1 แสดงข้อมูลก่อนการเตรียมข้อมูลของชุดข้อมูล SECOM จาก UCI

จากนั้นนำข้อมูล SECOM มาทำการเตรียมข้อมูลใหม่โดยใช้โปรแกรม Microsoft Excel ช่วยในการเตรียมข้อมูลโดยจะบันทึกข้อมูลให้อยู่ในรูปแบบ csv เพื่อที่จะได้ใช้งานได้อย่างสะดวกในการนำข้อมูลเข้าโปรแกรม RStudio โดยเมื่อเตรียมข้อมูลด้วยโปรแกรม Microsoft Excel แล้วจะได้ผลลัพธ์ดังรูปที่ 4.2

	A	B	C	D	E	F	G	H	I	J
1	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
2	3030.93	2564	2187.733	1411.127	1.3602	100	97.6133	0.1242	1.5005	0.0162
3	3095.78	2465.14	2230.422	1463.661	0.8294	100	102.3433	0.1247	1.4966	-0.0005
4	2932.61	2559.94	2186.411	1698.017	1.5102	100	95.4878	0.1241	1.4436	0.0041
5	2988.72	2479.9	2199.033	909.7926	1.3204	100	104.2367	0.1217	1.4882	-0.0124
6	3032.24	2502.87	2233.367	1326.52	1.5334	100	100.3967	0.1235	1.5031	-0.0031
7	2946.25	2432.84	2233.367	1326.52	1.5334	100	100.3967	0.1235	1.5287	0.0167
8	3030.27	2430.12	2230.422	1463.661	0.8294	100	102.3433	0.1247	1.5816	-0.027
9	3058.88	2690.15	2248.9	1004.469	0.7884	100	106.24	0.1185	1.5153	0.0157
10	2967.68	2600.47	2248.9	1004.469	0.7884	100	106.24	0.1185	1.5358	0.0111
11	3016.11	2428.37	2248.9	1004.469	0.7884	100	106.24	0.1185	1.5381	0.0159
12	2994.05	2548.21	2195.122	1046.147	1.3204	100	103.34	0.1223	1.5144	-0.019
13	2928.84	2479.4	2196.211	1605.758	0.9959	100	97.9156	0.1257	1.469	0.017
14	2920.07	2507.4	2195.122	1046.147	1.3204	100	103.34	0.1223	1.531	-0.0259
15	3051.44	2529.27	2184.433	877.6266	1.4668	100	107.8711	0.124	1.5236	-0.0209

รูปที่ 4.2 แสดงข้อมูล SECOM หลังจากใช้โปรแกรม Microsoft Excel ช่วยในการเตรียมข้อมูล

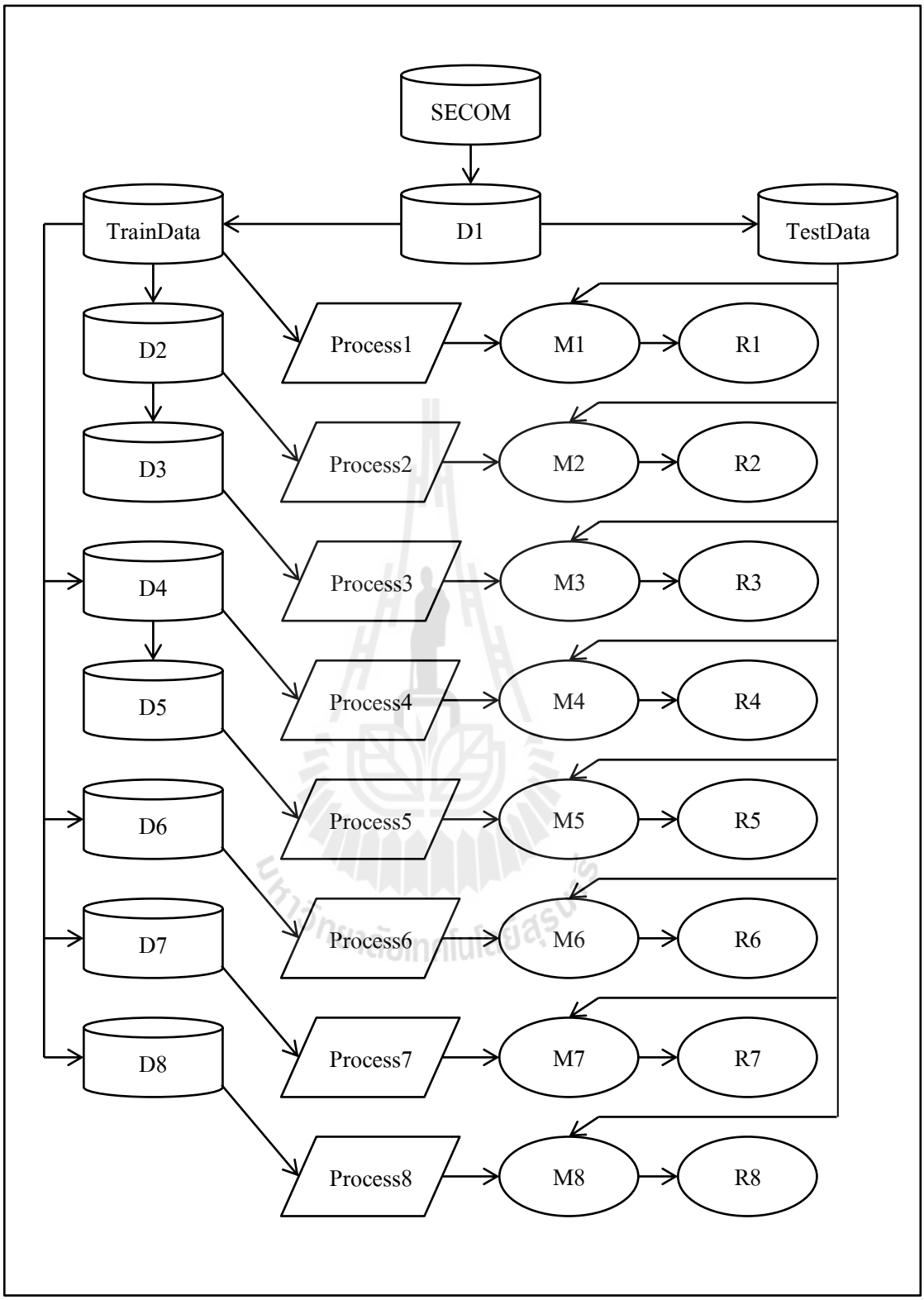
เมื่อได้ข้อมูลที่เป็นไฟล์ csv ที่เตรียมไว้เรียบร้อยแล้วสามารถนำข้อมูลเข้ามาใช้งานในโปรแกรม RStudio ได้โดยใช้คำสั่ง “secom <- read.csv(“secom.csv”)” ซึ่งเป็นการเก็บข้อมูลทั้งหมดไว้ในตัวแปรชื่อ secom เพื่อความสะดวกในการเรียกใช้

4.2 การออกแบบวิธีทดสอบ

สำหรับการออกแบบวิธีทดสอบนั้นได้ออกแบบไว้ดังรูปที่ 4.3 ซึ่งเป็นการทดสอบเทคนิคต่าง ๆ ที่ใช้ในการค้นหาคลาสที่ค้นพบได้ยาก ในงานวิจัยนี้ได้ออกแบบวิธีทดสอบด้วย 8 เทคนิค ได้แก่ Process 1 ถึง Process 8 ตามรูปที่ 4.3 ทั้ง 8 เทคนิคมีขั้นตอนเริ่มต้นเหมือนกันคือนำเข้าข้อมูลและเตรียมข้อมูลให้สมบูรณ์ แบ่งข้อมูลเป็นชุดฝึกสอนและชุดทดสอบ จากนั้นทดสอบใช้วิธีการต่าง ๆ ในอัลกอริทึม RCD สรุปได้ดังนี้

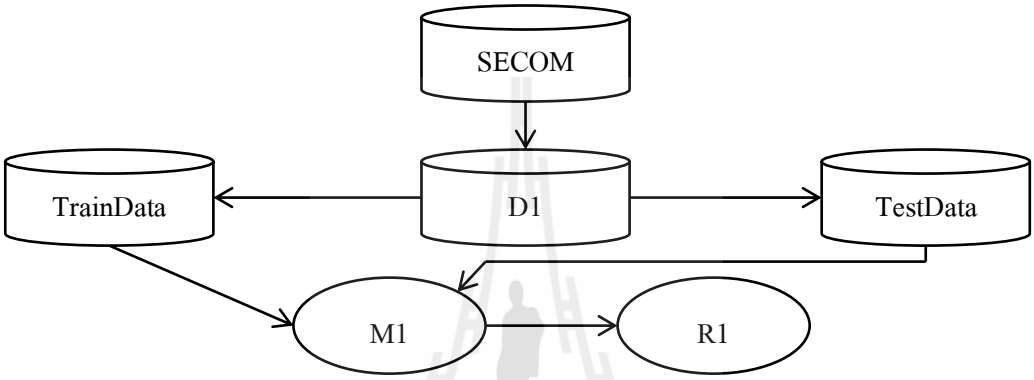
- Process 1 สร้างโมเดลด้วย Ctree (โดยไม่ต้องใช้เทคนิคใด ๆ เพิ่มเติม)
- Process 2 คัดเลือกคอลัมน์ด้วยการวิเคราะห์สหสัมพันธ์ และใช้แถวที่เป็นข้อมูล Outlier
- Process 3 คัดเลือกคอลัมน์และเลือกแถวเหมือน Process 2 จากนั้นใช้เทคนิค Over-Sampling ให้ขนาดของทุกคลาสสมดุลกัน
- Process 4 ใช้เฉพาะเทคนิคคัดเลือกคอลัมน์ (โดยไม่ทำ Over-Sampling และไม่เลือก Outlier)
- Process 5 คัดเลือกคอลัมน์ด้วยการวิเคราะห์สหสัมพันธ์ แบ่งข้อมูลฝึกสอนเป็นกลุ่มย่อย และสร้างโมเดลในแต่ละกลุ่มย่อย
- Process 6 ใช้เทคนิคคัดเลือกคอลัมน์ และการสุ่มเกิน (Over-Sampling) โดยไม่ต้องพิจารณาเกี่ยวกับ Outlier
- Process 7 ใช้เทคนิคคัดเลือกคอลัมน์ การสุ่มเกิน และตัดข้อมูล Outlier ทิ้ง
- Process 8 ใช้เทคนิคการสุ่มเกินเพียงอย่างเดียว

รายละเอียดแต่ละ Process อธิบายได้ดังต่อไปนี้



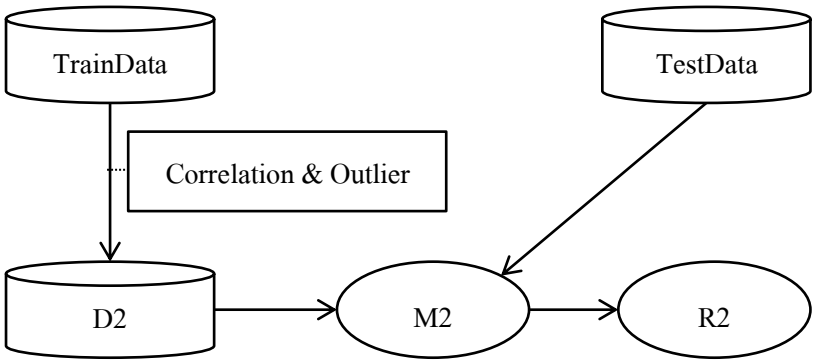
รูปที่ 4.3 แสดงแผนภาพการออกแบบวิธีดำเนินการวิจัยเพื่อทดสอบประสิทธิภาพของการค้นหาคลาสที่ค้นพบได้ยาก

Process1 ทำการค้นหาและแทนที่ค่าที่ไม่ทราบค่าของข้อมูล SECOM ด้วยค่ากึ่งกลางข้อมูล (Median) จนได้ข้อมูล SECOM ที่มีข้อมูลครบถ้วน (D1) แล้วนำข้อมูล (D1) มาทำการแบ่งเป็น 2 กลุ่มข้อมูลคือกลุ่มข้อมูลฝึกสอน และกลุ่มข้อมูลทดสอบ โดยแบ่งเป็นกลุ่มข้อมูลฝึกสอน (TrainData) 70% และกลุ่มข้อมูลทดสอบ (TestData) 30% แล้วจึงทำการสร้างโมเดลด้วยโครงสร้างต้นไม้ (M1) จากกลุ่มข้อมูลฝึกสอน และนำกลุ่มข้อมูลทดสอบมาทำการทดสอบโมเดลจึงทำให้ได้ค่าความถูกต้องของโมเดล (R1) ดังรูปที่ 4.4



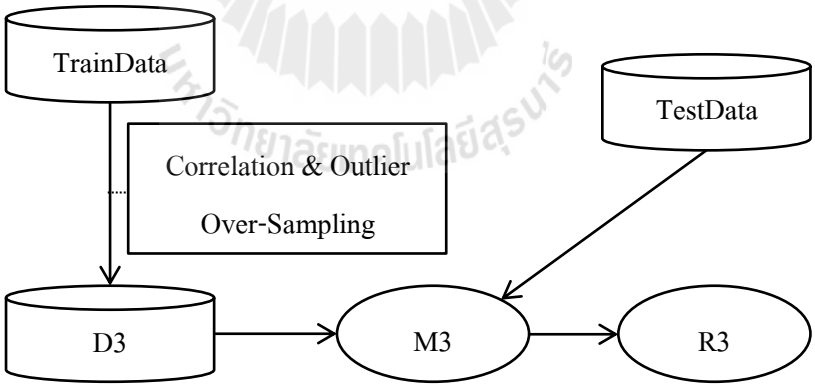
รูปที่ 4.4 แสดงวิธีการดำเนินการวิจัยของ Process1

Process2 นำกลุ่มข้อมูลฝึกสอนจาก Process1 มาทำการหาค่าสหสัมพันธ์ระหว่างคอลัมน์ทุก ๆ คอลัมน์ ด้วยค่าสหสัมพันธ์โดยยกเว้นคอลัมน์คลาส แล้วทำการคัดเลือกคอลัมน์ที่มีค่าสหสัมพันธ์มากกว่าที่ผู้ใช้กำหนด (ในที่นี้ใช้ค่าสหสัมพันธ์ขั้นต่ำที่ 0.5) จะถือว่าอยู่กลุ่มเดียวกันและเลือกตัวแทนกลุ่มออกมาเพียงคอลัมน์เดียว แต่ถ้าหากค่าสหสัมพันธ์น้อยกว่าที่ผู้ใช้กำหนด จะทำการเลือกคอลัมน์นั้นออกมาเพียงคอลัมน์เดียว และทำการหาเรคคอร์ดที่เป็นข้อมูลที่ผิดปกติ (Outlier) ในทุก ๆ คอลัมน์ แล้วสร้างข้อมูลใหม่ (D2) ซึ่งมีเฉพาะข้อมูลที่ผิดปกติจากชุดข้อมูลฝึกสอน และมีจำนวนคอลัมน์ที่ลดลงหรืออาจจะเท่าเดิมถ้าหากทุก ๆ คอลัมน์ไม่มีคอลัมน์ใดเลยที่มีความสัมพันธ์กันมากกว่าที่กำหนด (0.5 ซึ่งเป็นค่าที่เหมาะสมเนื่องจากไม่ทำให้การคัดเลือกคอลัมน์ตัดคอลัมน์ออกมากเกินความจำเป็น และถ้าหากกำหนดมากเกินไปก็จะไม่มีประโยชน์เนื่องจากจะไม่สามารถลดคอลัมน์ในขั้นตอนนี้ได้เลย) หลังจากนั้นนำข้อมูลที่สร้างขึ้นมาใหม่ (D2) มาทำการสร้างโมเดลด้วยโครงสร้างต้นไม้ (M2) จากนั้นทำการทดสอบโมเดลด้วยข้อมูลทดสอบ (TestData) และได้ค่าความถูกต้องของโมเดล (R2) ดังรูปที่ 4.5



รูปที่ 4.5 แสดงวิธีการดำเนินการวิจัยของ Process2

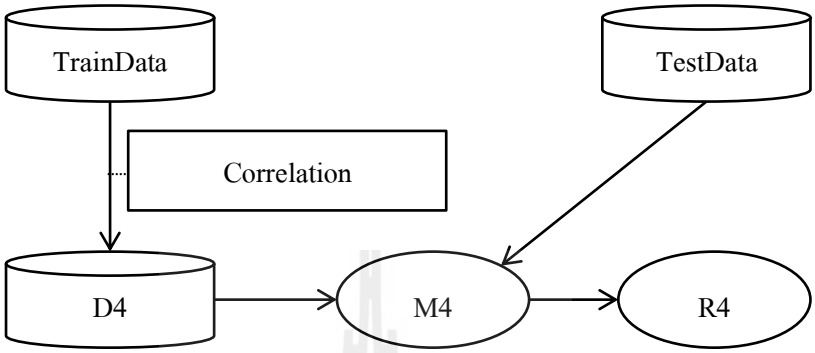
Process3 สร้างข้อมูลใหม่ (D3) จากชุดข้อมูลฝึกสอนซึ่งทำการคัดเลือกคอลัมน์ด้วยค่าสหสัมพันธ์และเลือกเฉพาะข้อมูลที่เป็นข้อมูลที่ผิดปกติแล้วจึงทำการสุ่มเกิน โดยทำการหาคลาสที่มีจำนวนข้อมูลมากที่สุด แล้วทำการเพิ่มข้อมูลในคลาสที่มีจำนวนน้อยกว่า 2 เท่าของคลาสที่มีจำนวนข้อมูลมากที่สุดในลักษณะของการสุ่มเกิน (Over-Sampling) โดยทำการเพิ่มแบบเท่าตัว เมื่อเพิ่มแล้วจะมีจำนวนไม่มากกว่าข้อมูลในคลาสที่มีจำนวนข้อมูลมากที่สุด หลังจากนั้นนำข้อมูลที่ได้สร้างเป็นชุดข้อมูลใหม่ (D3) มาทำการสร้างโมเดลด้วยโครงสร้างต้นไม้ (M3) จากนั้นทำการทดสอบโมเดลด้วยข้อมูลทดสอบ (TestData) และได้ค่าความถูกต้องของโมเดล (R3) ดังรูปที่ 4.6



รูปที่ 4.6 แสดงวิธีการดำเนินการวิจัยของ Process3

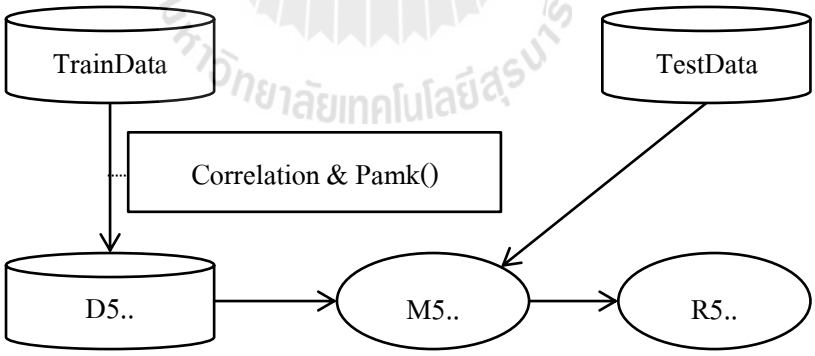
Process4 นำกลุ่มข้อมูลฝึกสอนจาก Process1 มาทำการหาความสัมพันธ์ระหว่างคอลัมน์ทุก ๆ คอลัมน์ ยกเว้นคอลัมน์คลาส แล้วทำการคัดเลือกคอลัมน์ที่มีค่าสหสัมพันธ์มากกว่าที่ผู้ใช้กำหนด (ในที่นี้ใช้ค่าสหสัมพันธ์ 0.5) จะถือว่าอยู่กลุ่มเดียวกันและเลือกตัวแทนกลุ่มออกมาเพียงคอลัมน์เดียว แต่ถ้าหากค่าสหสัมพันธ์น้อยกว่าที่กำหนด (0.5) จะเลือกคอลัมน์นั้นออกมาเพียง

คอลัมน์เดียว โดยสร้างเป็นข้อมูลชุดใหม่ (D4) แล้วทำการสร้างโมเดลด้วยโครงสร้างต้นไม้ (M4) จากนั้นทดสอบโมเดลด้วยข้อมูลทดสอบ (TestData) และได้ค่าความถูกต้องของโมเดล (R4) ดังรูปที่ 4.7



รูปที่ 4.7 แสดงวิธีการดำเนินการวิจัยของ Process4

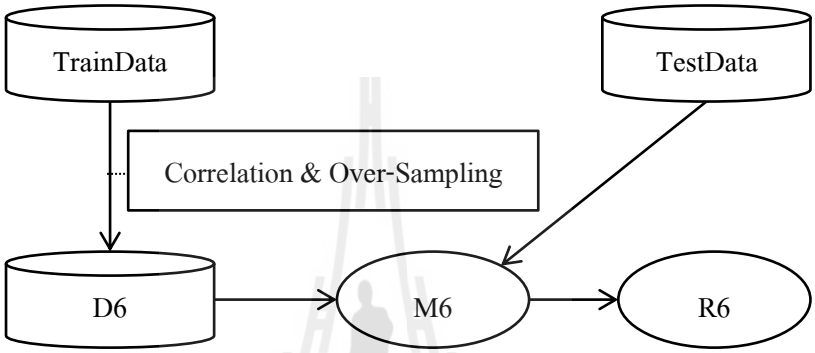
Process5 สร้างข้อมูลใหม่ (D5) จากชุดข้อมูลฝึกสอนมาทำการคัดเลือกคอลัมน์ด้วยค่าสหสัมพันธ์และทำการจัดกลุ่ม (Clustering) ด้วยฟังก์ชัน pamk ซึ่งจะได้จำนวนกลุ่มที่เหมาะสม แล้วจึงนำกลุ่มข้อมูลทุกกลุ่มทำการสร้างโมเดลด้วยโครงสร้างต้นไม้ (M5..) และทำการทดสอบโมเดลด้วยข้อมูลทดสอบ (TestData) ได้ค่าความถูกต้องของโมเดล (R5..) ดังรูปที่ 4.8



รูปที่ 4.8 แสดงวิธีการดำเนินการวิจัยของ Process5

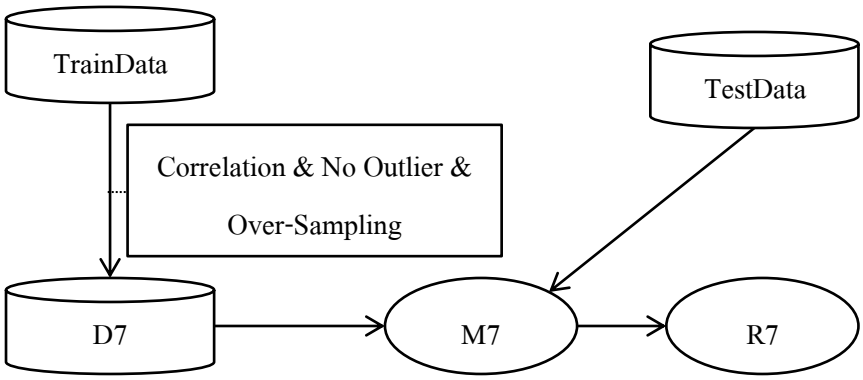
Process6 นำกลุ่มข้อมูลฝึกสอนจาก Process1 มาทำการหาค่าสหสัมพันธ์ระหว่างคอลัมน์ทุก ๆ คอลัมน์ยกเว้นคอลัมน์คลาส แล้วทำการคัดเลือกคอลัมน์ที่มีค่าสหสัมพันธ์มากกว่าที่ผู้ใช้กำหนด (ในที่นี้ใช้ค่าสหสัมพันธ์ 0.5) จะถือว่าอยู่กลุ่มเดียวกันและเลือกตัวแทนกลุ่มออกมา

เพียงคอลัมน์เดียว แต่ถ้าหากค่าสหสัมพันธ์น้อยกว่าที่กำหนด (0.5) จะเลือกคอลัมน์นั้นออกมาเพียงคอลัมน์เดียว แล้วทำการหาคลาสที่มีจำนวนข้อมูลมากที่สุด แล้วทำการเพิ่มข้อมูลคลาสที่มีจำนวนน้อยกว่า 2 เท่าของคลาสที่มีจำนวนข้อมูลมากที่สุด โดยทำการเพิ่มแบบเท่าตัวเมื่อเพิ่มแล้วจะต้องมีจำนวนไม่มากกว่าคลาสที่มีจำนวนข้อมูลมากที่สุด หลังจากนั้นสร้างเป็นชุดข้อมูลใหม่แล้วนำข้อมูลนี้ (D6) มาทำการสร้างโมเดลด้วยโครงสร้างต้นไม้ (M6) จากนั้นทดสอบโมเดลด้วยข้อมูลทดสอบ (TestData) และได้ค่าความถูกต้องของโมเดล (R6) ดังรูปที่ 4.9



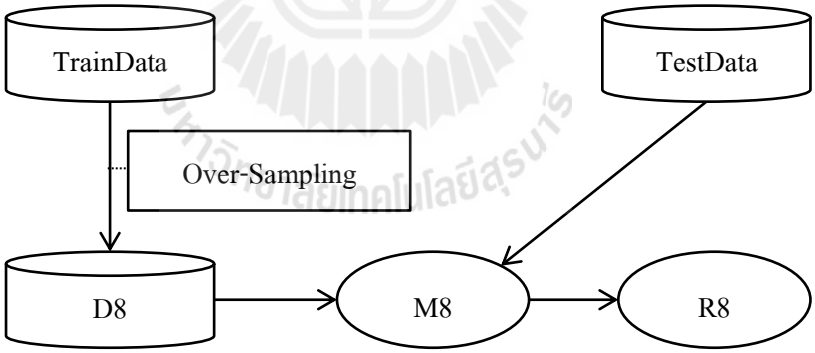
รูปที่ 4.9 แสดงวิธีการดำเนินการวิจัยของ Process6

Process7 นำกลุ่มข้อมูลฝึกสอนจาก Process1 มาทำการหาค่าสหสัมพันธ์ระหว่างคอลัมน์ทุก ๆ คอลัมน์ (Correlation) ยกเว้นคอลัมน์คลาส แล้วทำการคัดเลือกคอลัมน์ที่มีค่าสหสัมพันธ์มากกว่าที่กำหนด (ในที่นี้ใช้ค่าความสัมพันธ์ 0.5) จะถือว่าอยู่กลุ่มเดียวกันและเลือกตัวแทนกลุ่มออกมาเพียงคอลัมน์เดียว แต่ถ้าหากความสัมพันธ์น้อยกว่าที่กำหนด (0.5) จะเลือกคอลัมน์นั้นออกมาเพียงคอลัมน์เดียว แล้วทำการลบเรคคอร์ดที่เป็นข้อมูลผิดปกติจากทุก ๆ คอลัมน์ แล้วจึงทำการหาคลาสที่มีจำนวนข้อมูลมากที่สุด แล้วทำการเพิ่มข้อมูลคลาสที่มีจำนวนน้อยกว่า 2 เท่าของคลาสที่มีจำนวนข้อมูลมากที่สุด โดยทำการเพิ่มแบบเท่าตัวเมื่อเพิ่มแล้วจะต้องมีจำนวนไม่มากกว่าคลาสที่มีจำนวนข้อมูลมากที่สุด หลังจากนั้นนำข้อมูลที่ได้สร้างเป็นชุดข้อมูลใหม่ (D7) แล้วทำการสร้างโมเดลด้วยโครงสร้างต้นไม้ (M7) จากนั้นทดสอบโมเดลด้วยข้อมูลทดสอบ (TestData) และได้ค่าความถูกต้องของโมเดล (R7) ดังรูปที่ 4.10



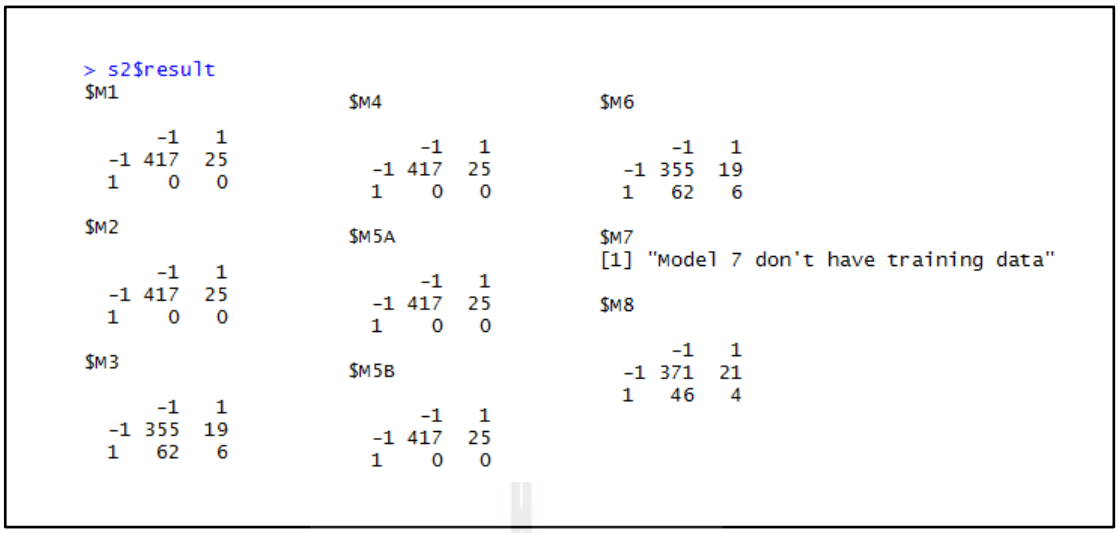
รูปที่ 4.10 แสดงวิธีการดำเนินการวิจัยของ Process7

Process8 สร้างข้อมูลใหม่ (D8) จากข้อมูลใน Process1 (TrainData) โดยทำการหาคลาสที่มีจำนวนข้อมูลมากที่สุด แล้วทำการเพิ่มข้อมูลคลาสที่มีจำนวนน้อยกว่า 2 เท่าของคลาสที่มีจำนวนข้อมูลมากที่สุด โดยทำการเพิ่มแบบเท่าตัวเมื่อเพิ่มแล้วจะมีจำนวนไม่มากกว่าคลาสที่มีจำนวนข้อมูลมากที่สุด หลังจากนั้นนำข้อมูลที่ได้สร้างเป็นชุดข้อมูลใหม่ (D8) มาทำการสร้างโมเดลด้วยโครงสร้างต้นไม้ (M8) จากนั้นทำการทดสอบโมเดลด้วยข้อมูลทดสอบ (TestData) และได้ค่าความถูกต้องของโมเดล (R8) ดังรูปที่ 4.11



รูปที่ 4.11 แสดงวิธีการดำเนินการวิจัยของ Process8

เมื่อทำการเขียนโปรแกรมเรียบร้อยแล้วทำการทดสอบโปรแกรมว่าสามารถทำงานได้ปกติดีหรือไม่โดยการทดสอบรันโปรแกรมข้างต้น ผลที่ได้แสดงดังรูปที่ 4.12



รูปที่ 4.12 แสดงตัวอย่างผลการรันโปรแกรมด้วยข้อมูล SECOM

4.3 ผลการทดสอบ

จากการออกแบบการทดสอบในหัวข้อที่ 4.2 ได้ใช้ข้อมูล SECOM และ GLASS ในการทดสอบซึ่งได้แสดงผลทุกขั้นตอนที่ได้ออกแบบดังต่อไปนี้ โดยข้อมูลแต่ละชุดข้อมูลมีรายละเอียดแสดงดังตารางที่ 4.1

ตารางที่ 4.1 แสดงรายละเอียดชุดข้อมูล SECOM และ GLASS

Data Set	Instances	Attributes	Class					
			1	2	3	4	5	6
SECOM	1567	591	1463	104	-	-	-	-
GLASS	214	10	70	76	17	13	9	29

ผลลัพธ์จาก Process1 ซึ่งเป็นขั้นตอนการแทนค่าที่สูญหายหรือค่าที่ผิดปกติแล้วทำการแยกเป็นชุดข้อมูลฝึกสอนและชุดข้อมูลทดสอบแสดงในตารางที่ 4.2 และ 4.3 หลังจากนั้นทำการสร้างโมเดลจากชุดข้อมูลฝึกสอนและทำการทดสอบด้วยข้อมูลทดสอบแสดงดังรูปที่ 4.13

ตารางที่ 4.2 แสดงรายละเอียดชุดข้อมูลฝึกสอนและชุดข้อมูลทดสอบของข้อมูล SECOM

Data Set	Instances	Attributes	Class	
			-1	1
Train	1078	591	1000	78
Test	489	591	463	26

ตารางที่ 4.3 แสดงรายละเอียดชุดข้อมูลฝึกสอนและชุดข้อมูลทดสอบของข้อมูล GLASS

Data Set	Instances	Attributes	Class					
			1	2	3	5	6	7
Train	152	10	49	57	12	9	6	19
Test	62	10	21	19	5	4	3	10

SECOM (M1)

	-1	1
-1	463	26
1	0	0

GLASS (M1)

	1	2	3	5	6	7
1	17	4	4	0	0	1
2	4	11	1	1	1	2
3	0	0	0	0	0	0
5	0	3	0	3	0	2
6	0	0	0	0	0	0
7	0	1	0	0	2	5

รูปที่ 4.13 แสดงผลการทดสอบโมเดล M1 จาก Process1

ผลลัพธ์จาก Process2 นำกลุ่มข้อมูลทดสอบจากขั้นตอนที่ 1 มาทำการเลือกคอลัมน์ด้วยค่าสหสัมพันธ์ และทำการเลือกเฉพาะข้อมูลที่ผิดปกติ ผลแสดงดังตารางที่ 4.4 และ 4.5 หลังจากนั้นทำการสร้างโมเดลจากชุดข้อมูล D2 และทำการทดสอบด้วยข้อมูลทดสอบแสดงดังรูปที่ 4.14

ตารางที่ 4.4 แสดงรายละเอียดข้อมูลฝึกสอนที่ผ่าน Process2 ของข้อมูล SECOM

Data Set	Instances	Attributes	Class	
			-1	1
D2	1078	304	1000	78

ตารางที่ 4.5 แสดงรายละเอียดข้อมูลฝึกสอนที่ผ่าน Process2 ของข้อมูล GLASS

Data Set	Instances	Attributes	Class					
			1	2	3	5	6	7
D2	55	6	10	16	1	7	2	19

SECOM (M2)

	-1	1
-1	463	26
1	0	0

GLASS (M2)

	1	2	3	5	6	7
1	19	12	5	0	0	2
2	2	5	0	3	0	2
3	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0
7	0	2	0	1	3	6

รูปที่ 4.14 แสดงผลการทดสอบโมเดล M2 จาก Process2

ผลลัพธ์จาก Process3 นำข้อมูล D2 มาทำการเพิ่มจำนวนคลาสที่น้อยกว่าด้วยวิธี Over-Sampling ผลที่ได้แสดงในตารางที่ 4.6 และ 4.7 หลังจากนั้นทำการสร้างโมเดลจากชุดข้อมูล D3 และทำการทดสอบด้วยข้อมูลทดสอบแสดงดังรูปที่ 4.15

ตารางที่ 4.6 แสดงรายละเอียดข้อมูลฝึกสอนที่ผ่าน Process3 ของข้อมูล SECOM

Data Set	Instances	Attributes	Class	
			-1	1
D3	1936	304	1000	936

ตารางที่ 4.7 แสดงรายละเอียดข้อมูลฝึกสอนที่ผ่าน Process3 ของข้อมูล GLASS

Data Set	Instances	Attributes	Class					
			1	2	3	5	6	7
D3	96	6	10	16	19	14	18	19

SECOM (M3)

	-1	1
-1	403	18
1	60	8

GLASS (M3)

	1	2	3	5	6	7
1	13	8	3	0	0	2
2	0	0	0	0	0	0
3	6	4	2	0	0	0
5	2	5	0	3	0	2
6	0	1	0	0	1	0
7	0	1	0	1	2	6

รูปที่ 4.15 แสดงผลการทดสอบโมเดล M3 จาก Process3

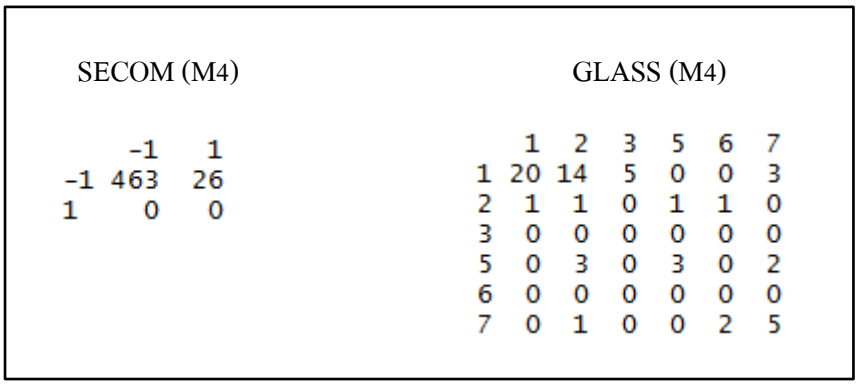
ผลลัพธ์จาก Process4 นำข้อมูลฝึกสอนมาทำการคัดเลือกคอลัมน์ด้วยค่าสหสัมพันธ์ ผลที่ได้แสดงในตารางที่ 4.8 และ 4.9 หลังจากนั้นทำการสร้างโมเดลจากชุดข้อมูล D4 และทำการทดสอบด้วยข้อมูลทดสอบแสดงดังรูปที่ 4.16

ตารางที่ 4.8 แสดงรายละเอียดข้อมูลฝึกสอนที่ผ่าน Process4 ของข้อมูล SECOM

Data Set	Instances	Attributes	Class	
			-1	1
D4	1078	304	1000	78

ตารางที่ 4.9 แสดงรายละเอียดข้อมูลฝึกสอนที่ผ่าน Process4 ของข้อมูล GLASS

Data Set	Instances	Attributes	Class					
			1	2	3	5	6	7
D4	152	6	46	57	12	9	6	19



รูปที่ 4.16 แสดงผลการทดสอบโมเดล M4 จาก Process4

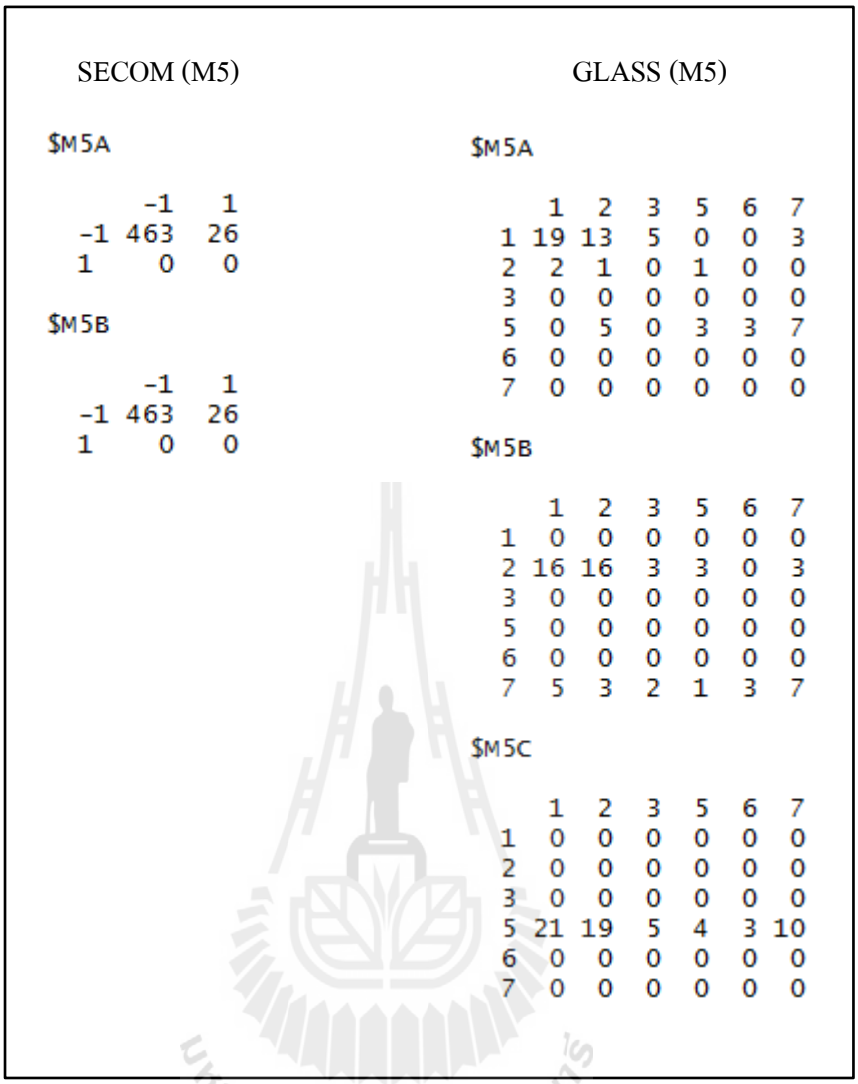
ผลลัพธ์จาก Process5 นำข้อมูลในขั้นตอนที่ 4 (D4) มาทำการจัดกลุ่ม (Clustering) ด้วยฟังก์ชัน pamk ผลที่ได้แสดงในตารางที่ 4.10 และ 4.11 หลังจากนั้นทำการสร้างโมเดลจากชุดข้อมูลทุกกลุ่มที่ได้และทำการทดสอบด้วยข้อมูลทดสอบแสดงดังรูปที่ 4.17

ตารางที่ 4.10 แสดงรายละเอียดข้อมูลฝึกสอนที่ผ่าน Process5 ของข้อมูล SECOM

Data Set	Instances	Attributes	Class	
			-1	1
A	838	304	781	57
B	240	304	219	21

ตารางที่ 4.11 แสดงรายละเอียดข้อมูลฝึกสอนที่ผ่าน Process5 ของข้อมูล GLASS

Data Set	Instances	Attributes	Class					
			1	2	3	5	6	7
A	118	6	49	50	12	3	3	1
B	32	6	0	7	0	4	3	18
C	2	6	0	0	0	2	0	0



รูปที่ 4.17 แสดงผลการทดสอบโมเดล M5 จาก Process5

ผลลัพธ์จาก Process6 นำข้อมูลฝึกสอนมาผ่านกระบวนการคัดเลือกคอลัมน์ด้วยค่าสหสัมพันธ์ และทำการเพิ่มจำนวนข้อมูลในคลาสด้วยวิธี Over-Sampling กำหนดให้ข้อมูลที่ผ่านกระบวนการดังกล่าวชื่อว่าชุดข้อมูล D6 แสดงดังในตารางที่ 4.12 และ 4.13 หลังจากนั้นทำการสร้างโมเดลจากชุดข้อมูล D6 และทำการทดสอบด้วยข้อมูลทดสอบแสดงดังรูปที่ 4.18

ตารางที่ 4.12 แสดงรายละเอียดข้อมูลฝึกสอนที่ผ่าน Process6 ของข้อมูล SECOM

Data Set	Instances	Attributes	Class	
			-1	1
D6	1936	304	1000	936

ตารางที่ 4.13 แสดงรายละเอียดข้อมูลฝึกสอนที่ผ่าน Process6 ของข้อมูล GLASS

Data Set	Instances	Attributes	Class					
			1	2	3	5	6	7
D6	319	6	46	57	48	54	54	57

SECOM (M6)

	-1	1
-1	403	18
1	60	8

GLASS (M6)

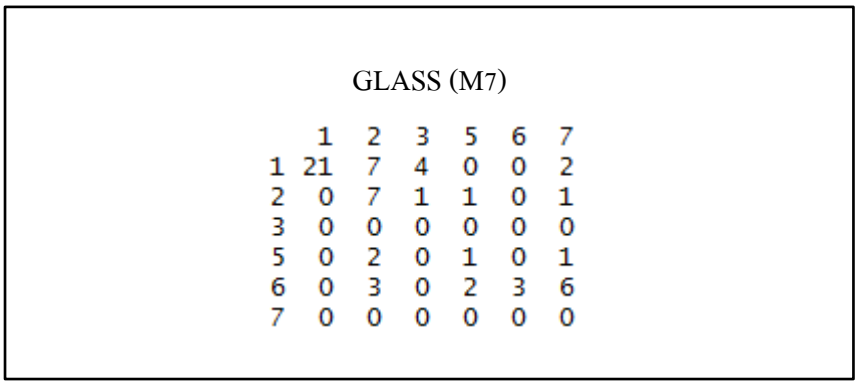
	1	2	3	5	6	7
1	21	14	5	1	0	3
2	0	0	0	0	0	0
3	0	0	0	0	0	0
5	0	3	0	3	0	2
6	0	2	0	0	2	0
7	0	0	0	0	1	5

รูปที่ 4.18 แสดงผลการทดสอบโมเดล M6 จาก Process6

ผลลัพธ์จาก Process7 นำข้อมูลฝึกสอนมาผ่านกระบวนการคัดเลือกคอลัมน์ด้วยค่าสหสัมพันธ์และนำเฉพาะข้อมูลที่เป็นข้อมูลปกติ แล้วทำการเพิ่มจำนวนข้อมูลในคลาสด้วยวิธี Over-Sampling กำหนดให้ข้อมูลที่ผ่านกระบวนการดังกล่าวชื่อว่าชุดข้อมูล D7 แสดงดังในตารางที่ 4.14 เนื่องจากเมื่อใช้ข้อมูล SECOM ในกระบวนการดังกล่าวพบว่าไม่มีข้อมูลแถวใดเลยที่ไม่เป็นข้อมูลที่ผิดปกติจากทุก ๆ คอลัมน์ หลังจากนั้นทำการสร้างโมเดลจากชุดข้อมูล D6 และทำการทดสอบด้วยข้อมูลทดสอบแสดงดังรูปที่ 4.19

ตารางที่ 4.14 แสดงรายละเอียดข้อมูลฝึกสอนที่ผ่าน Process7 ของข้อมูล GLASS

Data Set	Instances	Attributes	Class					
			1	2	3	5	6	7
D7	193	6	39	41	33	40	40	0


GLASS (M7)

	1	2	3	5	6	7
1	21	7	4	0	0	2
2	0	7	1	1	0	1
3	0	0	0	0	0	0
5	0	2	0	1	0	1
6	0	3	0	2	3	6
7	0	0	0	0	0	0

รูปที่ 4.19 แสดงผลการทดสอบโมเดล M7 จาก Process7

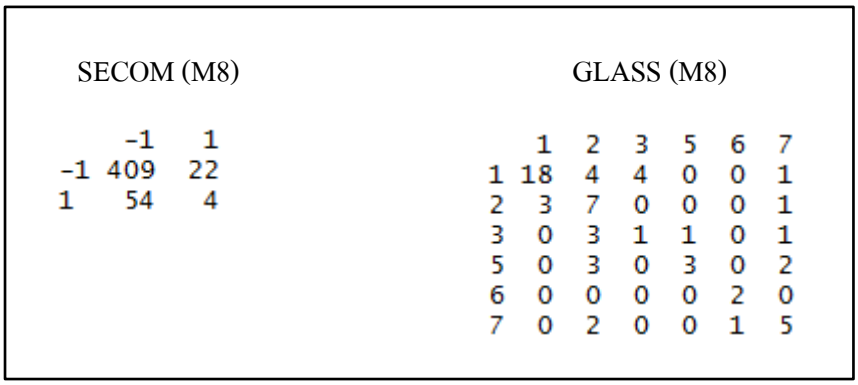
ผลลัพธ์จาก Process8 นำข้อมูลฝึกสอนมาทำการเพิ่มจำนวนข้อมูลในคลาสด้วยวิธี Over-Sampling โดยกำหนดให้เป็นชุดข้อมูล D8 แสดงในตารางที่ 4.15 และ 4.16 หลังจากนั้นทำการสร้างโมเดลจากชุดข้อมูล D8 และทำการทดสอบด้วยข้อมูลทดสอบแสดงดังรูปที่ 4.20

ตารางที่ 4.15 แสดงรายละเอียดข้อมูลฝึกสอนที่ผ่าน Process8 ของข้อมูล SECOM

Data Set	Instances	Attributes	Class	
			-1	1
D8	1936	591	1000	936

ตารางที่ 4.16 แสดงรายละเอียดข้อมูลฝึกสอนที่ผ่าน Process8 ของข้อมูล GLASS

Data Set	Instances	Attributes	Class					
			1	2	3	5	6	7
D8	319	10	46	57	48	54	54	57

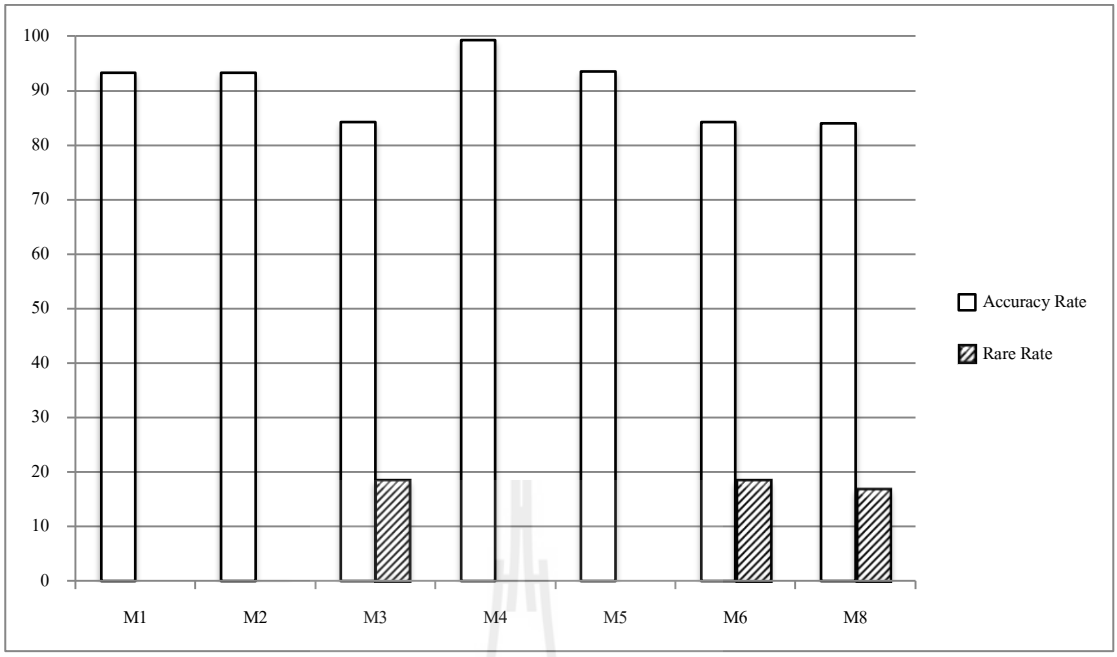


รูปที่ 4.20 แสดงผลการทดสอบโมเดล M8 จาก Process8

เมื่อทำการทดลองซ้ำ 10 ครั้งและหาค่าเฉลี่ยของค่าความถูกต้องของโมเดลโดยแบ่งออกเป็น 2 แบบคือค่าความถูกต้องโดยรวม และค่าความถูกต้องของคลาสที่ค้นพบได้ยาก โดยในโมเดล M5 ที่มีการแบ่งข้อมูลเป็นกลุ่มย่อยแล้วหาโมเดลในแต่ละกลุ่มย่อยจะใช้ค่าโดยเฉลี่ยของทุกกลุ่มย่อย สามารถแสดงได้ดังตารางที่ 4.17 ซึ่งเป็นผลการทดสอบของข้อมูล SECOM ซึ่งไม่สามารถสร้างโมเดล M7 ได้ และ 4.18 เป็นผลการทดสอบของข้อมูล GLASS และแสดงในรูปแบบกราฟได้ดังรูปที่ 4.21 และ 4.22

ตารางที่ 4.17 แสดงผลเฉลี่ยการทดสอบของข้อมูล SECOM

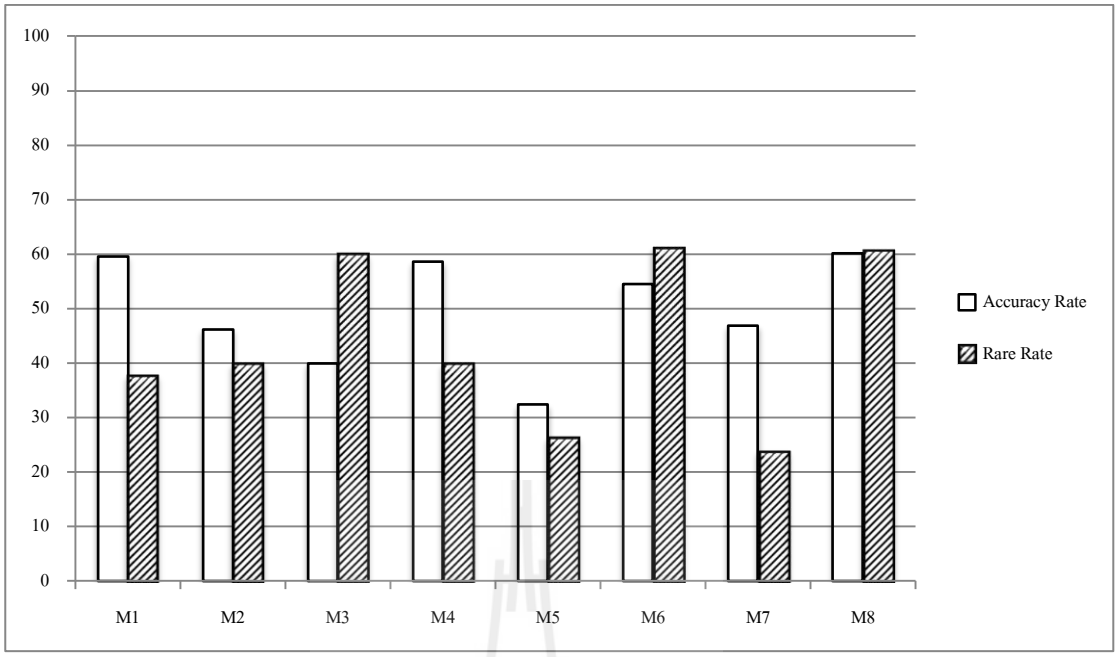
Model	Accuracy Rate	Rare Rate
M1	93.3	0.0
M2	93.3	0.003
M3	84.2	18.4
M4	99.3	0.003
M5	93.5	0.0
M6	84.2	18.4
M8	84.0	16.8



รูปที่ 4.21 แสดงผลเฉลี่ยการทดสอบของข้อมูล SECOM

ตารางที่ 4.18 แสดงผลเฉลี่ยการทดสอบของข้อมูล GLASS

Model	Accuracy Rate	Rare Rate
M1	59.6	37.7
M2	46.2	39.9
M3	40.0	59.9
M4	58.6	39.9
M5	32.4	26.4
M6	54.5	61.0
M7	46.9	23.9
M8	60.1	60.5



รูปที่ 4.22 แสดงผลเฉลี่ยการทดสอบของข้อมูล GLASS

4.4 อภิปรายผล

จากผลการทดสอบตามที่ได้ออกแบบการทดสอบโมเดลด้วยวิธีการต่าง ๆ และได้ใช้ข้อมูล SECOM และข้อมูล GLASS ในการทดสอบโมเดล โดยในการวิจัยนี้มุ่งเน้นในการค้นพบรูปแบบที่สามารถจำแนกข้อมูลที่ค้นพบได้ยากสามารถสรุปได้ดังนี้ จากรูปที่ 4.21 แสดงผลการทดสอบของข้อมูล SECOM นั้นสรุปได้ว่าโมเดลที่มีความสามารถในการจำแนกข้อมูลที่ค้นพบได้ยากดีที่สุดเรียงตามลำดับคือโมเดล M3, M6, M8 ซึ่งโมเดลที่ 3 และโมเดลที่ 6 มีความสามารถในการจำแนกข้อมูลที่ค้นพบได้ยากเท่ากัน และมีค่าความถูกต้องของของโมเดลเท่ากันอีกด้วย เมื่อเปลี่ยนชุดข้อมูลที่ทำการทดสอบด้วยชุดข้อมูล GLASS นั้นสรุปได้ว่าโมเดลที่มีความสามารถในการจำแนกข้อมูลที่ค้นพบได้ยากดีที่สุดเรียงตามลำดับคือโมเดล M6, M8, M3 จึงสามารถสรุปได้ว่าโมเดลที่มีความสามารถในการจำแนกข้อมูลที่ค้นพบได้ยากดีที่สุดคือโมเดลที่ 6 ซึ่งสร้างได้จากการนำข้อมูลทั้งหมดผ่านการคัดเลือกคอลัมน์ด้วยค่าสหสัมพันธ์และทำการเพิ่มจำนวนข้อมูลในคลาสกลุ่มเล็กด้วยวิธีการสุ่มเกิน

บทที่ 5
สรุปผลการวิจัยและข้อเสนอแนะ

ในการทำเหมืองข้อมูลในปัจจุบันส่วนใหญ่จะทำเหมืองข้อมูลกับชุดข้อมูลที่มีขนาดใหญ่ ซึ่งการค้นหารูปแบบเพื่อการจำแนกข้อมูลที่ค้นพบได้ยากในชุดข้อมูลขนาดใหญ่ นั้นสามารถทำได้ยากเนื่องจากคุณสมบัติส่วนมากของข้อมูลที่ค้นพบได้ยากมีคุณสมบัติคล้ายกับคุณสมบัติของข้อมูลส่วนใหญ่ จึงทำให้การสร้างรูปแบบเพื่อจำแนกข้อมูลที่ค้นพบได้ยากออกจากข้อมูลส่วนใหญ่สามารถทำได้ลำบาก หรืออาจจะทำให้ไม่สามารถหารูปแบบของข้อมูลที่ค้นพบได้ยาก

ในงานวิจัยนี้มีจุดประสงค์คือออกแบบอัลกอริทึมเพื่อพัฒนากระบวนการค้นหารูปแบบข้อมูลที่ค้นพบได้ยากเพื่อการจำแนกข้อมูล และพัฒนาโปรแกรมเพื่อให้ผู้ใช้สามารถใช้งานได้สะดวก ปัจจัยที่ทำให้การค้นหารูปแบบข้อมูลของข้อมูลที่ค้นพบได้ยากทำได้ลำบากหรือไม่สามารถทำได้นั้นเกิดขึ้นจากจำนวนของข้อมูลที่ค้นพบได้ยากมีจำนวนน้อยมากในชุดข้อมูลจึงทำให้การค้นหารูปแบบของข้อมูลทำได้ยาก จึงได้ใช้เทคนิคการสุ่มเกินในการเพิ่มจำนวนข้อมูลที่ค้นพบได้ยากเพื่อให้สามารถค้นหารูปแบบของข้อมูลได้

เมื่อทำการทดสอบตามที่ได้ออกแบบการทดลองไว้นั้นพบว่าโมเดลที่ให้ค่าความถูกต้องโดยรวมและค่าความถูกต้องของการจำแนกคลาสที่ค้นพบได้ยากนั้นมี 3 โมเดลที่มีความถูกต้องใกล้เคียงกันได้แก่โมเดล M3, M6 และ M8 ซึ่งแต่ละโมเดลมีกระบวนการดังต่อไปนี้

โมเดล M3 ทำการคัดเลือกข้อมูลจาก TrainData เฉพาะข้อมูลที่เป็นข้อมูลที่ผิดปกติ (Outlier) เท่านั้นแล้วจึงนำข้อมูลดังกล่าวมาผ่านกระบวนการคัดเลือกคอลัมน์ด้วยการวิเคราะห์ค่าสหสัมพันธ์ หลังจากนั้นนำข้อมูลที่ได้คัดเลือกคอลัมน์แล้วมาเข้าสู่กระบวนการสุ่มเกินเพื่อเพิ่มจำนวนข้อมูลในคลาสขนาดเล็ก แล้วจึงทำการสร้างโมเดลจากข้อมูลที่ได้ผ่านการสุ่มเกิน เมื่อได้โมเดลเรียบร้อยแล้วทำการทดสอบด้วยข้อมูล TestData

โมเดล M6 นำข้อมูล TrainData ทั้งหมดมาผ่านกระบวนการคัดเลือกคอลัมน์ด้วยการวิเคราะห์ค่าสหสัมพันธ์หลังจากนั้นนำข้อมูลที่ได้มาผ่านกระบวนการสุ่มเกิน และนำข้อมูลที่ได้จากกระบวนการสุ่มเกินไปทำการสร้างโมเดล หลังจากที่ได้โมเดลเรียบร้อยแล้วนำโมเดลที่ได้ไปทดสอบด้วยข้อมูล TestData (โมเดล M6 ใช้ข้อมูลทุกเรกคอร์ดทั้งที่เป็น Outlier และไม่เป็น Outlier)

โมเดล M8 นำข้อมูล TrainData ทั้งหมดมาผ่านกระบวนการสุ่มเกินเพียงเทคนิคเดียวเท่านั้น โดยไม่มีการคัดเลือกคอลัมน์และไม่พิจารณาเกี่ยวกับ Outlier และนำข้อมูลที่ได้จากกระบวนการสุ่มเกินไปทำการสร้างโมเดล หลังจากที่ได้โมเดลเรียบร้อยแล้วนำโมเดลไปทำการทดสอบด้วยข้อมูล TestData

5.1 สรุปผลการวิจัย

จากผลการทดสอบค่าความถูกต้องโดยรวมและค่าความถูกต้องของการจำแนกคลาสที่ค้นพบได้ยาก สามารถสรุปได้ว่าโมเดลที่สามารถจำแนกคลาสที่ค้นพบได้ยากที่แม่นยำที่สุดคือโมเดล M6 ซึ่งใช้กระบวนการคัดเลือกคอลัมน์ด้วยการวิเคราะห์ค่าสหสัมพันธ์และทำการเพิ่มจำนวนข้อมูลในคลาสที่เป็นคลาสขนาดเล็กมากด้วยวิธีเพิ่มแบบเท่าตัว ซึ่งในชุดข้อมูล SECOM โมเดล M6 จะให้ผลลัพธ์เท่ากับโมเดล M3 ซึ่งใช้กระบวนการเช่นเดียวกันกับ M6 แต่จะใช้ข้อมูลที่เป็นค่าที่ผิดปกติเท่านั้น จากการพิสูจน์แล้วพบว่าข้อมูล SECOM เป็นข้อมูลที่มีความผิดปกติทุกแถว เนื่องจากเป็นข้อมูลที่มีจำนวนคอลัมน์เป็นจำนวนมากถึง 591 คอลัมน์ ดังนั้นค่าความถูกต้องโดยรวมและค่าความถูกต้องของคลาสที่ค้นพบได้ยากของข้อมูล SECOM จึงมีค่าที่เท่ากัน แต่เมื่อเปลี่ยนชุดข้อมูลโดยใช้ชุดข้อมูล GLASS แล้วทำให้เห็นชัดขึ้นว่าโมเดล M6 นั้นมีความสามารถในการจำแนกคลาสที่ค้นพบได้ยากได้ดีกว่าโมเดล M3 และยังมีค่าความถูกต้องโดยรวมมากกว่าโมเดล M3 อย่างชัดเจน ส่วนโมเดล M8 ที่มีค่าความถูกต้องของการจำแนกคลาสที่ค้นพบได้ยากใกล้เคียงกับโมเดล M6 และมีค่าความถูกต้องโดยรวมมากกว่าโมเดล M6 นั้นใช้กระบวนการเพิ่มจำนวนข้อมูลในคลาสด้วยวิธีเพิ่มแบบเท่าตัวอย่างเดียวเท่านั้น ผู้ทำการวิจัยมีความเห็นว่าถ้าหากข้อมูลที่ใช้ในการทดสอบมีจำนวนคอลัมน์ไม่มาก และจำนวนแถวไม่มากควรเลือกใช้โมเดล M8 เนื่องจากโมเดลที่ได้มีความถูกต้องของคลาสที่ค้นพบได้ยากที่ใกล้เคียงกันมาก และยังมีค่าความถูกต้องโดยรวมที่ดีกว่า แต่ถ้าหากชุดข้อมูลมีจำนวนคอลัมน์และจำนวนแถวมาก ๆ แล้วควรใช้โมเดล M6 เนื่องจากจะมีการคัดเลือกคอลัมน์ที่ไม่จำเป็นออกทำให้โมเดลที่ได้เมื่อทำการทดสอบกับข้อมูลจำนวนมากแล้วจะสามารถทำงานได้เร็วกว่าโมเดล M8 ซึ่งมีค่าความถูกต้องใกล้เคียงกัน

ส่วนผลการทดสอบของโมเดล M3 เมื่อทำการทดสอบกับข้อมูล GLASS พบว่าค่าความถูกต้องโดยรวมของโมเดลด้อยกว่าโมเดล M6 และ M8 อาจจะเกิดจากข้อมูล GLASS นั้นมีจำนวนคอลัมน์ที่น้อยและมีข้อมูลที่ผิดปกติจำนวนน้อยจึงทำให้โมเดล M3 มีประสิทธิภาพที่ลดลง ดังนั้นโมเดล M3 อาจจะเหมาะกับข้อมูลที่มีข้อมูลที่ผิดปกติจำนวนมาก

5.2 ปัญหาและข้อเสนอแนะ

ในการค้นหารูปแบบของคลาสที่ค้นพบได้ยากเป็นปัญหาที่ไม่สามารถแก้ไขได้ด้วยวิธีการเดียว หรือเทคนิคเพียงเทคนิคเดียวและเมื่อต้องการเพิ่มความสามารถในการจำแนกคลาสที่ค้นพบได้ยากจะทำให้ค่าความถูกต้องโดยรวมของโมเดลลดลงเนื่องจากคลาสที่ค้นพบได้ยากมีคุณสมบัติคล้ายกับคุณสมบัติของคลาสอื่น ๆ ที่มีจำนวนมากทำให้การเพิ่มความสามารถในการจำแนกคลาสที่ค้นพบได้ยาก และการคงค่าความถูกต้องโดยรวมของโมเดลให้มีความถูกต้องในระดับสูงนั้นทำได้ยาก การประยุกต์ใช้เทคนิคต่าง ๆ จึงต้องเลือกใช้ตามลักษณะของข้อมูล

ในอนาคตควรปรับปรุงแนวทางที่เสนอในวิทยานิพนธ์ฉบับนี้ทำให้การค้นหารูปแบบของคลาสที่ค้นพบได้ยากสามารถทำงานกับข้อมูลทุกรูปแบบเนื่องจากในงานวิจัยนี้สามารถทำได้กับข้อมูลชนิดตัวเลขเท่านั้น ซึ่งไม่สามารถทำงานกับข้อมูลเชิงกลุ่มได้

รายการอ้างอิง

H. Alhammady, and K. Ramamohanarao, (2004). **The Application of Emerging Patterns for Improving the Quality of Rareclass Classification**. In Proceedings of PKDD, pp. 207-211.

H. Alhammady, and K. Ramamohanarao. (2004). **Using Emerging Patterns and Decision Trees in Rareclass Classification.** In Proceedings of the Fourth IEEE International Conference on Data Mining (ICDM '04), pp.315-318.

N.V. Chawla. (2005). **Data Mining for Imbalanced Datasets: An Overview.** The Data Mining and Knowledge Discovery Handbook, p853-867.

S. Han, B. Yuan, and W. Liu, (2009). **Rare Class Mining: Progress and Prospect.** In Proceedings of Chinese Conference on Pattern Recognition, pp. 1-5.

J. Han, M. Kamber, (2006). **Data mining : concepts and techniques.** Amsterdam ; Boston : Elsevier.

H. He, and A. Ghodsi. (2010). **Rare Class Classification by Support Vector Machine.** In Proceedings 20th International Conference on Pattern Recognition, pp.548-551.

K. McCarthy, B. Zabar, and G. Weiss (2005). **Does Cost-sensitive Learning Beat Sampling for Classifying Rare Classes?** In Proceedings of the 1st International Workshop on Utilitybased Data Mining, pp.69–77.

C. Molnar. (2012). **Conditional Trees or Unbiased Recursive Partitioning A Conditional Inference framework [Online]**.
Available URL: http://www.slideshare.net/christophmolnar/conditional-trees

C. Seiffert, T. M. Khoshgoftaar, and J. Van Hulse, (2009). **Improving Software-Quality Predictions with Data Sampling and Boosting.** IEEE Transactions on Systems, Man, Cybernetics – Part A: Systems and Humans, vol. 39, no. 6, pp. 1283-1294.

K. Thearling (2012). **An Introduction to Data Mining**. Retrieved November 3, 2012, from http://www.thearling.com/text/dmwhite/dmwhite.htm

G. Weiss. (2004). **Mining With Rarity: A Unifying Framework.** ACM-SIGKDD Explorations Vol. 6 Issue 1, p7-19.

J. Wu, H. Xiong, P. Wu, and J. Chen (2007)**. Local Decomposition for Rare Class Analysis.** In Proceedings of International Conference on Knowledge Discovery and Data Mining (KDD), pp.814–823.

ภาคผนวก ก

บทความวิจัยที่ได้รับการตีพิมพ์เผยแพร่

รายชื่อบทความวิจัยที่ได้รับการตีพิมพ์เผยแพร่

K. Chomboon, K. Kerdprasop, and N. Kerdprasop. (2013). **Rare Class Discovery Techniques for Highly Imbalanced Data.** In Proceedings of International MultiConference of Engineers and Computer Scientists 2013, pp. 269-272.

Rare Class Discovery Techniques for Highly Imbalanced Data

Kittipong Chomboon*, Kittisak Kerdprasop, and Nittaya Kerdprasop

***Abstract*—This research aims at proposing a method to induce a classification model of minority data cases that are always predominant by a much larger majority cases. Our proposed method applies feature selection technique to choose 25% of attributes that show highly correlation between classes. Then use over-sampling technique on minority cases before making a classification. We have found from the experimental results that data of rare cases, which is normally disappeared, can be detected through our proposed method. In this research, we use R language for implementing our proposed method and other four discovery techniques.**

***Index Terms*—Data Mining, Rare Class, Imbalanced Data, Data Classification, R Language.**

I. Introduction

Present computer technology has progressed at a very rapid speed and it has tremendous effect to the storage of electronic data. With enormously increasing data, conventional method to analyze data is inappropriate. Data mining is thus an essential technology and has been proven useful for automatic analysis of large data [5], [7]. Data mining is the discovery of interesting patterns from large data. Discovered patterns can be in various forms such as a tree model for classification, a set of rules for association, or a group of representative centroids for data clustering. In this paper, we focus on the discovery of a tree model. This model can be used to predict events in the future.

Decision tree induction is normally a powerful technique to discover a tree model for future event prediction. But for a highly imbalanced data in which the number of data instances in one class is extremely smaller than the number of instances in other classes. Dataset of a smaller group is called a rare class [2], [6] or a minority class. A dataset with high imbalance between majority and minority classes can cause much trouble to the tree induction algorithm. During the tree building process data instances from a minority class are normally pruned and disappear from the final tree model. This makes the tree model predict the majority class correctly, but minority class tends to be incorrectly predicted. This is a challenging problem known as a rare class mining.

Manuscript received December 5, 2012; revised January 10, 2013. This work was supported in part by grant from Suranaree University of Technology through the funding of Data Engineering Research Unit.

K. Chomboon is a master student with the School of Computer Engineering, Suranaree University of Technology, Nakhon Ratchasima 30000, Thailand (e-mail: chomboon.k@gmail.com).

K. Kerdprasop is an associate professor with the School of Computer Engineering, Suranaree University of Technology, Nakhon Ratchasima 30000, Thailand.

N. Kerdprasop is an associate professor with the School of Computer Engineering, Suranaree University of Technology, Nakhon Ratchasima 30000, Thailand.

In this research, we comparatively study five data preparation techniques to help decision tree induction algorithm creating a model that can predict rare class data. The various data preparation techniques include clustering, over-sampling, correlation analysis, outlier detection, and a mixture of correlation and outlier analyses. The experimentation has been performed on the RStudio environment and the program is implemented with the R language. Source code of our implementation is available in the Appendix.

II. Rare class classification

Rare class classification is the data mining task aiming at building a model that can correctly classify both the majority and minority classes. Classifying minority or rare class is difficult because size of the rare class is too small. Many researchers have tried to solve this problem. Alhammady and Ramanohanarao [1] proposed an algorithm called EPDT (Emerging Pattern Decision Tree) to train a decision tree that can classify rare class.

He and Ghodsi [3] proposed a classification algorithm based on the SVM (Support Vector Machine) for classifying rare objects. Wu et al. [8] proposed a technique called COG (Classification using lOcal clusterinG) that also based on the SVM classifier. McCarthy et al. [4] proposed the algorithm called Cost-Sensitive and compared the algorithm with the under-sampling and over-sampling techniques.

Over-sampling and clustering techniques as applied in several work seem to give a good result. We then perform a comparative study by applying these techniques together with the correlation and outlier detection methods. These techniques are applied in the data preparation step, whereas the tree induction is our classification method. The research methodology and the experimental results are explained in the following sections.

III. Methodology

In this section we present the discovery process of rare class on imbalanced dataset. We use SECOM dataset from the UCI (http://archive.ics.uci.edu/ml/datasets/SECOM) SECOM is data from a semi-conductor manufacturing process. It is highly-skewed. The dataset contains 1567 data instances taken from a wafer fabrication production line. Each data instance contains 590 sensor measurements. The data are labeled as -1 (means pass the test), or 1 (means fail the test). There are only 104 fail cases, whereas the pass

case are 1463. This is a 1:14 proportion. The goal of SECOM dataset is a good or bad semi-conductor from manufacturing process. The overview our techniques show as Fig. 1.

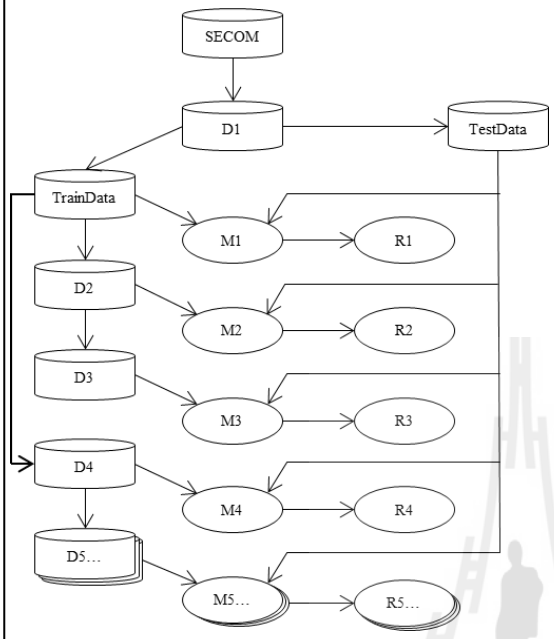


Fig. 1 The overview of five techniques to discover rare class

Then we will describe these techniques step by step. Step one: we manage missing values in SECOM dataset by replacing with median value and create a new dataset called D1, graphically show in Fig. 2.

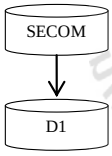


Fig. 2 Step 1: manage missing value

Step 2: we split D1 into train and test data. TrainData and TestData has ratio 70:30. Then create model from train data and evaluate accuracy with TestData, shown in Fig. 3.

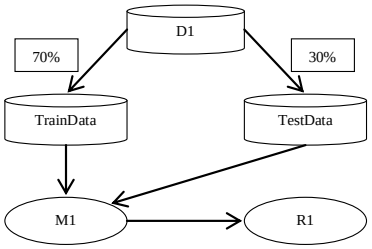


Fig. 3 Step 2: evaluate accuracy of Traindata.

Step 3: find top 25% of correlation between the class attribute and other attributes of TrainData then union with outliers. Outliers are found from each attribute and include all outliers into the TrainData. Create new dataset called D2 with correlated feature selection and outliers inclusion. Then create model form D2 and examine accuracy with TestData, show as Fig. 4.

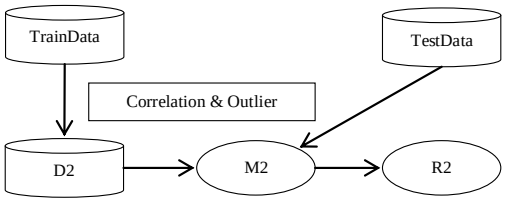


Fig. 4 Step 3: feature selection by correlation analysis and inclusion of outliers

Step 4: apply over-sampling technique on the minority class (that is, the fail cases) D2 dataset. The new dataset is called D3. Then create model from D3 and evaluate accuracy with TestData, shown in Fig. 5.

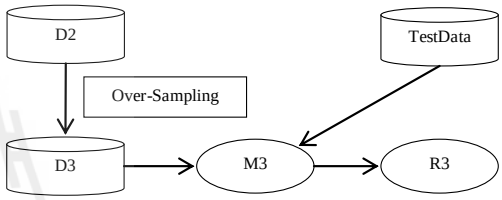


Fig. 5 Step 4: oversampling the fail cases

Step 5: find top 25% of correlation between the class attribute and other attributes of TrainData, then create new dataset with selected features and call this dataset D4. After that create a model from D4 and evaluate its accuracy with TestData, as shown in Fig. 6.

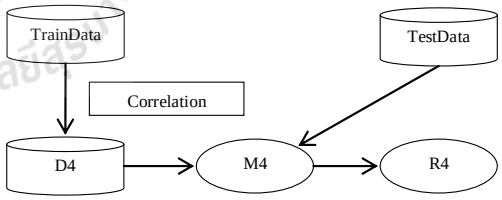


Fig. 6 Step 5: feature selection with correlation analysis

Step 6: clustering dataset D4 with the pamk function available in the RStudio program. Function pamk returns a group of two clusters. Then create model of each cluster and evaluate accuracy of each model with TestData, shown in Fig. 7.

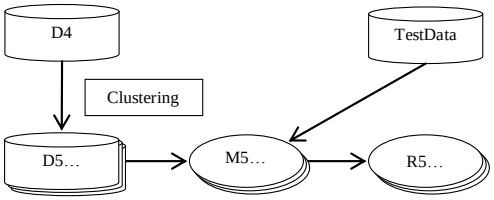


Fig. 7 Step 6: cluster data then create model from each cluster

After design techniques, we implemented by using decision tree algorithm for classification with R language by RStudio. RStudio is an open source program and easy to use as there are several statistical and data mining functions available in the library.

I. Experimental Results

This research experimentation used SECOM (semi-conductor manufacturing process) dataset from UCI Machine Learning Repository. SECOM dataset has 591 attributes and 1567 data instances.

For rare class discovery technique comparison, we used decision tree algorithm for classification on imbalanced dataset and used R language coding on RStudio program. The classification models of the designed techniques are shown in Fig. 8. Evaluation results of the 5 models (M1, M2, M3, M4, M5) can be displayed as a confusion matrix as shown in Fig. 9.

```
$M1
    1) A104 <= -0.0087; criterion = 1, statistic = 25.017
      2) A34 <= 9.1757; criterion = 0.993, statistic = 19.267
        3) A474 <= 116.5288; criterion = 0.997, statistic = 20.873
          4)*  weights = 657
        3) A474 > 116.5288
          5)*  weights = 9
      2) A34 > 9.1757
        6)*  weights = 110
    1) A104 > -0.0087
      7)*  weights = 323

$M2
    1) A60 <= 8.0636; criterion = 0.997, statistic = 18.452
      2)*  weights = 938
    1) A60 > 8.0636
      3)*  weights = 161

$M3
    1) A130 <= -2.128; criterion = 1, statistic = 124.127
      2) A121 <= 5.8821; criterion = 1, statistic = 87.221
        3)*  weights = 9
      2) A121 > 5.8821
        4)*  weights = 158
    1) A130 > -2.128
      5) A317 <= 7.1479; criterion = 1, statistic = 85.43
        6) A282 <= 0.0153; criterion = 1, statistic = 72.096
          7) A116 <= 779.3134; criterion = 1, statistic = 34.839
            8) A96 <= 0; criterion = 1, statistic = 29.867
              9) A105 <= 0; criterion = 1, statistic = 32.245
                                  ⋮

$M4
    1) A60 <= 8.0636; criterion = 0.997, statistic = 18.452
      2)*  weights = 938
    1) A60 > 8.0636
      3)*  weights = 161

$M5A
    1) A60 <= 8.0636; criterion = 0.999, statistic = 20.059
      2)*  weights = 710
    1) A60 > 8.0636
      3)*  weights = 144

$M5B
    1)*  weights = 245
```

Fig. 8 Classification models of the designed techniques

```
$M1                    $M4

     -1   1                 -1   1
-1 438  30             -1 438  30
 1   0   0              1   0   0

$M2                    $M5A

     -1   1                 -1   1
-1 438  30             -1 438  30
 1   0   0              1   0   0

$M3                    $M5B

     -1   1                 -1   1
-1 392  20             -1 438  30
 1  46  10              1   0   0
```

Fig. 9 Results of classification model evaluation

A comparative performance of each model can be summarized and shown in Table 1. It can be seen from the result that model M3 that has been built from the top 25% correlated attributes and data in rare class have been over-sampling shows the best rare class detection rate of 10 from the total 30 rare data instances. Other techniques have zero rare class detection rate.

TABLE 1
Comparative Results of Designed Techniques

Model	Accuracy	Rare class, detection rate	Recall	Precision
M1	0.93	0	0.93	1
M2	0.93	0	0.93	1
M3	0.85	0.33	0.95	0.89
M4	0.93	0	0.93	1
M5	0.93	0	0.93	1
M6	0.93	0	0.93	1

II. Conclusion

This research aims to study rare class discovery techniques for highly imbalance data using decision tree algorithm as a classification method. The results show that the best model is model 3 that used correlation-based feature selection, outlier and over-sampling techniques to prepare data for classification. Model 3 has rare class detection rate at 33%, whereas other models cannot detect rare class.

References

[1] H. Alhammady, and K. Ramamohanarao. (2004). "Using Emerging Patterns and Decision Trees in Rareclass Classification," *In Proceedings of the Fourth IEEE International Conference on Data Mining (ICDM '04)*, pp.315-318.

[2] N.V. Chawla. (2005). "Data Mining for Imbalanced Datasets: An Overview," *The Data Mining and Knowledge Discovery Handbook*, pp.853-867.

[3] H. He, and A. Ghodsi. (2010). "Rare Class Classification by Support Vector Machine," *In Proceedings of the 20th International Conference on Pattern Recognition*, pp.548-551.

[4] K. McCarthy, B. Zabar, and G. Weiss (2005). "Does Cost-sensitive Learning Beat Sampling for Classifying Rare Classes?" *In Proceedings of the 1st International Workshop on Utilitybased Data Mining*, pp.69–77.

[5] K. Thearling (2012). "An Introduction to Data Mining," Retrieved November 3, 2012, from http://www.thearling.com/text/dmwhite/dmwhite.htm

[1] G. Weiss. (2004). "Mining With Rarity: A Unifying Framework," *ACM-SIGKDD Explorations Vol. 6 Issue 1*, pp.7-19.
[2] Wikipedia, the free encyclopedia (2012). "Data Mining [Online]," Available URL: http://en.wikipedia.org/wiki/Data_mining
[3] J. Wu, H. Xiong, P. Wu, and J. Chen (2007). "Local Decomposition for Rare Class Analysis," *In Proceedings of International Conference on Knowledge Discovery and Data Mining (KDD)*, pp.814–823.

Appendix

Source code of R language for comparing rare class discovery techniques is given below. Start the program by calling "my.fn" function.

```
my.fn <- function(data.f, seed=0){
  library("party")
  library("DMwR")
  library("fpc")
  ## initial variable
  stats <- list()
  model <- list()
  result <- list()
  outlier <- vector()
  cluster <- list()
  n.col <- ncol(data.f)
  decition.name <- names(secom)[n.col]
  my.formula <- formula(paste(decition.name, "~."))

  ## initial data
  data.f[,n.col] <- factor(data.f[,n.col])
  for(i in 1:(n.col-1)){
    data.f[is.na(data.f[,i]),i] <- median(data.f[,i], na.rm=T)
  }

  ## split data
  if(seed == 0){
    seed <- sample(1000:9999,1)
  }
  set.seed(seed)
  ind <- sample(2, nrow(data.f), replace=T, prob=(c(0.7,0.3)))
  data.train <- data.f[ind==1,]
  data.test <- data.f[ind==2,]
  print(paste("$$$$ SetSeed ==",seed,"$$$$"))
  print("Split Data Complete")

  ## test for M1 model
  model$M1 <- ctree(my.formula, data=data.train)
  result$M1 <- table(predict(model$M1, newdata=data.test),
data.test[,n.col])
  print("Model M1 Complete")

  ## find outlier by boxplot.stats()
  for(i in 1:(n.col-1)){
    outlier <- union(outlier,which(data.train[,i] %in%
boxplot.stats(data.train[,i])$out))
  }
  outlier <- sort(outlier)

  ## find corelation by cor and select 25%
  cor.sample <- sample(2, ncol(data.train), replace=T,
prob=(c(0.25,0.75)))
  cor.n <- length(cor.sample[cor.sample==1])
  cor.tmp <- cor(data.train[,-n.col],
as.numeric(levels(data.train[,n.col]))[as.numeric(data.train[,n.col])]
)
  cor.name <- dimnames(cor.tmp)[1][[1]][sort(cor.tmp,
decreasing=T, index.return=T)$ix[1:cor.n]]

  ## create data1 from outlier and corelation then create
model&test to M2
  data1 <- data.train[outlier, c(cor.name,names(data.train)[n.col])]
  model$M2 <- ctree(my.formula, data=data1)
  result$M2 <- table(predict(model$M2, newdata=data.test),
data.test[,n.col])
  print("Model M2 Complete")

  ## create data2 form data1 by decition(A591) -1 == 1 then
create model&test to M3
  data2 <- data1
  data2.n <- nrow(data2)
  data2.bad.rownames <-
rownames(data2[data2[,decition.name]==1,])
  data2.bad.length <- length(data2.bad.rownames)
  data2.inc <- data2.n-(data2.bad.length*2)
  data2.sample <- sample(data2.bad.rownames, data2.inc,
replace=T)
  data2[(data2.n+1):(data2.n+data2.inc),] <- data2[data2.sample,]
  model$M3 <- ctree(my.formula, data=data2)
  result$M3 <- table(predict(model$M3, newdata=data.test),
data.test[,n.col])
  print("Model M3 Complete")

  ## create data3 form data1 by corelation then create model&test
to M4
  data3 <- data.train[,c(cor.name,names(data.train)[n.col])]
  model$M4 <- ctree(my.formula, data=data3)
  result$M4 <- table(predict(model$M4, newdata=data.test),
data.test[,n.col])
  print("Model M4 Complete")

  ## find k form data3
  train.pam <- pamk(data3[,-ncol(data3)])
  for(i in 1:train.pam$nc){
    cluster.name <- paste("c", i, sep="")
    model.name <- paste("M5", rawToChar(as.raw(i+64)),
sep="")
    cluster[[cluster.name]] <-
data3[train.pam$pamobject$clustering==i,]
    model[[model.name]] <- ctree(my.formula,
data=cluster[[cluster.name]])
    result[[model.name]] <- table(predict(model[[model.name]],
newdata=data.test), data.test[,n.col])
    print(paste("Model",paste("M5", rawToChar(as.raw(i+64)),
sep=""),"Complete"))
  }
  print("#### Complete ####")

  stats$seed <- seed
  stats$data <- list(data=data.f, data.train=data.train,
data.test=data.test, data1=data1, data2=data2, data3=data3)
  stats$cluster <- cluster

  stats$model <- model
  stats$result <- result
  stats

}
```

ประวัติผู้เขียน

นายกิตติพงศ์ ชมบุญ เกิดเมื่อวันที่ 6 กันยายน พ.ศ. 2532 ที่อำเภอบัวเชด จังหวัดสุรินทร์ เริ่มเข้าศึกษาระดับชั้นอนุบาล 1 ถึงชั้นประถมศึกษาปีที่ 5 ที่โรงเรียนบ้านสวาท อำเภอบัวเชด จังหวัดสุรินทร์ หลังจากนั้นได้ย้ายไปศึกษาต่อในระดับชั้นประถมศึกษาปีที่ 6 ที่โรงเรียนเมือง อำเภอเมือง จังหวัดสุรินทร์ จากนั้นได้เข้าศึกษาต่อในระดับมัธยมศึกษาตอนต้นและตอนปลาย ที่โรงเรียนสุร วิทยาคาร อำเภอเมือง จังหวัดสุรินทร์ ในปีการศึกษา 2551 ได้เข้าศึกษาต่อในระดับปริญญาตรีในสาขาวิชาวิศวกรรมคอมพิวเตอร์ ที่มหาวิทยาลัยเทคโนโลยีสุรนารี และสำเร็จการศึกษาในปีการศึกษา 2554 ภายหลังจากสำเร็จการศึกษาในระดับปริญญาตรีได้เข้าศึกษาระดับปริญญาโท สาขาวิชาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยเทคโนโลยีสุรนารีในปีการศึกษา 2555

ในระหว่างการศึกษาได้รับความอนุเคราะห์อย่างดีจากอาจารย์ที่ปรึกษาและอาจารย์ประจำวิชา Database Systems และได้รับความไว้วางใจให้เป็นผู้ช่วยสอนปฏิบัติการในรายวิชา Database Systems และ Knowledge Discovery and Data Mining หลังจากนั้นได้รับการตีพิมพ์เผยแพร่บทความวิจัยซึ่งรายละเอียดสามารถดูได้ที่ภาคผนวก ก