รหัสโครงการ SUT7-705-45-06-41

รายงานการวิจัย

อัลกอริทึมและเทคนิคที่เหมาะสมกับการสังเคราะห์โมเดลที่ช่วยวินิจฉัยโรคได้อัตโนมัติ

(A proper algorithm and technique for mining the medical diagnosis data sets)

ผู้วิจัย

หัวหน้าโครงการ

ผู้ช่วยศาสตราจารย์ ดร.นิตยา เกิดประสพ

สาขาวิชาวิศวกรรมคอมพิวเตอร์

สำนักวิชาวิศวกรรมศาสตร์

มหาวิทยาลัยเทคโนโลยีสุรนารี

ได้รับทุนอุดหนุนการวิจัยจากมหาวิทยาลัยเทคโนโลยีสุรนารี ปีงบประมาณ พ.ศ. 2545

ผลงานวิจัยเป็นความรับผิดชอบของหัวหน้าโครงการวิจัยแต่เพียงผู้เดียว

มกราคม 2547

กิตติกรรมประกาศ

ผู้วิจัยขอขอบคุณมหาวิทยาลัยเทคโนโลยีสุรนารีที่ได้จัดสรรงบประมาณ เพื่อสนับสนุนการวิจัยนี้ให้ในปีงบประมาณ 2545 และขอขอบคุณผู้ช่วยศาสตราจารย์ ดร. กิตติศักดิ์ เกิดประสพ ที่ช่วยตรวจสอบและแก้ไขต้นฉบับรายงานการวิจัย นายประธาน ทรายทอง นายสุรสิริ นพกัณฑ์ และนักศึกษาวิศวกรรมคอมพิวเตอร์รุ่นที่ 3, 4 และ 5 ที่มีโอกาสได้เรียนวิชา Data Mining และมีส่วนร่วมในการทดสอบอัลกอริทึมการทำเหมืองข้อมูลเพื่อยืนยันผลลัพธ์กับรายงานฉบับนี้ ขอขอบคุณนางสาวอุษารัตน์ แสนปากดี ที่มีส่วนร่วมในการเตรียมข้อมูล และนางสาวสุนิดา ศรีสุริยชัย ที่ช่วยจัดรูปแบบคู่มือการใช้งานโปรแกรม WEKA ให้สวยงามขึ้น

บทคัดย่อภาษาไทย

ในปัจจุบันมนุษย์มีความสามารถในการเก็บบันทึกข้อมูลไว้เป็นจำนวนมหาศาล แต่ปริมาณที่มากเกินไปนี้กลับเป็นอุปสรรคต่อการวิเคราะห์ ตีความ และนำความรู้ที่ได้มาใช้ประโยชน์ต่อการตัดสินใจ การทำเหมืองข้อมูลจึงเกิดขึ้นในฐานะของศาสตร์แห่งการวิเคราะห์ข้อมูลอัตโนมัติ งานวิเคราะห์ข้อมูลนี้เป็นได้หลายรูปแบบ เช่น การวิเคราะห์เพื่อจำแนกข้อมูลได้อัตโนมัติ เพื่อค้นหาความสัมพันธ์ภายในข้อมูล ไปจนถึงเพื่อการตรวจจับรูปแบบที่เบี่ยงเบนไปจากข้อมูลปกติ

งานวิจัยนี้เน้นที่การทำเหมืองข้อมูลประเภทการจำแนกข้อมูลอัตโนมัติ โดยเจาะจงที่กลุ่มข้อมูลการวินิจฉัยโรค โดยมุ่งหวังเพื่อเอื้อประโยชน์กับงานทางการแพทย์ จุดมุ่งหมายหลักของการวิจัย คือ ทดสอบอัลกอริทึมต่างๆในการทำเหมืองข้อมูล เพื่อค้นหาอัลกอริทึมที่เหมาะสมกับข้อมูลการวินิจฉัยโรค งานวิจัยนี้ยังได้ตรวจสอบเทคนิคการเรียนรู้หลายครั้ง เพื่อเพิ่มความแม่นตรงในการทำนายและจำแนกประเภทข้อมูล โดยเน้นการศึกษาที่สองเทคนิค คือ bagging และ boosting

ผลการทดสอบอัลกอริทึมพื้นฐาน 4 อัลกอริทึมกับข้อมูล 12 ชุด พบว่าอัลกอริทึมที่ใช้หลักการค้นไม้ตัดสินใจ ทำงานได้ดีกับข้อมูลประเภทข้อความและสัญลักษณ์ที่มีจำนวนคลาสเพียงสองคลาส เมื่อจำนวนแอททริบิวต์เพิ่มมากขึ้นอัลกอริทึมประเภทนี้จะมีประสิทธิภาพลดลงอย่างชัดเจน ในขณะที่อัลกอริทึมที่ใช้หลักการเบย์ส์ไม่ได้รับผลกระทบจากจำนวนแอททริบิวต์ หรือจากจำนวนคลาสแต่อย่างใด

เทคนิคการเรียนรู้หลายครั้งสามารถเพิ่มประสิทธิภาพการจำแนกข้อมูลได้ แต่มีข้อยกเว้นในกรณีที่ข้อมูลมีการกระจุกตัวในบางคลาสมากเกินไป ประสิทธิภาพการจำแนกจะไม่เพิ่มขึ้นไปจนถึงลดลงในบางครั้ง บทสรุปของงานวิจัยนี้ได้เสนอแนะโมเดลในการคัดเลือกอัลกอริทึมและเทคนิคที่เหมาะสมโดยจะต้องพิจารณาร่วมกับลักษณะของข้อมูล

บทคัดย่อภาษาอังกฤษ

We are flooded with a huge volume of data and information. The tremendous amount of data, collected and stored in large databases, has far exceeded the human ability to analyze and extract valuable information for the purpose of decision-making support. Data mining has thus emerged as a new technology that can intelligently transform the vast amount of data into useful information and knowledge. Data mining tasks can vary from classification, association, to deviation detection.

This research focuses on the classification data mining. We have investigated the performance of four basic classification algorithms on twelve data sets, all are taken from a specific domain of medical diagnosis. Our main objective is to discover the appropriate technique for classifier induction on the medical data sets. Mulitple learning techniques such as bagging and boosting have also been employed to study the improvement on inducing a more accurate and sensitive model.

On the single-learning approach, we have found that the decision-tree induction algorithm performs well on the binary-class nominal data sets. However, the performance of the tree-based classifiers significantly degraded on high-dimensional data sets. The naive Bayes algorithm, on the contrary, is not affected by neither the dimension nor the number of classes.

The multiple-learning approach can, in general, improve the accuracy of the classification model. Nevertheless, we have discovered that if the distribution of data in each class is highly non-uniform, the multiple-learning techniques cannot improve,or even lower, the classifier's accuracy. We conclude our experimentations with the proposed decision model to suggest users how to choose the algorithm most appropriate for their specific medical data set.

สารบัญ

สารบัญตาราง

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มาของปัญหาการวิจัย

การขุดค้นข้อมูล หรือ การทำเหมืองข้อมูล (data mining) เป็นเทคโนโลยีใหม่ของการประยุกต์ใช้ข้อมูลที่เก็บอยู่ในฐานข้อมูล ให้เกิดประโยชน์สูงสุดแก่หน่วยงานที่เป็นเจ้าของข้อมูล การประยุกต์ใช้ข้อมูลที่กล่าวถึงนี้มีได้หลายแนวทาง แต่โดยทั่วไปมักจะเป็นการสรุปภาพรวมของข้อมูลในฐานข้อมูล, การวิเคราะห์แนวโน้มการเปลี่ยนแปลงของข้อมูล หรือ การค้นหาความสัมพันธ์ที่ซ่อนอยู่ภายในกลุ่มของข้อมูล

การวิเคราะห์ข้อมูลด้วยโปรแกรมช่วยงาน เช่น SPSS, SAS นักวิเคราะห์ข้อมูลจะต้องเป็นผู้กำหนดว่าจะศึกษาลักษณะใดจากข้อมูล และจะใช้ข้อมูลส่วนใดบ้าง แต่ data mining จะกระทำขั้นตอนต่างๆเหล่านี้ให้โดยอัตโนมัติ โปรแกรม data mining มีความสามารถที่จะค้นหาแนวโน้ม รูปแบบร่วม หรือลักษณะอื่นๆที่น่าสนใจ โดยไม่ต้องพึ่งพาการสั่งงานทุกขั้นตอนจากนักวิเคราะห์ข้อมูล และอาจจะสามารถค้นพบลักษณะที่น่าสนใจจากข้อมูลซึ่งนักวิเคราะห์ข้อมูลไม่ได้คาดหมายมาก่อน

ระบบ data mining มีความแตกต่างจากระบบผู้เชี่ยวชาญ (expert system) ตรงที่ *ฐานความรู้ของ data mining ได้จากการสังเคราะห์ขึ้นจากข้อมูลโดยตรง* สามารถปรับปรุงฐานความรู้ของตัวเองได้อัตโนมัติตามข้อมูลใหม่ที่ได้รับเพิ่มขึ้น ซึ่งต่างจากระบบผู้เชี่ยวชาญที่ฐานความรู้ถูกป้อนเข้ามาในระบบโดยผู้ที่ทำหน้าที่สร้างฐานความรู้ และจะคงตัวอยู่เช่นนั้นตลอดการใช้งาน

ถึงแม้การประยุกต์ใช้ data mining จะมีได้หลากหลาย แต่สามารถจัดกลุ่มกว้างๆได้เป็นสองกลุ่ม คือ กลุ่มที่ใช้ data mining เพื่อการทำนาย และกลุ่มที่ใช้เพื่อการอธิบาย

การทำ data mining <u>เพื่อการทำนาย</u> เป็นการนำความรู้*ที่เรียนรู้มาจากข้อมูลที่มีอยู่เพื่อ*ประโยชน์ในการทำนายข้อมูลใหม่ที่จะเกิดขึ้นในอนาคต เช่น จากข้อมูลลูกค้าของแผนกสินเชื่อของธนาคารที่ได้มีการจัดลำดับชั้นของลูกค้าไว้แล้วว่าใครเป็นลูกค้าชั้นดี ใครเป็นลูกค้าในระดับปานกลาง และใครเป็นลูกค้าที่มักจะผิดนัดชำระหนี้ โปรแกรม data mining สามารถเรียนรู้จากข้อมูลเหล่านี้และค้นหาโมเดลที่สามารถใช้อธิบายลักษณะของลูกค้าชั้นดี ลูกค้าระดับปานกลาง และลูกค้าที่ไม่เป็นที่ต้องการ จากโมเดลที่ได้นี้สามารถนำไปใช้ทำนายลูกค้าใหม่ที่มาขอสินเชื่อได้ว่าเขาน่าจะเป็นลูกค้าประเภทใดและสมควรได้รับอนุมัติสินเชื่อหรือไม่

การทำ data mining <u>เพื่อการอธิบาย</u> เป็นการค้นหารูปแบบที่น่าสนใจจากกลุ่มข้อมูล รูปแบบนี้มักจะเป็นความสัมพันธ์ หรือลักษณะที่เชื่อมโยงกันของข้อมูล การทำ data mining แบบนี้ต่างจากแบบแรกตรงที่ผู้ใช้ไม่ได้กำหนดล่วงหน้าว่าจะให้โปรแกรม data mining ค้นหารูปแบบหรือโมเดลของอะไร แต่ให้ค้นหาทุกรูปแบบที่น่าสนใจจากข้อมูล

จะเห็นได้ว่างาน data mining มีได้หลากหลายประเภท ทั้งนี้เนื่องจากความรู้ที่ต้องการได้จากข้อมูลมีได้หลายลักษณะ ตัวอย่างต่อไปนี้แสดงผลสำเร็จของการนำ data mining ไปใช้

- ด้านการแพทย์: ใช้ data mining ค้นหาผลข้างเคียงของการใช้ยาโดยอาศัยข้อมูลจากแฟ้มประวัติผู้ป่วย, ใช้ในการวิเคราะห์หาความสัมพันธ์ของสารพันธุกรรม
- ด้านการเงิน: ใช้ data mining ตัดสินว่าควรจะอนุมัติเครดิตให้ลูกค้ารายใดบ้าง, ใช้ในการคาดหมายความน่าจะเป็นว่าธุรกิจนั้นๆมีโอกาสที่จะล้มละลายหรือไม่ และใช้คาดหมายการขึ้น/ลงของหุ้นในตลาดหุ้น
- ด้านการเกษตร: ใช้จำแนกประเภทของโรคพืชที่เกิดกับถั่วเหลืองและมะเขือเทศ
- ด้านวิศวกรรม: ใช้วิเคราะห์และวินิจฉัยสาเหตุการทำงานผิดพลาดของเครื่องจักรกล
- ด้านอาชญวิทยา: ใช้วิเคราะห์หาเจ้าของลายนิ้วมือ
- ด้านอวกาศ: ใช้วิเคราะห์ข้อมูลที่ส่งมาจากดาวเทียมขององค์การนาซ่า

หัวใจสำคัญของกระบวนการ data mining คือส่วนของโปรแกรมที่ทำหน้าที่สังเคราะห์ความรู้ขึ้นมาจากข้อมูลจำนวนมากในฐานข้อมูล ส่วนสังเคราะห์ความรู้นี้เรียกว่า learning algorithm ซึ่งมีผู้เสนอแนวคิดและพัฒนาอัลกอริทึมส่วนนี้ขึ้นเป็นจำนวนมาก ได้แก่ อัลกอริทึมที่ใช้หลักการของการสร้างต้นไม้ตัดสินใจ (decision-tree induction algorithm) ตัวอย่างเช่น โปรแกรม ID3, C4.5, J48 อัลกอริทึมที่ใช้หลักการทางสถิติและทฤษฎีของเบย์ส์ ตัวอย่างเช่น โปรแกรม naïve Bayes อัลกอริทึมที่ใช้หลักการของ neural network และอัลกอริทึมอื่นๆอีกมาก ได้มีนักคอมพิวเตอร์ทดสอบเปรียบเทียบความสามารถของอัลกอริทึมแต่ละประเภท เพื่อค้นหาว่าอัลกอริทึมใดมีความสามารถสูงที่สุด ผลการทดสอบส่วนใหญ่ที่ปรากฏจะชี้ว่าไม่มีอัลกอริทึมใดที่ทำงานได้ดีที่สุดในข้อมูลทุกประเภท ทั้งนี้เนื่องจากข้อมูลแต่ละประเภทมีลักษณะเฉพาะตัวที่ต่างกัน เช่น ข้อมูลทางการแพทย์ จะต่างจากข้อมูลด้านกฎหมาย และต่างจากข้อมูลด้านอวกาศ ดังนั้นจึงไม่มีอัลกอริทึมใดที่ดีที่สุดสำหรับข้อมูลทุกประเภท

โครงการวิจัยนี้มีวัตถุประสงค์เพื่อศึกษาลักษณะของอัลกอริทึมสังเคราะห์ความรู้ หรือ learning algorithm ที่เหมาะสมกับข้อมูลด้านการแพทย์โดยเน้นที่ข้อมูลการวิเคราะห์และวินิจฉัยโรค เพื่อวิเคราะห์ว่าอัลกอริทึมในกลุ่มใดหรือประเภทใดที่มีประสิทธิภาพการสังเคราะห์หรือการเรียนรู้ที่ดีที่สุด เมื่อคัดเลือกอัลกอริทึมหรือกลุ่มของอัลกอริทึมได้แล้วขั้นตอนต่อไปจะ

เป็นการพิจารณาปรับปรุงอัลกอริทึมนั้นด้วยเทคนิคต่างๆ เพื่อจะเพิ่มขีดความสามารถการวิเคราะห์และวินิจฉัยโรคได้แม่นยำมากขึ้น โครงการศึกษาวิจัยนี้ยังจะสามารถใช้เป็นแนวทางสำคัญในการพัฒนา data mining เฉพาะทางด้านการแพทย์ที่เกี่ยวข้องกับการวินิจฉัยและตรวจรักษาโรคได้ต่อไปในอนาคต

1.2 วัตถุประสงค์ของการวิจัย

(1) เพื่อศึกษาค้นคว้า และวิเคราะห์อัลกอริทึมสังเคราะห์ความรู้ (learning algorithm) ที่สามารถให้โมเดลของการวินิจฉัยโรคต่างๆ ได้แม่นตรงที่สุด

(2) เพื่อค้นหาเทคนิคที่จะช่วยปรับปรุงโมเดลที่ได้ให้มีอัตราความถูกต้อง (success rate) ในการวินิจฉัยโรคเพิ่มขึ้น

(3) สามารถสรุปลักษณะของอัลกอริทึม และเทคนิคที่เหมาะสมกับข้อมูลการวินิจฉัยทางการแพทย์เพื่อเป็นแนวทางในการพัฒนาระบบ data mining เพื่องานทางการแพทย์ได้ต่อไปในอนาคต

1.3 ขอบเขตของการวิจัย

โครงการวิจัยนี้จะมุ่งเน้นไปที่การวิเคราะห์อัลกอริทึมที่สังเคราะห์ความรู้ ในลักษณะของการจำแนกข้อมูล (classification) เกณฑ์ในการศึกษาเพื่อคัดเลือกอัลกอริทึมที่เหมาะสมกับข้อมูลการวินิจฉัยโรค จะประกอบด้วย เวลาที่ใช้ในการสังเคราะห์ความรู้ (time to build model), ความแม่นตรง (accuracy) ของโมเดลที่สังเคราะห์ขึ้น รวมถึง sensitivity (ความแม่นตรงเฉพาะในกลุ่ม positive instances) และ specificity (ความแม่นตรงเฉพาะในกลุ่ม negative instances)

ข้อมูลวินิจฉัยโรคที่เลือกมาใช้ทดสอบจะครอบคลุมข้อมูลทุกลักษณะ ได้แก่ ข้อมูลชนิดตัวเลขที่มีข้อมูลครบทุกแอททริบิวต์ (numeric data with no missing value), ข้อมูลชนิดตัวเลขที่มีบางข้อมูลหายไป (numeric data with missing values), ข้อมูลชนิดข้อความที่มีข้อมูลครบทุกแอททริบิวต์ (nominal data with no missing value), ข้อมูลชนิดข้อความที่มีบางข้อมูลหายไป (nominal data with missing values)

วิธีการทดสอบความแม่นตรง (accuracy estimation method) ของโมเดลที่สังเคราะห์ได้จะใช้วิธี stratified 10-fold cross-validation

1.4 ประโยชน์ที่ได้รับจากการวิจัย

ในปัจจุบันมีซอฟต์แวร์ด้าน data mining ที่พัฒนาขึ้นเพื่อการค้าเป็นจำนวนมาก ซอฟต์แวร์เหล่านี้ใช้ในงาน data mining ได้เอนกประสงค์ นั่นคือ ใช้ในการค้นหาความรู้จากข้อมูลประเภทใดก็ได้ แต่จากผลการทดลองโดยนักวิจัยจำนวนมากให้ผลสรุปที่ตรงกันคือ ไม่มีโปรแกรม data mining ใดที่ใช้ได้ดีกับข้อมูลทุกประเภท งานวิจัยนี้จึงมุ่งที่จะศึกษาค้นคว้าหาเทคนิค data mining ที่เหมาะสมกับข้อมูลเฉพาะทางด้านการวินิจฉัยโรค เพื่อให้ได้เทคนิคและอัลกอริทึมที่ดีที่สุด ให้ผลการวินิจฉัยโรคที่แม่นยำมากที่สุด ซึ่งจะสามารถนำมาใช้ประโยชน์ในการช่วยแพทย์วินิจฉัยโรค หรือเพื่อเป็นการบอกแนวโน้มว่าคนไข้แต่ละรายมีโอกาสจะพัฒนาความเสี่ยงไปสู่การเป็นโรคร้ายแรงใดหรือไม่

นอกจากนี้องค์ความรู้ที่ได้จากโครงการวิจัยนี้ จะสามารถใช้เป็นพื้นฐานในการพัฒนาระบบ data mining เพื่อการวินิจฉัยโรคขึ้นใช้เองได้ต่อไปในอนาคต หน่วยงานที่สามารถนำผลการวิจัยไปใช้ประโยชน์ ประกอบด้วยโรงพยาบาลทั่วประเทศ สถาบันและหน่วยงานทางด้านสาธารณสุขและวิทยาศาสตร์สุขภาพ รวมถึงกลุ่มนักวิจัยในสาขา medical data mining

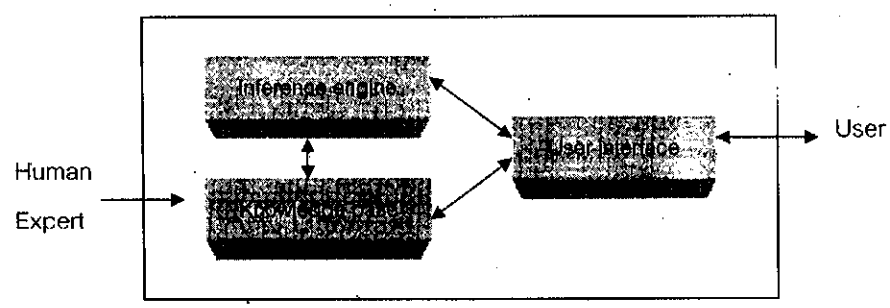
บทที่ 2
ระบบอัตโนมัติเพื่อสนับสนุนการวินิจฉัยโรค

ความก้าวหน้าทางด้านเทคโนโลยีสารสนเทศ และระบบคอมพิวเตอร์ที่มีประสิทธิภาพสูง ราคาไม่แพง ทำให้มีการนำคอมพิวเตอร์มาใช้ช่วยในการเก็บข้อมูล เพื่อความสะดวกรวดเร็วในการสืบค้นข้อมูล การนำคอมพิวเตอร์มาช่วยประมวลผลข้อมูลเกิดขึ้นในทุกวงการ รวมถึงวงการสาธารณสุขที่การจัดเก็บระเบียนประวัติ และประวัติการตรวจรักษา เริ่มถูกปรับเปลี่ยนให้เป็นระบบอัตโนมัติมากขึ้นด้วยการนำคอมพิวเตอร์มาเป็นเครื่องมือสำคัญในระบบอัตโนมัตินี้ แต่ขีดความสามารถของระบบคอมพิวเตอร์มีสูงกว่าการเป็นเพียงเครื่องมือสืบค้นข้อมูลอัตโนมัติ นั่นคือคอมพิวเตอร์สามารถถูกปรับให้มีความฉลาดมากขึ้น ด้วยการนำเทคโนโลยีทางด้านปัญญาประดิษฐ์ (Artificial Intelligence หรือ AI) เข้ามาผสมผสาน ระบบที่ประสบความสำเร็จมากได้แก่ ระบบผู้เชี่ยวชาญ (expert system) ในปัจจุบันงานวิจัยด้าน AI ที่ก้าวหน้ามากขึ้น ทำให้นักวิจัยสามารถพัฒนาระบบอัตโนมัติที่ฉลาดขึ้นกว่าระบบผู้เชี่ยวชาญ ระบบรุ่นใหม่นี้เรียกว่า ระบบเหมืองข้อมูล (data mining system) ระบบนี้มองข้อมูลที่เก็บอยู่เป็นเสมือนแหล่งความรู้ขนาดใหญ่ หน้าที่ของระบบคือ พยายามค้นหาความรู้ออกมาจากกลุ่มข้อมูลขนาดใหญ่ เนื้อหาในบทนี้อธิบายความแตกต่างระหว่างระบบเหมืองข้อมูลและระบบผู้เชี่ยวชาญ กระบวนการและเทคนิคการทำงานของระบบเหมืองข้อมูล รวมทั้งการประยุกต์ใช้ระบบเหมืองข้อมูลเพื่อสนับสนุนงานทางการแพทย์โดยเฉพาะการวินิจฉัยโรค

2.1 ระบบผู้เชี่ยวชาญ (Expert System)

ในช่วงปลายทศวรรษที่ 1970 ต่อเนื่องถึงต้นทศวรรษที่ 1980 ได้มีการนำ AI มาประยุกต์ใช้กับงานด้านการแพทย์ โดยการพัฒนาเป็นระบบผู้เชี่ยวชาญที่ช่วยสนับสนุนการตัดสินใจในเรื่องการวินิจฉัยโรค ระบบผู้เชี่ยวชาญในยุคแรกที่ประสบความสำเร็จและมีชื่อเสียงมากที่สุด ได้แก่ ระบบ MYCIN (Shortliffe, 1976) จากความสำเร็จของ MYCIN ทำให้มีผู้พัฒนาระบบผู้เชี่ยวชาญเพื่อสนับสนุนงานทางการแพทย์ขึ้นอีกเป็นจำนวนมาก ตัวอย่างเช่น HODGKINS (Safrans et al., 1976), PIP (Pauker et al., 1976; Szolovits and Rauker, 1978), CASNET (Weiss et al.,1978), HEADMED (Haiser et al., 1978), PUFF (Kunz et al., 1987), CENTAUR (Aikins, 1997), VM (Fagan et al., 1980), ONCOCIN (Shortliffe et al., 1981), ABEL (Patil et al., 1982), GALEN (Thompson et al., 1983), MDX (Chandrasekaran and Mittal, 1983)

ในจำนวนระบบผู้เชี่ยวชาญเพื่องานด้านการแพทย์ที่ได้รับการพัฒนาขึ้นจำนวนมากเหล่านี้ ระบบ Internist-I (Pople, 1982; Miller et al., 1982) ซึ่งต่อมาได้รับการพัฒนาเป็นระบบ CADUCEUS (Miller, 1984) เป็นระบบที่ครอบคลุมการวินิจฉัยทางการแพทย์ได้กว้างขวางที่สุด และมีการผนวกโครงข่ายทางด้านพยาธิวิทยาและสรีรวิทยา เพื่อให้สามารถอธิบายสาเหตุของโรคได้ละเอียดขึ้น งานวิจัยด้านระบบผู้เชี่ยวชาญจะมุ่งเน้นที่การหารูปแบบที่เหมาะสมในการสร้างฐานความรู้ของผู้เชี่ยวชาญ รวมถึงการสืบค้นความรู้จากฐานความรู้และการอธิบายให้เหตุผล โครงสร้างโดยทั่วไปของระบบผู้เชี่ยวชาญ แสดงได้ดังรูปที่ 2.1



รูปที่ 2.1 โครงสร้างของระบบผู้เชี่ยวชาญ

หัวใจสำคัญของระบบผู้เชี่ยวชาญ คือ ฐานความรู้ (knowledge base) ที่รวบรวมความรู้ของผู้ชำนาญการ (human expert) จำนวนมาก และมักจะเก็บอยู่ในรูปแบบของกฎแบบมีเงื่อนไข (IF..THEN rules) การรวบรวมความรู้มักจะใช้วิธีการสัมภาษณ์ผู้ชำนาญการในสาขานั้นๆ แล้วแปลงให้เป็นกฎแบบมีเงื่อนไข แต่ปัญหาที่มักจะเกิดขึ้น คือ ความรู้จากการสัมภาษณ์ผู้ชำนาญการแต่ละคนอาจจะไม่ตรงกัน และอาจจะไม่สมบูรณ์ครอบคลุมการวินิจฉัยได้ครบทุกกรณี

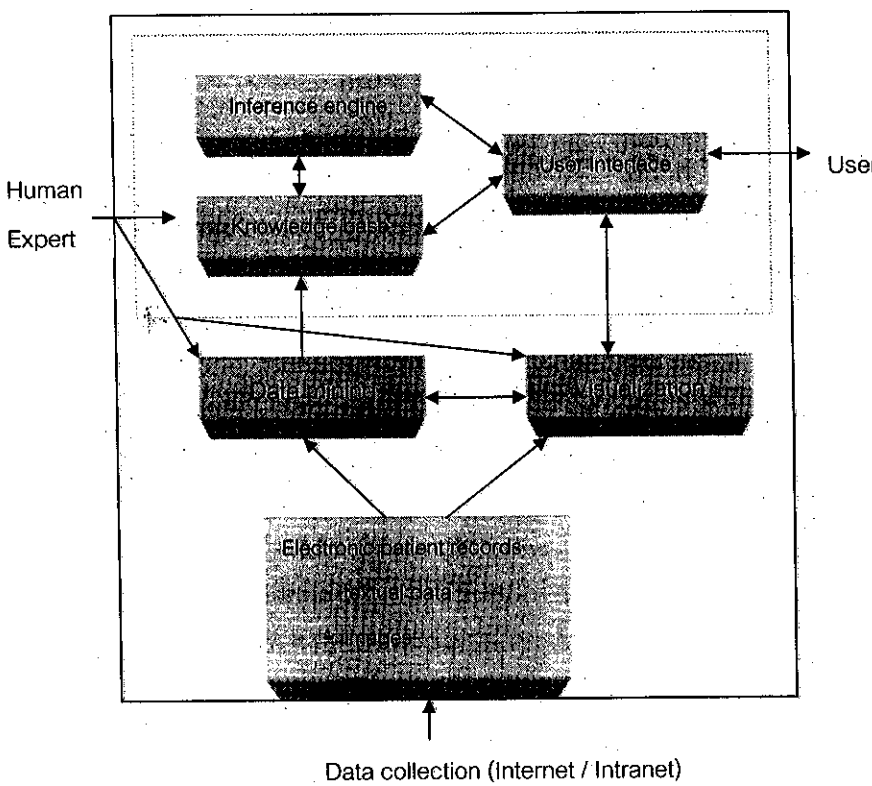
ในช่วงเวลาใกล้เคียงกันนั้นงานวิจัยในสาขาการเรียนรู้ของเครื่องคอมพิวเตอร์ (machine learning) ได้มีความก้าวหน้าขึ้นมาก สามารถพัฒนาให้คอมพิวเตอร์เรียนรู้และสร้างกฎแบบมีเงื่อนไขขึ้นมาจากข้อมูลกรณีตัวอย่าง (case-based learning) จึงได้มีการพัฒนาระบบผู้เชี่ยวชาญให้สามารถสร้างฐานความรู้ขึ้นมาจากข้อมูลได้โดยตรง แทนที่จะใช้การสอบถามความรู้จากผู้ชำนาญการเพียงอย่างเดียว ระบบผู้เชี่ยวชาญที่เริ่มนำเทคนิค machine learning เข้ามาผสมผสานได้แก่ ระบบ KARDIO (Bratko et al., 1989) ที่ใช้ช่วยวินิจฉัยคลื่นหัวใจ

จากความสำเร็จของระบบ KARDIO ทำให้ระบบอัตโนมัติเพื่อสนับสนุนการวินิจฉัยโรคในช่วงทศวรรษที่ 1990 เริ่มเปลี่ยนแปลงจากระบบผู้เชี่ยวชาญที่สร้างฐานความรู้จากประสบการณ์และความเชี่ยวชาญของผู้ชำนาญการที่เป็นมนุษย์ เป็นระบบที่มีความสามารถสังเคราะห์ความรู้ขึ้นได้เอง โดยอาศัยข้อมูลกรณีตัวอย่างที่รวบรวมไว้ในฐานข้อมูลเป็นข้อมูลฝึกในกระบวนการสังเคราะห์

ความรู้ ความสามารถในการสังเคราะห์ความรู้เป็นเทคนิคที่นำมาจากสาขา machine learning ซึ่งเป็นสาขาย่อยของ AI ระบบในรูปแบบใหม่นี้มีชื่อเรียกว่า ระบบเหมืองข้อมูล (data-mining system)

2.2 ระบบเหมืองข้อมูล (Data Mining System)

ระบบเหมืองข้อมูล เป็นระบบที่สามารถสังเคราะห์และค้นหาความรู้จากกลุ่มของข้อมูลได้โดยอัตโนมัติ ทำให้มีชื่อเรียกอีกชื่อหนึ่งว่า KDD ซึ่งย่อมาจาก Knowledge Discovery in Databases ความสามารถในการสังเคราะห์ความรู้ขึ้นได้เองนี้ทำให้ระบบเหมืองข้อมูลเหมาะสมที่จะถูกนำไปใช้ช่วยในระบบสนับสนุนการตัดสินใจ (decision-support system) โครงสร้างของระบบสนับสนุนการตัดสินใจที่ผนวกส่วนของ data mining เข้ามาทำหน้าที่สังเคราะห์ความรู้จากข้อมูล แสดงได้ดังรูปที่ 2.2



รูปที่ 2.2 โครงสร้างของระบบสนับสนุนการตัดสินใจทางการแพทย์ที่ผนวกส่วน data mining

จากโครงสร้างของระบบสนับสนุนการตัดสินใจ จะเห็นได้ว่า data mining เป็นส่วนที่เพิ่มเข้ามาเพื่อช่วยให้การสร้างฐานความรู้เป็นระบบอัตโนมัติมากขึ้น แต่ผลลัพธ์ที่ได้จากการสังเคราะห์ความรู้โดยระบบ data mining ยังคงต้องให้ผู้ชำนาญการเป็นผู้ตัดสินใจในขั้นสุดท้ายว่า มีความถูกต้อง และมีประโยชน์สมควรที่จะเพิ่มความรู้นั้นเข้าไปในฐานความรู้ของระบบอัตโนมัตินี้หรือไม่

ความรู้ที่สังเคราะห์ขึ้นโดยระบบ data mining แสดงได้ในหลายรูปแบบ เช่น แสดงเป็นภาพ แสดงเป็นกราฟ หรือเป็นสมการคณิตศาสตร์ รูปแบบที่เหมาะสมที่สุดที่จะนำมาใช้ในงานทางการแพทย์เป็นรูปแบบของกฎแบบมีเงื่อนไข ดังตัวอย่างในรูปที่ 2.3 ที่แสดงความรู้ที่สังเคราะห์ขึ้นจากข้อมูลจำนวนมากของคนไข้ที่ป่วยด้วยโรคเกี่ยวกับข้อ (rheumatic disease) ในลักษณะของกฎแบบมีเงื่อนไข (Dzeroski and Lavrac, 1996)

```
IF Sex = male
    AND  Age > 46
    AND  Number_of_painful_joints > 3
    AND  Skin_manifestations = psoriasis
THEN
    Diagnosis = Crystal_induced_synovitis
```

รูปที่ 2.3 ตัวอย่างความรู้ที่ค้นพบโดยระบบ data mining

ระบบ data mining ที่มีความสามารถในการค้นหาความรู้จากข้อมูลคนไข้จำนวนมาก และแสดงความรู้ในรูปของกฎแบบมีเงื่อนไข ได้แก่ ระบบ AQ15 (Michalski, 1986) และ CN2 (Dzeroski and Lavrac, 1996) เทคนิคที่ใช้ในการค้นหาความรู้ของทั้งสองระบบนี้ ใช้วิธีการสังเคราะห์กฎเชิงอุปนัย (rule induction) โดยเริ่มต้นการทำงานจากกฎที่ยังไม่มีเงื่อนไข จากนั้นใช้ข้อมูลคนไข้ที่รวบรวมไว้ในฐานข้อมูลเป็นเครื่องมือในการปรับกฎให้อยู่ในรูปแบบที่ถูกต้องมากขึ้น

นอกจากการค้นหาความรู้ด้วยเทคนิคการสังเคราะห์กฎเชิงอุปนัยแล้ว ยังมีเทคนิคอื่นๆ ที่ใช้ได้ผลดีอีกจำนวนมาก เช่น เทคนิค rough set ที่นำมาใช้ช่วยวิเคราะห์ข้อมูลการวินิจฉัยจากคลื่นหัวใจ (Grzymala-Busse,1998; Komorowski and Ohm, 1998) เทคนิคการค้นหาความสัมพันธ์ (Agrawal et al., 1996) ที่แสดงความรู้ในลักษณะของความสัมพันธ์ภายในชุดข้อมูล เช่น "80% ของคนไข้ที่เป็นปอดบวม จะมีอาการไข้สูงร่วมด้วย โดยปรากฏข้อมูลสนับสนุนกฎนี้ 10% จากข้อมูลคนไข้ทั้งหมด"

การแสดงความรู้ที่ค้นพบในรูปของกฎ บางครั้งจะใช้การซ้อนเงื่อนไขหลายระดับชั้น ดังตัวอย่างในรูปที่ 2.4 ที่แสดงการวินิจฉัยการเลือกชนิดของคอนแท็กเลนส์ให้กับคนไข้

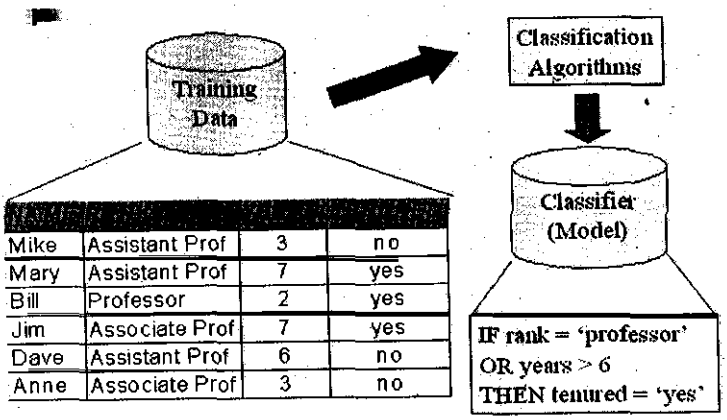
```
IF astigmatism = no AND tear_production = normal
THEN soft_lenses
ELSE
        IF prescription = myope AND tear_production = normal
        THEN hard_lenses
```

รูปที่ 2.4 แสดงบางส่วนของการวินิจฉัยเลือกชนิดคอนแท็กเลนส์

2.3 อัลกอริทึมและเทคนิคการทำเหมืองข้อมูล

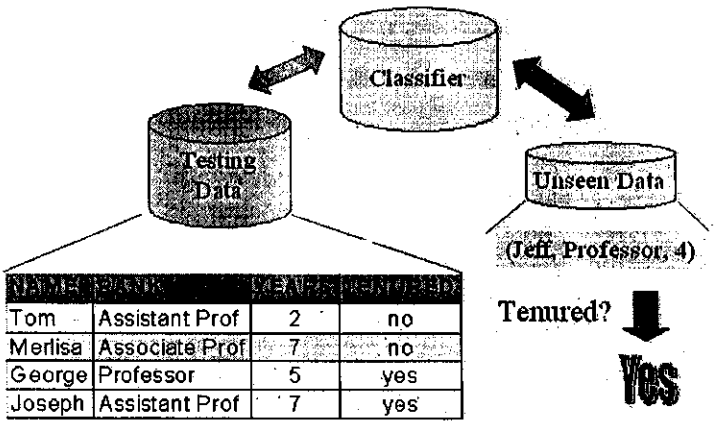
การทำเหมืองข้อมูลมีได้หลายประเภทตามลักษณะของงาน เช่นการค้นหารูปแบบหรือโมเดลเพื่อการจำแนก (classification) การค้นหาความสัมพันธ์ภายในกลุ่มข้อมูล (association) การค้นหารูปแบบเพื่อการจัดกลุ่มข้อมูล (clustering) การสรุปข้อมูล (summarization) ในงานวิจัยนี้จะเน้นการศึกษาเฉพาะงาน classification เนื่องจากใช้ประโยชน์ได้โดยตรงกับงานวินิจฉัยโรคอัตโนมัติ

การทำเหมืองข้อมูลประเภท classification แบ่งการทำงานเป็นสองช่วง คือช่วงสร้างโมเดล (รูปที่ 2.5) และช่วงทดสอบโมเดล (รูปที่ 2.6) เพื่อวัดความแม่นตรงของโมเดล โมเดลที่ได้จะเรียกว่าตัวจำแนก (classifier) นำไปใช้ประโยชน์ในการทำนายข้อมูลใหม่ เช่น ใช้ตัวจำแนกทำนายได้ว่า Professor Jeff ซึ่งมีอายุงาน 4 ปี จะมีสถานภาพ Tenured



Mike	Assistant Prof	3	no
Mary	Assistant Prof	7	yes
Bill	Professor	2	yes
Jim	Associate Prof	7	yes
Dave	Assistant Prof	6	no
Anne	Associate Prof	3	no

รูปที่ 2.5 แสดงขั้นตอนการสร้างโมเดลเพื่อใช้เป็นตัวจำแนกข้อมูล



รูปที่ 2.6 แสดงการทดสอบโมเดลและการใช้โมเดลจำแนกข้อมูลในอนาคต

อัลกอริทึมที่ใช้ในการทำ classification มีหลายอัลกอริทึม เช่น tree-based learners, rule learners, statistical learners, support vector machine, instance-based learners, multi-layer perceptron

ในงานวิจัยนี้ศึกษาเปรียบเทียบอัลกอริทึมในสี่กลุ่ม คือ rule learner (ใช้อัลกอริทึม OneR (Holt, 1993)), tree-based learner (ใช้อัลกอริทึม J48 ที่มีวิธีการทำงานเหมือน C4.5 (Quinlan, 1993)), statistical learner (ใช้อัลกอริทึม naive Bayes), และ instance-based learner (ใช้อัลกอริทึม 10-nearest neighbors)

ข้อมูลที่จะใช้ประกอบการอธิบายอัลกอริทึมทั้งสี่กลุ่ม เป็นข้อมูลสภาพอากาศที่ใช้ประกอบการตัดสินใจการเล่นกีฬากอล์ฟว่าสภาพอากาศเช่นไร จึงจะเล่น (play = yes) และสภาพอากาศเช่นไร จึงจะไม่เล่น (play = no) ข้อมูลที่เป็นจุดมุ่งหมาย (goal attribute) ของการจำแนกคลาส คือ แอททริบิวต์ play โดย แอททริบิวต์ outlook, temperature, humidity , และ windy ทำหน้าที่เป็นแอททริบิวต์ประกอบการทำนาย (predicting attributes)

ตารางที่ 2.1 ข้อมูลที่ใช้ประกอบการตัดสินใจเล่นกอล์ฟ

No.	Attributes				Play
	Outlook	Temperature	Humidity	Windy	
1	sunny	hot	high	false	N
2	sunny	hot	high	true	N
3	overcast	hot	high	false	P
4	rainy	mild	high	false	P
5	rainy	cool	normal	false	P
6	rainy	cool	normal	true	N
7	overcast	cool	normal	true	P
8	sunny	mild	high	false	N
9	sunny	cool	normal	false	P
10	rainy	mild	normal	false	P
11	sunny	mild	normal	true	P
12	overcast	mild	high	true	P
13	overcast	hot	normal	false	P
14	rainy	mild	high	true	N

2.3.1 Rule learner

อัลกอริทึมในกลุ่ม Rule learner ได้แก่ OneR (หรือบางครั้งเรียกว่า simple-rule algorithm หรือ 1R) เป็นอัลกอริทึมอย่างง่ายที่ใช้ในการสร้าง classifier ที่เรียกชื่อว่า OneR เนื่องจาก อัลกอริทึมนี้จะสร้าง classifier ในรูปแบบ "IF *condition* THEN *specified-class*" โดยจะมีเพียง แอททริบิวต์เดียวปรากฏใน *condition* เช่น IF outlook = sunny THEN play = no

ขั้นตอนในการสร้าง classifier ประกอบด้วย

(1) พิจารณาแต่ละค่าของทุกแอททริบิวต์ (ยกเว้น goal attribute) เพื่อนำมาสร้าง rules

(1.1) นับจากข้อมูลว่าค่านั้นๆ ของ attribute ให้ผลของ class ใดมากที่สุด

(1.2) สร้าง rule ด้วยรูปแบบ

IF attribute = value THEN majority_class

(1.3) คำนวณอัตราความผิดพลาดของ rule

(เช่น ถ้าแอททริบิวต์ outlook ปรากฏค่า sunny รวม 5 ครั้ง ในจำนวน 5 ครั้งนี้ให้ผลว่า play = yes 2 ครั้ง และให้ผลว่า play = no 3 ครั้ง ดังนั้น rule ที่สร้างขึ้น

IF outlook = sunny THEN play = no

มีอัตราความผิดพลาด = $\frac{2}{5}$)

(2) คัดเลือกเฉพาะ rule ที่ให้ค่าอัตราความผิดพลาดต่ำ

แสดงผลที่ได้จากการทำงานของอัลกอริทึม

กลุ่มที่ 1 :

IF outlook = sunny THEN play = no (error rate = 2/5)
IF outlook = overcast THEN play = yes (error rate = 0/4)
IF outlook = rainy THEN play = yes (error rate = 2/5)

กลุ่มที่ 2 :

IF temperature = hot THEN play = no (error rate = 2/4)
IF temperature = mild THEN play = yes (error rate = 2/6)
IF temperature = cool THEN play = yes (error rate = 1/4)

กลุ่มที่ 3 :

IF humidity = high THEN play = no (error rate = 3/7)
IF humidity = normal THEN play = yes (error rate = 1/7)

กลุ่มที่ 4 :

IF windy = false THEN play = yes (error rate = 2/8)
IF windy = true THEN play = no (error rate = 3/6)

เลือกกลุ่มที่ให้ค่าความผิดพลาดต่ำที่สุดเป็นโมเดลที่จะใช้ประโยชน์ในการจำแนก ซึ่งผลลัพธ์ที่ได้คือ

IF outlook = sunny THEN play = no
IF outlook = overcast THEN play = yes
IF outlook = rainy THEN play = yes

2.3.2 Tree-based learner

อัลกอริทึมในกลุ่มนี้ใช้ต้นไม้ตัดสินใจ (decision tree) เป็นเครื่องมือในการสังเคราะห์โมเดล ขั้นตอนในการสร้าง decision tree เพื่อใช้จำแนกข้อมูล มีดังนี้

(1) เลือกแอททริบิวต์ที่ทำหน้าที่เป็น root node

(2) จาก root node *สร้างเส้นทางเชื่อมไปยังโหนดลูก จำนวนเส้นทางเชื่อมจะเท่ากับจำนวน* ค่าที่เป็นไปได้ทั้งหมดของแอททริบิวต์ที่เป็น root node

(3) ถ้าโหนดลูก เป็นกลุ่มของข้อมูลที่อยู่ในคลาสเดียวกันทั้งหมด ให้หยุดการสร้าง tree แต่ถ้าโหนดลูกมีข้อมูลของหลายคลาสปะปนกันอยู่ ต้องสร้าง subtree เพื่อจำแนกข้อมูลต่อไป โดยเลือกแอททริบิวต์มาทำหน้าที่เป็น root node ของ subtree และทำซ้ำในขั้นตอนที่ 2, 3

ปัญหาที่ต้องพิจารณาในการสร้าง decision tree คือ ควรจะตัดสินใจเลือกแอททริบิวต์ใดมาทำหน้าที่เป็น root node ในแต่ละขั้นตอนของการสร้าง tree และ subtree

เกณฑ์ที่ใช้ช่วยประกอบการเลือกแอททริบิวต์คือ ทดลองเลือกแต่ละแอททริบิวต์มาทำหน้าที่เป็น root node และวัดค่า gain ซึ่งเป็นค่าที่ชี้ว่าแอททริบิวต์นั้นจะช่วยจำแนกคลาสของข้อมูลได้ดีเพียงใด (การจำแนกที่ดีที่สุด คือให้ leaf node ที่เป็นข้อมูล คลาสเดียวกันทั้งหมด) ค่า gain ที่สูงที่สุดหมายถึง การจำแนกคลาสที่ดีที่สุด

จากตัวอย่างข้อมูลสภาพอากาศที่ใช้ตัดสินใจว่าจะเล่นกีฬาหรือไม่ นำมาสร้าง decision tree โดยเริ่มจากการสร้าง root node ข้อมูลสภาพอากาศประกอบด้วย 4 แอททริบิวต์คือ outlook, temperature, humidity และ windy โดยแอททริบิวต์ play เป็น goal attribute (เป้าหมายของการทำ classification) ดังนั้นทั้งสี่แอททริบิวต์มีโอกาสทำหน้าที่เป็น root node ได้

ทดลองสร้าง tree โดยในครั้งแรกให้แอททริบิวต์ outlook ทำหน้าที่เป็น root node ความสามารถในการจำแนกคลาสของข้อมูล = 0.247 ครั้งที่สองทดลองสร้าง tree โดยให้แอททริบิวต์ temperature ทำหน้าที่เป็น root node คำนวณความสามารถในการจำแนกคลาสของข้อมูลได้เท่ากับ 0.029 การทดลองครั้งที่สาม ให้แอททริบิวต์ humidity เป็น root node คำนวณความสามารถในการจำแนกคลาสของข้อมูลได้เท่ากับ 0.152 และในครั้งสุดท้ายทดลองให้แอททริบิวต์ windy เป็น root node ซึ่งให้ค่าความสามารถจำแนกคลาสของข้อมูล = 0.048 นำค่าความสามารถจำแนกคลาสของข้อมูล ของการทดลองสร้าง decision tree ทั้ง 4 ครั้งมาเปรียบเทียบ

gain (outlook)	=	0.247	bits
gain (temperature)	=	0.029	bits
gain (humidity)	=	0.152	bits
gain (windy)	=	0.048	bits

ตัดสินใจเลือกแอททริบิวต์ outlook เป็น root node เพราะให้ค่า gain สูงที่สุด

gain เป็นค่าที่บอกระดับความสามารถของการจำแนกคลาสของแอททริบิวต์ ที่ถูกเลือกให้ทำหน้าที่เป็นตัวตรวจสอบเพื่อจัดกลุ่มของข้อมูล แอททริบิวต์ที่ให้ค่า gain สูง คือแอททริบิวต์ที่จัดกลุ่มข้อมูลแล้วได้ข้อมูลในแต่ละ leaf node เป็นคลาสเดียวกันทั้งหมด หรือมีข้อมูลต่างคลาสปะปนมาบ้างเพียงเล็กน้อย

ถ้าให้ T แทน เซตของข้อมูลฝึก (training data)

X แทน attribute ที่ถูกเลือกให้เป็นตัวตรวจสอบเพื่อจัดกลุ่มข้อมูล

gain (X) = info(T) – info $_x$ (T)

info(T) คือ ฟังก์ชันที่ระบุปริมาณข้อมูลที่ต้องการเพื่อให้สามารถจำแนกคลาสของข้อมูลได้

$$= - \sum_{j=1 \text{ to } k} [\text{freq}(C_j, T) / |T|] \times \log_2 [\text{freq}(C_j, T) / |T|] \quad \text{bits}$$

เมื่อ |T| คือ จำนวนข้อมูลทั้งหมดในเซตของข้อมูลฝึก

freq (C_j, T) คือ ความถี่ที่ข้อมูลใน T ปรากฏเป็นคลาส C_j

info$_X$ (T) คือ ฟังก์ชันที่ระบุปริมาณข้อมูลที่ต้องการเพื่อการจำแนกคลาสของข้อมูล โดยใช้แอททริบิวต์ X เป็นตัวตรวจสอบเพื่อจำแนกกลุ่มของข้อมูล

$$= \sum_{i=1 \text{ to } n} (|T_i| / |T|) \times \text{info}(T_i) \quad \text{bits}$$

เมื่อ i คือ จำนวนค่าที่เป็นไปได้ของแอททริบิวต์ X

$|C_i|$ คือ จำนวนข้อมูลที่มีค่า X = i

จากตัวอย่างข้อมูลสภาพอากาศ เซตของข้อมูลฝึก T ประกอบด้วยข้อมูล 2 คลาส คือ play = yes และ play = no การจะระบุว่าข้อมูลหนึ่งเรคคอร์ดอยู่ในคลาส yes หรือ no ต้องการปริมาณข้อมูลประกอบการตัดสินใจจำแนกคลาสดังนี้

info(play) = [- (ความถี่ของการปรากฏข้อมูลเป็นคลาส yes / จำนวนข้อมูลทั้งหมด)
x $\log_2$ (ความถี่ของการปรากฏข้อมูลเป็นคลาส yes / จำนวนข้อมูลทั้งหมด)]
+ [- (ความถี่ของการปรากฏข้อมูลเป็นคลาส no / จำนวนข้อมูลทั้งหมด)
x $\log_2$ (ความถี่ของการปรากฏข้อมูลเป็นคลาส no / จำนวนข้อมูลทั้งหมด)]

$= - (9/14) \times \log_2 (9/14) - (5/14) \times \log_2 (5/14)$

$= 0.940$ bits

การจะจำแนกคลาสของข้อมูลออกเป็น play = yes หรือ play = no ต้องใช้ข้อมูลจากแอททริบิวต์อื่นประกอบการตัดสินใจ ถ้าเลือกแอททริบิวต์ outlook จะต้องการปริมาณข้อมูลเพิ่มเพื่อประกอบการเลือกคลาส ดังนี้

$\text{info}_{\text{outlook}}(T) = (5/14) \times [- (2/5) \times \log_2(2/5) - (3/5) \times \log_2(3/5)]$
$+ (4/14) \times [- (4/4) \times \log_2(4/4) - (0/4) \times \log_2(0/4)]$
$+ (5/14) \times [- (3/5) \times \log_2(3/5) - (2/5) \times \log_2(2/5)]$
$= 0.693$ bits

นั่นคือ ถ้ามีข้อมูลใหม่เข้ามา เมื่อพิจารณาจากค่า outlook ของข้อมูลใหม่นี้ จะต้องใช้ข้อมูลพิ่มอีก 0.693 bits จึงจะบอกคลาสที่ถูกต้องของข้อมูลใหม่นี้ได้

ค่า info(T) นี้เรียกได้อีกอย่างว่า ค่า ***entropy***

$$entropy(P_1,P_2, \dots , P_n) = -P_1 \log P_1 - P_2 \log P_2 - \dots - P_n \log P_n$$

$$info([2,4,3]) = entropy(2/9, 4/9, 3/9)$$
$$= -(2/9) \log (2/9) - (4/9) \log (4/9) - (3/9) \log (3/9)$$
$$= (-2 \log 2 - 4 \log 4 - 3 \log 3 + 9 \log 9) / 9$$

แอททริบิวต์ที่สามารถถูกเลือกมาเป็นตัวทดสอบเพื่อจัดกลุ่มของข้อมูลฝึก คือ แอททริบิวต์ outlook, temperature, humidity และ windy คำนวณค่า gain จากการเลือกแต่ละแอททริบิวต์ได้ดังนี้

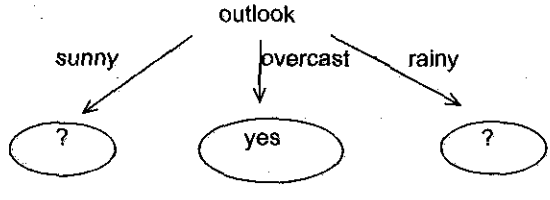
$$gain(outlook) = info(T) - info_{outlook}(T)$$
$$= 0.940 - 0.693$$
$$= 0.247 \text{ bits}$$

$$gain(temperature) = info(T) - info_{temperature}(T)$$
$$= 0.940 - 0.911$$
$$= 0.029 \text{ bits}$$

$$gain(humidity) = info(T) - info_{humidity}(T)$$
$$= 0.940 - 0.788$$
$$= 0.152 \text{ bits}$$

$$gain(windy) = info(T) - info_{windy}(T)$$
$$= 0.940 - 0.892$$
$$= 0.048 \text{ bits}$$

แอททริบิวต์ที่ให้ค่า gain สูงที่สุด คือ outlook ดังนั้นแอททริบิวต์ outlook จึงถูกเลือกเป็น root node ของ decision tree เนื่องจาก attribute outlook ยังไม่สามารถจัดกลุ่มข้อมูลให้เป็นคลาสเดียวกันทั้งหมด จึงต้องสร้าง decision tree ต่อไป โดยพิจารณาเลือกแอททริบิวต์ที่จะมาเป็นโหนดในระดับที่ 2 ต่อจาก root node ในกรณี outlook = overcast ไม่จำเป็นต้องสร้างโหนดเพิ่มเติม เนื่องจากสามารถจัดกลุ่มข้อมูลที่เป็นคลาส yes ได้ทั้งหมด



แอททริบิวต์ที่สามารถถูกเลือกเป็นโหนดในระดับที่ 2 ประกอบด้วย temperature, humidity และ windy (แอททริบิวต์ outlook จะไม่ถูกใช้อีก เพราะสภาพอากาศ จะไม่มีโอกาสเกิดเหตุการณ์ outlook = sunny AND outlook = rainy)

พิจารณาการสร้างโหนดลูกทางด้านซ้ายมือ (outlook = sunny) ถ้าเลือกแอททริบิวต์ temperature จะคำนวณค่า gain ได้ดังนี้

gain (temperature) = info (outlook = sunny) – $info_{temperature}$ (outlook = sunny)

เนื่องจาก outlook = sunny จัดกลุ่มข้อมูลที่เป็นคลาส yes 2 เรคคอร์ด และข้อมูลที่เป็นคลาส no 3 เรคคอร์ด ดังนั้น

$$\text{info (outlook = sunny)} = -\frac{2}{5}\times\log_2\left(\frac{2}{5}\right)-\frac{3}{5}\times\log_2\left(\frac{3}{5}\right)$$

$$= 0.971 \text{ bits}$$

$$info_{temperature}(\text{outlook = sunny}) = \text{info ([0,2] , [1, 1] , [1, 0])}$$

$$= \frac{2}{5}\times\left[-\frac{0}{2}\times\log_2\left(\frac{0}{2}\right)-\frac{2}{2}\times\log_2\left(\frac{2}{2}\right)\right] + \frac{2}{5}\times\left[-\frac{1}{2}\times\log_2\left(\frac{1}{2}\right)-\frac{1}{2}\times\log_2\left(\frac{1}{2}\right)\right] + \frac{1}{5}\times\left[-\frac{1}{1}\times\log_2\left(\frac{1}{1}\right)-\frac{0}{1}\times\log_2\left(\frac{0}{1}\right)\right]$$

$$= 0.4 \text{ bits}$$

$\therefore$ gain (temperature) = 0.971 – 0.4 bits

= 0.571 bits

เมื่อ outlook = sunny แล้วทดลองจำแนกกลุ่มข้อมูลต่อไปนี้ด้วยแอททริบิวต์ attribute humidity และ windy

gain (humidity) = info (outlook = sunny) – $info_{humidity}$ (outlook = sunny)

= 0.971 – info ([0,3] , [2, 0])

= 0.971 – 0 bits

= 0.971 bits

gain (windy) = info (outlook = sunny) – $info_{windy}$ (outlook = sunny)

= 0.971 – info ([1,2] , [1, 1])

= 0.971 – 0.951 bits

= 0.020 bits

โดยสรุปแล้วการทดลองสร้าง decision tree ต่อด้วยแอททริบิวต์อื่น ๆ จะได้ค่า gain ดังนี้

gain (temperature) = 0.571 bits

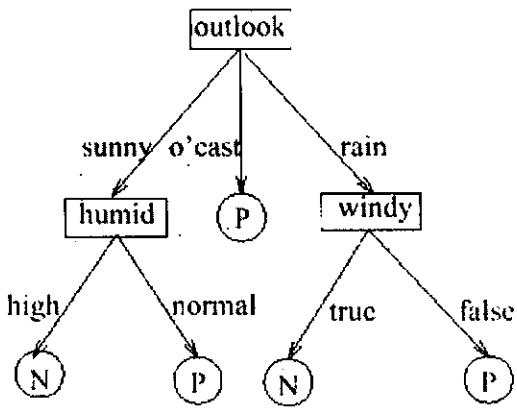
gain (humidity) = 0.971 bits

gain (windy) = 0.020 bits

จึงพิจารณาเลือกแอททริบิวต์ humidity เป็นโหนดในระดับที่สองต่อจากโหนด outlook

decision tree ยังเหลือโหนดลูกทางขวาของโหนด outlook ที่ต้องพิจารณาเลือกแอททริบิวต์ และจากวิธีการคำนวณค่า gain ที่แสดงด้วยตัวอย่างก่อนหน้านี้ สามารถเลือกได้ว่าแอททริบิวต์ windy จะให้ค่า gain ที่สูงที่สุด (จากกลุ่มของแอททริบิวต์ temperature, humidity และ windy)

กระบวนการสร้าง decision tree จะสิ้นสุดเมื่อ leaf nodes เป็นกลุ่มของข้อมูลคลาสเดียวกันทั้งหมด และภาพของ tree ที่ได้จะเป็นดังรูปที่ 2.7



รูปที่ 2.7 ภาพ decision tree ที่ได้จากข้อมูลการเล่นกอล์ฟ

decision tree สามารถแปลงเป็นกฎแบบมีเงื่อนไขได้ดังนี้

rule 1 : IF (outlook = sunny) AND (humidity = high) THEN play = no

rule 2 : IF (outlook = sunny) AND (humidity = normal) THEN play = yes

rule 3 : IF (outlook = overcast) THEN play = yes

rule 4 : IF (outlook = rainy) AND (windy = false) THEN play = yes

rule 5 : IF (outlook = rainy) AND (windy = true) THEN play = no

ในกรณีที่มีข้อมูลใหม่ที่ยังไม่ทราบคลาส

"outlook = sunny, temperature = cool, humidity = high, windy = true"

สามารถใช้ decision tree ทำนายคลาสของข้อมูลนี้ว่า play = no โดยพิจารณาจากเพียงสองแอททริบิวต์ คือ outlook = sunny และ humidity = high

2.3.3 Statistical learner

วิธีการสร้างโมเดลเพื่อการจำแนกข้อมูลโดยใช้หลักการทางสถิติ เป็นการใช้ทฤษฎีความน่าจะเป็นของเบส์ (Bayes' theorem) วิธีนี้ตั้งสมมติฐานว่าทุกแอททริบิวต์ในข้อมูลฝึกมีความเป็นอิสระจากกัน ค่าที่เกิดขึ้นในแอททริบิวต์หนึ่งจะไม่มีผลต่อค่าในแอททริบิวต์อื่น จึงเรียกวิธีการนี้ว่า การใช้ทฤษฎีเบส์อย่างง่าย (naïve Bayes method)

วิธีการนี้ใช้การแจกแจงค่าแอททริบิวต์แต่ละค่าที่เป็นไปได้ และนับว่าในแต่ละค่านั้นอยู่ในคลาสใด คิดเป็นสัดส่วนเท่าใด ตัวอย่างเช่น ในตารางที่ 2.1 ข้อมูลฝึกมีข้อมูลทั้งหมด 14 เรคคอร์ด อยู่ในคลาสของ play = yes รวม 9 เรคคอร์ด และอยู่ในคลาสของ play = no รวม 5 เรคคอร์ด เมื่อพิจารณา attribute outlook

outlook = sunny ปรากฏ 5 เรคคอร์ด ในจำนวนนี้เป็นคลาสของ play = yes 2 เรคคอร์ด คลาสของ play = no 3 เรคคอร์ด

คิดเป็นสัดส่วนของคลาส yes = 2/9

คิดเป็นสัดส่วนของคลาส no = 3/5

outlook = overcast ปรากฏ 4 เรคคอร์ด

คิดเป็นสัดส่วนของคลาส yes = 4/9

คิดเป็นสัดส่วนของคลาส no = 0/5

outlook = rainy ปรากฏ 5 เรคคอร์ด

คิดเป็นสัดส่วนของคลาส yes = 3/9

คิดเป็นสัดส่วนของคลาส no = 2/5

ค่าสัดส่วนแต่ละคลาสของทุกแอททริบิวต์จะเป็นตัว classifier ที่ใช้ในการจำแนกคลาสของข้อมูล จากข้อมูลการเล่นกอล์ฟในตารางที่ 2.1 จะได้ classifier ดังรูปที่ 2.8

	Outlook			Temperature			Humidity			Windy		Play	
	yes	no		yes	no		yes	no		yes	no	yes	no
sunny	2	3	hot	2	2	high	3	4	false	6	2	9	5
overcast	4	0	mild	4	2	normal	6	1	true	3	3		
rainy	3	2	cool	3	1								
sunny	2/9	3/5	hot	2/9	2/5	high	3/9	4/5	false	6/9	2/5	9/14	5/14
overcast	4/9	0/5	mild	4/9	2/5	normal	6/9	1/5	true	3/9	3/5		
rainy	3/9	2/5	cool	3/9	1/5								

รูปที่ 2.8 classifierที่ได้จากอัลกอริทึม naive Bayes

ถ้าการพยากรณ์อากาศในสัปดาห์หน้าเป็นดังนี้

"outlook = sunny, temperative = cool, humidity = high, windy = true"

ใช้ Bayes classifier ตามรูปที่ 2.8 คำนวณความน่าจะเป็นว่า play = yes หรือ play = no ได้ดังนี้

ความน่าจะเป็นกรณี play = yes คือ

2/9 x 3/9 x 3/9 x 3/9 x 9/14 = 0.0053

ความน่าจะเป็นที่ play = yes เมื่อ outlook = sunny

ความน่าจะเป็นที่ play = yes เมื่อ humidity = high

ความน่าจะเป็นของคลาส play = yes

ความน่าจะเป็นที่ play = yes เมื่อ temperature = cool

ความน่าจะเป็นที่ play = yes เมื่อ windy = true

ความน่าจะเป็นกรณี play = no คือ 3/5 x 1/5 x 4/5 x 3/5 x 5/14 = 0.0206

ดังนั้นจากข้อมูลพยากรณ์อากาศข้างต้นจึงสรุปว่า play = no

กรณีที่ข้อมูลไม่ครบถ้วน (missing values) Bayes classifier ยังคงใช้ทำนายคลาสของข้อมูลได้ โดยตัดค่าความน่าจะเป็นของแอททริบิวต์นั้นออกไป เช่น ถ้าข้อมูลพยากรณ์อากาศเป็นดังต่อไปนี้

"temperature = cool, humidity = high, windy = true"

ความน่าจะเป็นกรณี play = yes คือ 3/9 x 3/9 x 3/9 x 9/14 = 0.0238

ความน่าจะเป็นกรณี play = no คือ 1/5 x 4/5 x 3/5 x 5/14 = 0.0343

จึงสรุปว่าข้อมูลพยากรณ์อากาศข้างต้น ทำนายได้ว่า play = no

ในกรณีที่ข้อมูลบางแอททริบิวต์เป็นค่าตัวเลข เช่น ค่าอุณหภูมิ และค่าความชื้นสัมพัทธ์ ดังในตารางที่ 2.2 วิธีการของเบส์ จะใช้การหาค่าเฉลี่ย และค่าความแปรปรวน ของทั้งกรณีที่ play = yes และ play = no ดังนั้น Bayes classifier จะเป็นดังรูปที่ 2.9

ตารางที่ 2.2 ข้อมูลที่ใช้ประกอบการตัดสินใจเล่นกอล์ฟที่บางแอททริบิวต์เป็นตัวเลข

No.	Attributes				Play
	Outlook	Temperature	Humidity	Windy	
1	sunny	85	85	false	N
2	sunny	80	90	true	N
3	overcast	83	86	false	P
4	rainy	70	96	false	P
5	rainy	68	80	false	P
6	rainy	65	70	true	N
7	overcast	64	65	true	P
8	sunny	72	95	false	N
9	sunny	69	70	false	P
10	rainy	75	80	false	P
11	sunny	75	70	true	P
12	overcast	72	90	true	P
13	overcast	81	75	false	P
14	rainy	71	91	true	N

	Outlook			Temperature			Humidity			Windy			Play	
	yes	no		yes	no		yes	no		yes	no		yes	no
sunny	2	3		83	85		86	85	false	6	2		9	5
overcast	4	0		70	80		96	90	true	3	3			
rainy	3	2		68	65		80	70						
				64	72		65	95						
				69	71		70	91						
				75			80							
				75			70							
				72			90							
				81			75							
sunny	2/9	3/5	mean	73	74.6	mean	79.1	86.2	false	6/9	2/5		9/14	5/14
overcast	4/9	0/5	SD	6.2	7.9	SD	10.2	9.7	true	3/9	3/5			
rainy	3/9	2/5												

รูปที่ 2.9 Bayes classifier ในกรณีที่ข้อมูลเป็นตัวเลข

การทำนายคลาสของข้อมูลในแอททริบิวต์ที่เป็นตัวเลข เช่น พยากรณ์อากาศให้ข้อมูลว่า outlook = sunny, temperature = 66, humidiyt = 90, windy = true จะใช้ค่าเฉลี่ยและค่าความแปรปรวนไปคำนวณหา probability density function (f(x))

$$f(x) = (1/(SD * \sqrt{2\pi}) * e^{-(x-mean)(x-mean)/2*SD*SD}$$

เช่น ถ้าค่าอุณหภูมิ = 66 องศาฟาเรนไฮต์

$$f(temperature = 66 \mid play = yes) = \left(1/\left(6.2 \times \sqrt{2\pi}\right)\right) e^{-(66-73)(66-73)/2\times6.2\times6.2}$$

$$= 0.0340$$

$$f(temperature = 66 \mid play = no) = \left(1/\left(7.9\sqrt{2\pi}\right)\right) e^{-(66-74.6)(66-74.6)/2\times7.9\times7.9}$$

$$= 0.0291$$

และถ้า humidity = 90 สามารถคำนวณ probability density function ด้วยสูตรเดียวกันข้างต้น แล้วนำค่าที่ได้ไปเป็นตัวคูณ เพื่อคำนวณความน่าจะเป็นกรณี play = yes และความน่าจะเป็นกรณี play = no

ความน่าจะเป็น play = yes คือ 2/9 * 0.0340 * 0.0221 * 3/9 * 9/14 = 0.000036

ความน่าจะเป็น play = no คือ 3/5 * 0.0291 * 0.0380 * 3/5 * 5/14 = 0.000136

ดังนั้นตามสภาพอากาศที่มีการพยากรณ์ว่า outlook = sunny, temperature = 66, humidiyt = 90, windy = true ทำนายได้ว่านักกอล์ฟจะตัดสินใจไม่ออกไปเล่นกอล์ฟ

2.3.4 Instance-based learner

อัลกอริทึมการเรียนรู้เพื่อสร้างโมเดลจำแนกประเภทข้อมูลในกลุ่มนี้ จะแตกต่างจากในสามกลุ่มข้างต้นตรงที่ไม่มีการสร้างโมเดลหรือ classifier ไว้ก่อนที่จะนำโมเดลไปใช้ทำนายข้อมูลใหม่ที่เกิดขึ้น แต่จะใช้วิธีการเก็บข้อมูลฝึกไว้แล้วนำมาคำนวณคลาสเมื่อต้องการทำนายข้อมูลใหม่ บางครั้งจึงเรียกอัลกอริทึมแบบนี้ว่า lazy evaluation เวลาที่ใช้ในขั้นตอนการสร้างโมเดลจะสั้นมากเนื่องจากจะเป็นเพียงการเก็บข้อมูลให้อยู่ในรูปแบบที่เหมาะสมเท่านั้น แต่จะใช้เวลานานในการพยากรณ์คลาสของข้อมูลใหม่ ซึ่งตรงกันข้ามกับอัลกอริทึมในสามกลุ่มข้างต้นที่จัดเป็น eager evaluation ที่ใช้เวลาส่วนใหญ่ในการสร้าง classifier แต่ในขั้นตอนการทำนายข้อมูลใหม่จะสั้นมาก

การทำนายคลาสของอัลกอริทึม instance-based จะใช้การเปรียบเทียบข้อมูลใหม่กับข้อมูลเดิมที่มีอยู่และเป็นข้อมูลที่มีการกำหนดคลาสไว้แล้ว ข้อมูลที่มีอยู่รายการใดลักษณะใกล้เคียงกับข้อมูลใหม่มากที่สุด ก็จะทำนายประเภทของข้อมูลใหม่ตามประเภทข้อมูลที่มีอยู่ อัลกอริทึมประเภทนี้บางครั้งเรียกชื่อว่า k-nearest neighbor โดย k เป็นตัวเลขจำนวนเต็ม ถ้า k เป็น 1 หมายถึงการทำนายประเภทของข้อมูลใหม่ จะใช้ข้อมูลที่มีอยู่เพียงรายการเดียวที่มีลักษณะใกล้เคียงกับข้อมูลใหม่

มากที่สุด ในงานวิจัยนี้ใช้อัลกอริทึม 10-nearest neighbor นั่นคือจะใช้ข้อมูล 10 รายการที่ใกล้เคียงกับข้อมูลใหม่มากที่สุดในการทำนายประเภทข้อมูล

การวัดความใกล้เคียงของข้อมูลที่มีอยู่กับข้อมูลใหม่ที่ต้องการทำนายประเภท จะใช้เกณฑ์ Euclidean distance ซึ่งมีวิธีคำนวณค่าดังนี้

$$\text{Distance}(X,Y) = \sqrt{\sum_{i=1 \text{ to } n} (X_i - Y_i)^2}$$

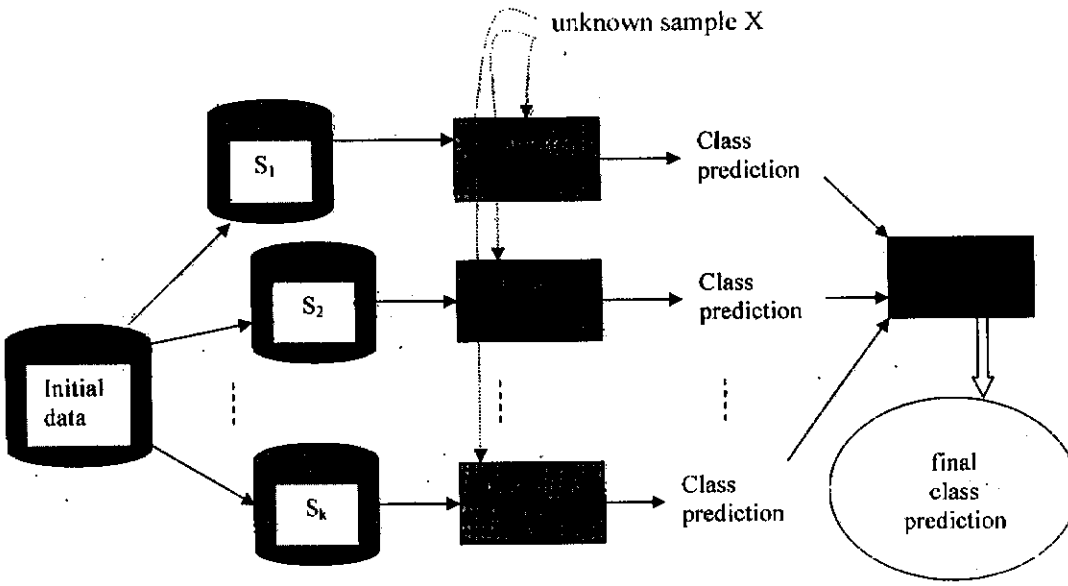
โดยข้อมูลรายการ X จะประกอบด้วยลักษณะ (attribute) ต่างๆ คือ $(X_1, X_2, ...,X_n)$ และข้อมูลรายการ Y จะประกอบด้วยลักษณะ $(Y_1, Y_2, ..., Y_n)$

2.3.5 เทคนิคที่ใช้เพิ่มประสิทธิภาพ classifier

การเพิ่มประสิทธิภาพของ classifier เป็นการเพิ่มความถูกต้องเที่ยงตรงของการทำนายข้อมูล ซึ่งทำได้ด้วยการใช้วิธี classification หลายครั้ง หรือเรียกว่า multiple learning เทคนิคที่นิยมใช้ได้แก่ bagging และ boosting

Bagging

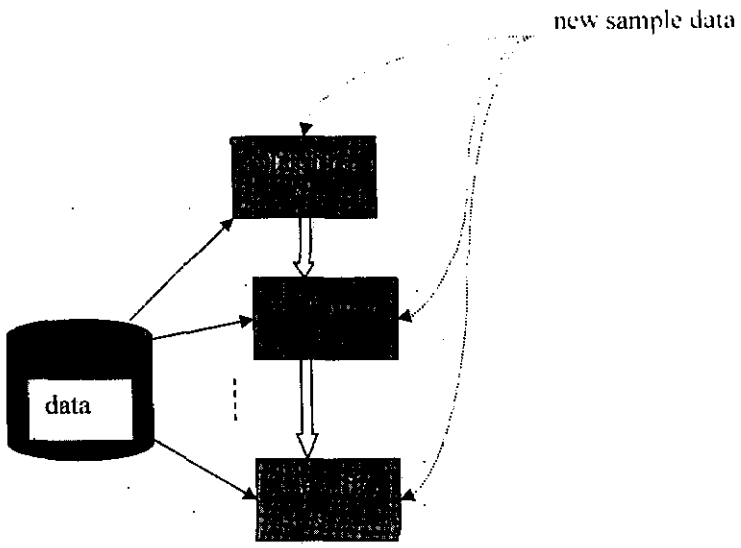
เทคนิค bagging หรือ bootstrap aggregation เป็นการทำ classification หลายครั้ง เช่น 10 ครั้ง ทำให้ได้ classifier 10 โมเดล เมื่อจะนำทั้ง 10 โมเดลไปใช้เพื่อการทำนายหรือจำแนกประเภทข้อมูลใหม่ จะส่งข้อมูลใหม่รายการนั้นไปให้ทั้ง 10 โมเดลทำนายคลาส ผลที่ได้จากการทำนายของทั้ง 10 โมเดลอาจจะเหมือนกันทั้งหมด หรืออาจจะมีการทำนายคลาสแตกต่างกันก็ได้ ผลลัพธ์สุดท้ายของการทำนายจะใช้วิธีนับโหวตว่าทั้ง10 โมเดลนั้น ส่วนใหญ่ทำนายว่าเป็นคลาสใด จะแสดงผลคลาสที่เป็นการทำนายของโมเดลส่วนใหญ่ วิธีการ bagging แสดงขั้นตอนการทำงานได้ดังรูปที่ 2.10



รูปที่ 2.10 เทคนิค Bagging เพื่อเพิ่มประสิทธิภาพการจำแนก

Boosting

เทคนิค boosting เป็นการทำ classification หลายครั้งเหมือนกับเทคนิค bagging แต่ต่างกันตรงที่การสร้าง classifier แต่ละครั้งไม่ได้เป็นอิสระต่อกัน เทคนิค boosting จะเริ่มต้นทำงานด้วยการให้ค่าน้ำหนักกับข้อมูลแต่ละเรคคอร์ด โดยเริ่มต้นทุกเรคคอร์ดมีน้ำหนักเป็น 1 เท่ากัน เมื่อทำ classification ครั้งแรกและทดสอบ classifier ด้วยข้อมูลทดสอบ เรคคอร์ดใดที่ classifier ทำนายผิด จะถูกเพิ่มค่าน้ำหนักเพื่อให้การทำ classification ครั้งต่อไปให้ความสนใจกับข้อมูลที่ถูกทำนายผิดมากกว่าข้อมูลอื่น การทำ classification จะดำเนินไปเช่นนี้หลายครั้งจนกระทั่งได้ classifier ที่มีความแม่นตรงสูงถึงเกณฑ์ที่กำหนด หรือเมื่อไม่สามารถเพิ่มความแม่นตรงได้อีกต่อไป กระบวนการจึงสิ้นสุด และ classifier ล่าสุดที่ได้ (ตามรูปที่ 2.11 คือ classifier C^*) จะถูกนำไปใช้ในการทำนายคลาสของข้อมูลที่จะเกิดขึ้นในอนาคต



รูปที่ 2.11 เทคนิค Boosting เพื่อเพิ่มประสิทธิภาพการจำแนก

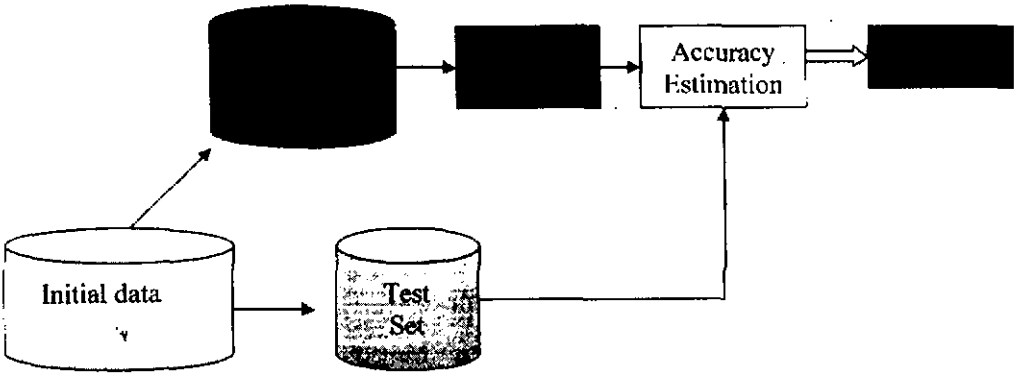
2.4 วิธีการวิเคราะห์ความแม่นตรงของโมเดล

การทำเหมืองข้อมูลประเภท classification เป็นการสร้าง classifier เพื่อใช้เป็นโมเดลในการจำแนกหรือทำนายประเภท (class) ของข้อมูลในอนาคต classifier ที่ใช้ประโยชน์ได้ดีจะต้องมีความแม่นตรง (accurate) ในการทำนายสูง คุณสมบัติความแม่นตรงนี้ยังสามารถใช้เป็นเกณฑ์ในการเปรียบเทียบประสิทธิภาพของอัลกอริทึมต่างๆ ที่ทำหน้าที่สร้าง classifier วิธีการที่ใช้วัดความแม่นตรงของ classifier มีหลายวิธีดังนี้

<u>วิธี Holdout</u>

วิธีการนี้จะแบ่งข้อมูลออกเป็นสองส่วน ส่วนแรกเรียกว่า ข้อมูลฝึก (training data) จะมีจำนวนประมาณสองในสาม หรือประมาณ 66% ของข้อมูลทั้งหมด ส่วนที่สองเรียกว่า ข้อมูลทดสอบ (test data) มีจำนวนประมาณหนึ่งในสาม หรือ 34% ของข้อมูลทั้งหมด

ข้อมูลฝึกจะถูกส่งเป็น input ให้อัลกอริทึม classification เพื่อใช้สร้างโมเดลของข้อมูลที่เรียกว่า classifier จากนั้นจะใช้ข้อมูลทดสอบวัดความถูกต้องในการจำแนกคลาสของ classifier วิธีการวัดความแม่นตรงนี้แสดงเป็นแผนภาพได้ดังรูปที่ 2.12 *วิธี holdout นี้จะเหมาะสมกับกรณีที่* ข้อมูลมีจำนวนมาก (เช่นจำนวนเรคคอร์ดมากกว่า 1,000 เรคคอร์ด)



รูปที่ 2.12 การวัดความแม่นตรงแบบ Holdout

วิธี k-fold cross-validation

วิธีการวัดประสิทธิภาพของ classifier แบบนี้ จะแบ่งข้อมูลออกเป็น k ส่วน และจะวัดประสิทธิภาพ k รอบ (k เป็นเลขจำนวนเต็ม เช่น 5, 10, 24)

รอบที่ 1 จะใช้ข้อมูลส่วนที่ 1 เป็นข้อมูลทดสอบ ข้อมูลส่วนที่ 2 ถึงส่วนที่ k ถูกใช้เป็นข้อมูลฝึก ผลการวัดประสิทธิภาพ จะได้ค่า accuracy#1

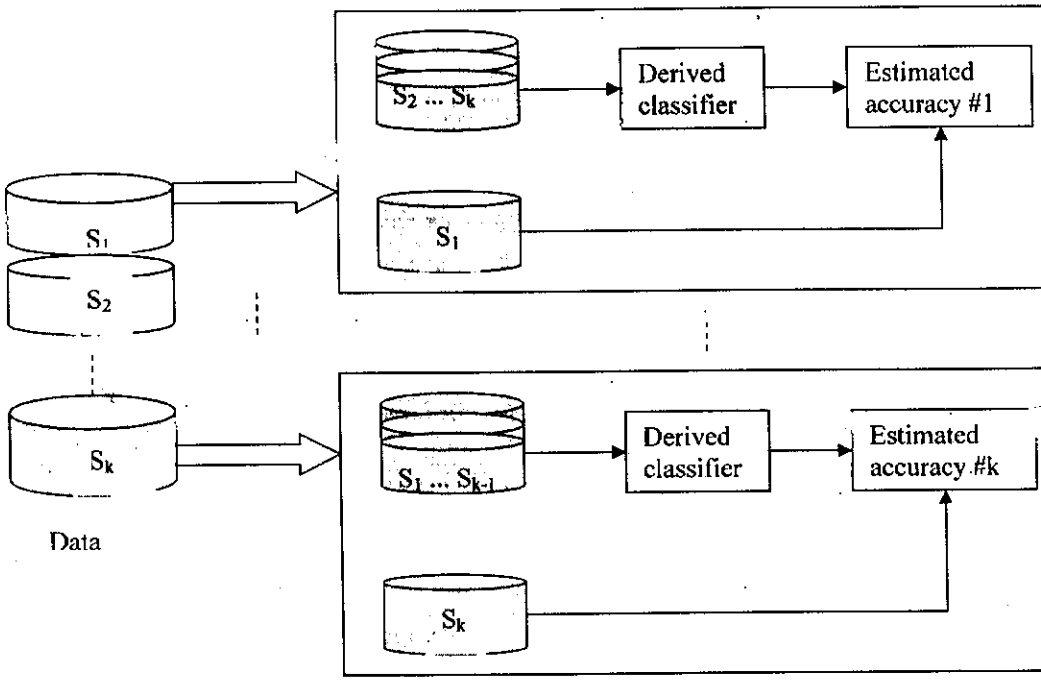
รอบที่ 2 จะใช้ข้อมูลส่วนที่ 2 เป็นข้อมูลทดสอบ ข้อมูลส่วนที่ 1 และข้อมูลส่วนที่ 3 ถึงส่วนที่ k ถูกใช้เป็นข้อมูลฝึก ผลการวัดประสิทธิภาพ จะได้ค่า accuracy#2

...

รอบที่ k จะใช้ข้อมูลส่วนที่ k เป็นข้อมูลทดสอบ ข้อมูลส่วนที่ 1 ถึงส่วนที่ k-1 ถูกใช้เป็นข้อมูลฝึก ผลการวัดประสิทธิภาพ จะได้ค่า accuracy#k

ค่าความแม่นตรงของ classifier จะเป็นค่าเฉลี่ยของ accuracy#1, accuracy#2, ..., accuracy#k

วิธีการนี้แสดงเป็นแผนภาพได้ดังรูปที่ 2.13 วิธีวัดประสิทธิภาพ classifier แบบนี้จะเหมาะกับกรณีข้อมูลมีน้อย



รูปที่ 2.13 การวัดความแม่นตรงแบบ k-fold cross-validation

ในกรณีที่การแบ่งส่วนของข้อมูลแบ่งจำนวนส่วนเท่ากับจำนวนข้อมูล เช่น ข้อมูลมี 50 เรคคอร์ด แบ่งข้อมูลเป็น 50 ส่วน เพื่อทำการทดสอบ 50-fold cross-validation จะเรียกการทดสอบนี้ได้อีกชื่อหนึ่งว่า leave-one-out ซึ่งจะใช้ในกรณีที่ข้อมูลมีน้อยมาก

วิธี stratified cross-validation

วิธีการทดสอบประสิทธิภาพแบบนี้ปรับปรุงเพิ่มเติมขึ้นมาจากวิธี k-fold cross-validation โดยให้ข้อมูลที่แบ่งออกเป็น k ส่วน แต่ละส่วนมีข้อมูลครบทุกคลาสด้วยสัดส่วนเดียวกับข้อมูลตั้งต้น ตัวอย่างเช่น ถ้าข้อมูลตั้งต้นมี 1,000 เรคคอร์ด ในจำนวนนี้เป็นคลาส A 600 เรคคอร์ด และคลาส B 400 เรคคอร์ด เมื่อแบ่งข้อมูลออกเป็น 10 ส่วน (นั่นคือ k = 10) แต่ละส่วนจะมีข้อมูล 100 เรคคอร์ด และในจำนวนนี้ 60 เรคคอร์ดเป็นข้อมูลคลาส A และ 40 เรคคอร์ดเป็นข้อมูลคลาส B

ในโครงการวิจัยนี้จะทดสอบเปรียบเทียบประสิทธิภาพของอัลกอริทึมสร้าง classifier ด้วยวิธีการ stratified 10-fold cross-validation ที่เป็นการผสมผสานวิธีการ k-fold cross-validation และวิธี stratified cross-validation เพื่อให้ได้ผลการทดสอบที่เชื่อถือได้มากที่สุด

บทที่ 3
วิธีดำเนินการวิจัย

โครงการวิจัยนี้มีจุดมุ่งหมายที่จะทดสอบประสิทธิภาพของอัลกอริทึมต่างๆ ที่ใช้ในการทำเหมืองข้อมูลประเภทสังเคราะห์โมเดลจำแนกข้อมูล เพื่อค้นหาอัลกอริทึมที่เหมาะสมที่สุดสำหรับอมูลการวินิจฉัยโรคทางการแพทย์ รายละเอียดเนื้อหาในบทนี้ประกอบด้วยระเบียบวิธีวิจัย ปรากฏนหัวข้อ 3.1 คำอธิบายข้อมูลทั้ง 12 ชุดที่ใช้ในการทดสอบอัลกอริทึม ปรากฏในหัวข้อ 3.2 และัวข้อ 3.3 เป็นรายละเอียดวิธีการที่ใช้ในการวิเคราะห์เปรียบเทียบผลลัพธ์จากแต่ละอัลกอริทึม

3.1 ระเบียบวิธีวิจัย

การค้นคว้าวิจัยจะแบ่งออกเป็น 7 ขั้นตอนดังนี้

(1) ศึกษาและรวบรวมสรุปงานวิจัยที่เกี่ยวข้อง

(2) ศึกษาการใช้งานระบบ Weka ซึ่งเป็น open-source environment ที่ใช้ในการทำ data mining

(3) รวบรวมข้อมูลการวินิจฉัยโรคจำนวน 12 ชุดข้อมูล โดยข้อมูลส่วนใหญ่จะสืบค้นจาก UCI Repository และคัดเลือกเฉพาะข้อมูลทางการแพทย์

(4) แปลงรูปแบบแฟ้มข้อมูล ให้อยู่ในรูปแบบ arff เพื่อใช้กับระบบ Weka

(5) วิเคราะห์อัลกอริทึมสังเคราะห์ความรู้ เพื่อคัดเลือกอัลกอริทึมที่ใช้เทคนิคพื้นฐานแตกต่างกัน อัลกอริทึมที่คัดเลือกแล้วประกอบด้วย

- OneR จากกลุ่มอัลกอริทึมที่ใช้หลักการ Rule learner
- J48 จากกลุ่มอัลกอริทึมที่ใช้หลักการ Tree-based learner
- naive Bayes จากกลุ่มอัลกอริทึมที่ใช้หลักการ Statistical learner
- IB10 จากกลุ่มอัลกอริทึมที่ใช้หลักการ Instance-based learner
- Bagging เป็นอัลกอริทึมประเภท multiple learning ใช้เพิ่มประสิทธิภาพ classifier ด้วยเทคนิค bagging โดยในการวิจัยนี้จะทดสอบเทคนิค bagging กับอัลกอริทึม OneR, J48, naive Bayes, และ IB10
- AdaBoost เป็นอัลกอริทึมประเภท multiple learning ใช้เพิ่มประสิทธิภาพ classifier ด้วยเทคนิค boosting โดยในการวิจัยนี้จะทดสอบเทคนิค boosting กับอัลกอริทึม OneR, J48, naive Bayes, และ IB10

(6) ทดสอบแต่ละอัลกอริทึมกับข้อมูลแต่ละชุดที่รวบรวมไว้ เพื่อบันทึกพฤติกรรมการสังเคราะห์ความรู้ของแต่ละอัลกอริทึม เครื่องคอมพิวเตอร์ที่ใช้ในการทดสอบเป็นคอมพิวเตอร์ PC Pentium III 700 MHz RAM 256 MB

(7) วิเคราะห์ผลและเสนอแนะลักษณะของอัลกอริทึมและเทคนิคที่เหมาะสมกับข้อมูลการวินิจฉัยโรค

3.2 แหล่งที่มาของข้อมูลและการจัดประเภทข้อมูล

ข้อมูลที่ใช้ในโครงการวิจัยนี้ได้มาจากแหล่งข้อมูลของมหาวิทยาลัยแห่งรัฐแคลิฟอร์เนีย เมืองเออร์ไวน์ (University of California at Irvine) (Blake, Keogh and Merz, 1998) โดยคัดเลือกมาเฉพาะข้อมูลทางการแพทย์ที่เกี่ยวข้องกับการวินิจฉัยโรค จำนวนชุดข้อมูลทั้งหมดมี 12 ชุด จำแนกได้เป็น 4 กลุ่มคือ

- ข้อมูลที่มีทั้งข้อความ, สัญลักษณ์, ตัวเลข ปะปนกัน และไม่มีข้อมูลส่วนใดสูญหาย
  (nominal data, no missing values)
  ได้แก่ข้อมูล Lymphography และ Post operative
- ข้อมูลที่มีทั้งข้อความ, สัญลักษณ์, ตัวเลข ปะปนกัน แต่มีข้อมูลบางส่วนสูญหาย
  (nominal data, missing values)
  ได้แก่ข้อมูล Primary tumor, Heart disease และ Breast cancer
- ข้อมูลที่เป็นตัวเลขทั้งหมด (ยกเว้นแอททริบิวต์ที่ระบุคลาส) และไม่มีข้อมูลส่วนใดสูญหาย
  (numeric data, no missing values)
  ได้แก่ข้อมูล Diabetes, Heart (Statlog), Thyroid และ Liver disorders
- ข้อมูลที่เป็นตัวเลขทั้งหมด (ยกเว้นแอททริบิวต์ที่ระบุคลาส) แต่มีข้อมูลบางส่วนสูญหาย
  (numeric data, missing values)
  ได้แก่ข้อมูล Wisconsin breast cancer, Hepatitis และ Lungcancer

รายชื่อชุดข้อมูล จำนวนแอททริบิวต์(หรือ ฟิลด์) และจำนวนคลาสในแต่ละชุดข้อมูล สรุปได้ดังตารางที่ 3.1

ตารางที่ 3.1 ชุดข้อมูลและรายละเอียดของแอททริบิวต์โดยสรุป

ชื่อข้อมูล	จำนวนข้อมูล	จำนวนแอททริบิวต์	จำนวนแอททริบิวต์จำแนกตามลักษณะ				จำนวนคลาส
			ค่าไบนารี่	ข้อความ	ตัวเลข	การสูญหาย	
1. Lymphography	148	19	9	6	3	No	4
2. Post operative	87	9	-	7	1	No	3
3. Primary tumor	339	18	13	4	-	Yes	22
4. Heart disease	597	14	2	5	6	Yes	2
5. Breast cancer	286	10	-	9	-	Yes	2
6. Diabetes	768	9	-	-	8	No	2
7. Heart (Statlog)	270	14	-	-	13	No	2
8. Thyroid	215	6	-	-	5	No	3
9. Liver disorder	345	7	-	-	6	No	2
10.Wisconsin breast cancer	699	10	-	-	9	Yes	2
11.Hepatitis	155	20	-	-	19	Yes	2
12.Lung cancer	32	57	-	-	56	Yes	4

ข้อมูลแต่ละชุดมีการจำแนกคลาสของข้อมูลดังนี้

(1) ข้อมูล Lymphography เป็นข้อมูลการวินิจฉัยต่อมน้ำเหลือง

มีจำนวนข้อมูล 148 เรคคอร์ด จำแนกออกเป็น 4 คลาส คือ

normal find (มีจำนวน 2 เรคคอร์ด)

metastases (มีจำนวน 81 เรคคอร์ด)

malignant lymph (มีจำนวน 61 เรคคอร์ด)

fibrosis (มีจำนวน 4 เรคคอร์ด)

(2) ข้อมูล Post operative เป็นข้อมูลการวินิจฉัยสภาพคนไข้หลังการผ่าตัด

มีจำนวนข้อมูล 87 เรคคอร์ด จำแนกออกเป็น 3 คลาส คือ

I หมายถึง คนไข้ถูกส่งไปห้องไอซียู (มีจำนวน 1 เรคคอร์ด)

S หมายถึง คนไข้ถูกส่งกลับบ้าน (มีจำนวน 24 เรคคอร์ด)

A หมายถึง คนไข้ถูกส่งไปหอผู้ป่วย (มีจำนวน 62 เรคคอร์ด)

(3) ข้อมูล Primary tumor เป็นข้อมูลการวินิจฉัยโรคมะเร็งที่เกิดกับอวัยวะต่างๆ

มีจำนวนข้อมูล 339 เรคคอร์ด จำแนกออกเป็น 22 คลาส คือ

lung (มีจำนวน 84 เรคคอร์ด)

head and neck (มีจำนวน 20 เรคคอร์ด)

esophagus	(มีจำนวน 9 เรคคอร์ด)
thyroid	(มีจำนวน 14 เรคคอร์ด)
stomach	(มีจำนวน 39 เรคคอร์ด)
duodenum and small intestine	(มีจำนวน 1 เรคคอร์ด)
colon	(มีจำนวน 14 เรคคอร์ด)
rectum	(มีจำนวน 6 เรคคอร์ด)
anus	(มีจำนวน 0 เรคคอร์ด)
salivary glands	(มีจำนวน 2 เรคคอร์ด)
pancreas	(มีจำนวน 28 เรคคอร์ด)
gall bladder	(มีจำนวน 16 เรคคอร์ด)
liver	(มีจำนวน 7 เรคคอร์ด)
kidney	(มีจำนวน 24 เรคคอร์ด)
bladder	(มีจำนวน 2 เรคคอร์ด)
testis	(มีจำนวน 1 เรคคอร์ด)
prostate	(มีจำนวน 10 เรคคอร์ด)
ovary	(มีจำนวน 29 เรคคอร์ด)
corpus uteri	(มีจำนวน 6 เรคคอร์ด)
cervix uteri	(มีจำนวน 2 เรคคอร์ด)
vagina	(มีจำนวน 1 เรคคอร์ด)
breast	(มีจำนวน 24 เรคคอร์ด)

(4) ข้อมูล Heart disease เป็นข้อมูลการวินิจฉัยอาการหลอดเลือดเลี้ยงหัวใจตีบโดยใช้ภาพเอ็กซเรย์ มีจำนวนข้อมูล 597 เรคคอร์ด จำแนกออกเป็น 2 คลาส คือ

< 50% diameter narrowing	(มีจำนวน 353 เรคคอร์ด)
≥ 50% diameter narrowing	(มีจำนวน 244 เรคคอร์ด)

(5) ข้อมูล Breast cancer เป็นข้อมูลการวินิจฉัยการเกิดขึ้นใหม่ของมะเร็งเต้านม มีจำนวนข้อมูล 286 เรคคอร์ด จำแนกออกเป็น 2 คลาส คือ

no recurrence	(มีจำนวน 201 เรคคอร์ด)
recurrence	(มีจำนวน 85 เรคคอร์ด)

(6) ข้อมูล Diabetes เป็นข้อมูลการทดสอบว่าคนไข้ส่ออาการโรคเบาหวานหรือไม่ โดยใช้มาตรฐานขององค์การอนามัยโลก ทดสอบกับคนไข้เพศหญิงที่เป็นชนเผ่าพื้นเมืองอินเดียนแดง รัฐอริโซน่า มีจำนวนข้อมูล 768 เรคคอร์ด จำแนกออกเป็น 2 คลาส คือ

positive (มีจำนวน 268 เรคคอร์ด)

negative (มีจำนวน 500 เรคคอร์ด)

(7) ข้อมูล Heart Disease (Statlog) เป็นข้อมูลการทดสอบว่าคนไข้มีอาการของโรคหัวใจหรือไม่
มีจำนวนข้อมูล 270 เรคคอร์ด จำแนกออกเป็น 2 คลาส คือ

absent of heart disease (มีจำนวน 150 เรคคอร์ด)

present of heart disease (มีจำนวน 120 เรคคอร์ด)

(8) ข้อมูล Thyroid gland เป็นข้อมูลการวินิจฉัยต่อมธัยรอยด์
มีจำนวนข้อมูล 215 เรคคอร์ด จำแนกออกเป็น 3 คลาส คือ

normal (มีจำนวน 150 เรคคอร์ด)

hyper-thyroid (มีจำนวน 35 เรคคอร์ด)

hypo-thyroid (มีจำนวน 30 เรคคอร์ด)

(9) ข้อมูล Liver disorders (ไม่ปรากฏรายละเอียดของข้อมูล)
มีจำนวนข้อมูล 345 เรคคอร์ด จำแนกออกเป็น 2 คลาส คือ

class 1 (มีจำนวน 145 เรคคอร์ด)

class 2 (มีจำนวน 200 เรคคอร์ด)

(10) ข้อมูล Wisconsin breast cancer เป็นการวินิจฉัยมะเร็งเต้านมว่าเป็นชนิดร้ายแรง (malignant) หรือไม่ร้ายแรง (benign)
มีจำนวนข้อมูล 699 เรคคอร์ด จำแนกออกเป็น 2 คลาส คือ

benign (มีจำนวน 458 เรคคอร์ด)

malignant (มีจำนวน 241 เรคคอร์ด)

(11) ข้อมูล Hepatitis (ไม่ปรากฏรายละเอียดของข้อมูล)
มีจำนวนข้อมูล 155 เรคคอร์ด จำแนกออกเป็น 2 คลาส คือ

die (มีจำนวน 32 เรคคอร์ด)

live (มีจำนวน 123 เรคคอร์ด)

(12) ข้อมูล Lung cancer (ไม่ปรากฏรายละเอียดของข้อมูล)
มีจำนวนข้อมูล 32 เรคคอร์ด จำแนกออกเป็น 3 คลาส คือ

class 1 (มีจำนวน 9 เรคคอร์ด)

class 2 (มีจำนวน 13 เรคคอร์ด)

class 3 (มีจำนวน 10 เรคคอร์ด)

ข้อมูลทั้ง 12 ชุดข้างต้น สามารถจัดกลุ่มได้หลายลักษณะ ดังนี้

แบ่งกลุ่มตามจำนวนคลาส จำแนกได้เป็น 2 กลุ่ม

กลุ่มที่ 1: ข้อมูลที่มีจำนวนคลาส = 2 (binary class)

- Breast cancer data
- Diabetes data
- Heart disease data (Statlog)
- Liver disorder data
- Wisconsin breast cancer data
- Hepatitis data
- Heart disease data

กลุ่มที่ 2: ข้อมูลที่มีจำนวนคลาส > 2 (multi-class)

- Lymphography data (4 classes)
- Post operative data (3 classes)
- Primary tumor data (22 classes)
- Thyroid data (3 classes)
- Lung cancer data (4 classes)

แบ่งกลุ่มตามจำนวนแอททริบิวต์ จำแนกได้เป็น 2 กลุ่ม

กลุ่มที่ 1: ข้อมูลที่มีจำนวนแอททริบิวต์ > 15 (high-dimensional data)

- Lymphography data (19 attributes)
- Primary tumor data (18 attributes)
- Hepatitis data (20 attributes)
- Lung cancer data (57 attributes)

กลุ่มที่ 2: ข้อมูลที่มีจำนวนแอททริบิวต์ $\leq$ 15

- Post operative data (9 attributes)
- Heart disease data (14 attributes)
- Breast cancer data (10 attributes)
- Diabetes data (9 attributes)
- Heart disease (Statlog) (14 attributes)
- Thyroid data (6 attributes)
- Liver disorder data (7 attributes)
- Wisconsin breast cancer (10 attributes)

<u>แบ่งกลุ่มตามความสมบูรณ์ของข้อมูล</u> จำแนกได้เป็น 2 กลุ่ม

กลุ่มที่ 1: ข้อมูลมี missing values

- Primary tumor data
- Heart disease data
- Breast cancer data
- Wisconsin breast cancer data
- Hepatitis data
- Lung cancer data

กลุ่มที่ 2: ข้อมูลไม่มี missing values

- Lymphography data
- Post operative data
- Diabetes data
- Heart disease (Statlog) data
- Thyroid data
- Liver disorder data

<u>แบ่งกลุ่มตามลักษณะแอททริบิวต์</u> จำแนกได้เป็น 3 กลุ่ม

กลุ่มที่ 1: ข้อมูลทั้งหมดเป็นตัวเลข (ยกเว้น goal attribute)

- Diabetes data
- Heart disease (Statlog) data
- Thyroid data
- Liver disorder data
- Wisconsin breast cancer data
- Hepatitis data
- Lung cancer data

กลุ่มที่ 2: ข้อมูลทั้งหมดเป็นข้อความหรือสัญลักษณ์

- Primary tumor data
- Breast cancer data

กลุ่มที่ 3: ข้อมูลผสมทั้งตัวเลขและข้อความ

- Lymphography data (15 nominal attributes, 3 numeric)
- Post operative data (7 nominal attributes, 1 numeric)
- Heart disease data (7 nominal attributes, 6 numeric)

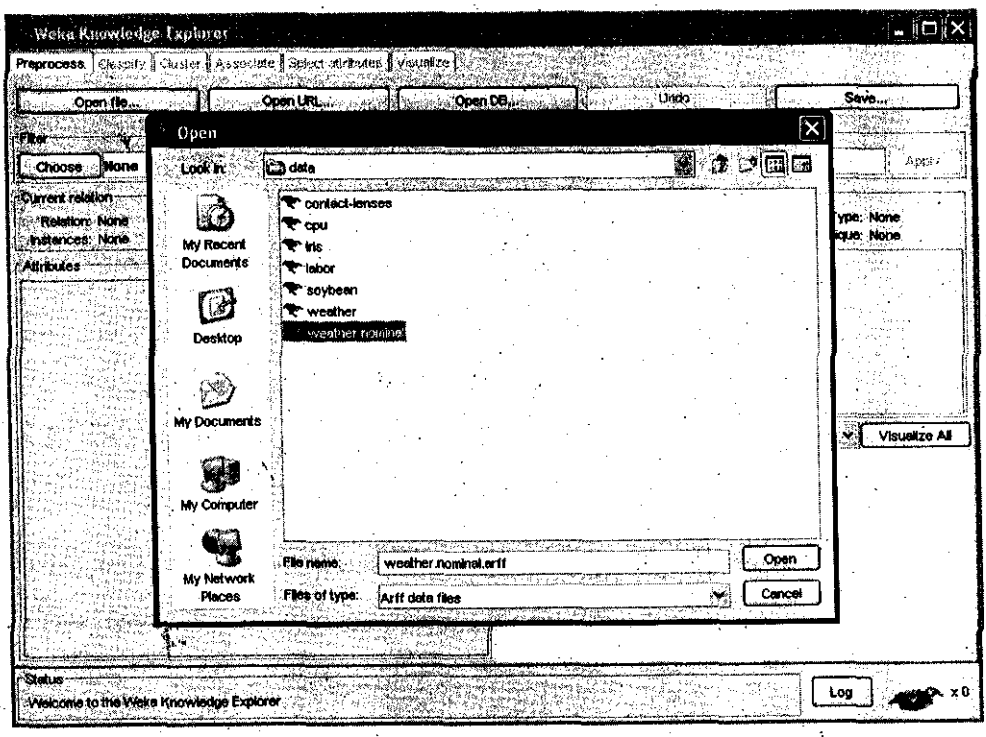
3.3 วิธีการทดสอบเปรียบเทียบอัลกอริทึมและเทคนิคการสังเคราะห์โมเดล

เครื่องมือที่ใช้ในการทดสอบอัลกอริทึมและเทคนิคต่างๆในการสังเคราะห์โมเดล คือ ระบบ WEKA (Waikato Environment for Knowledge Analysis) ซึ่งเป็นซอฟต์แวร์ที่เผยแพร่ฟรีโดยมหาวิทยาลัย Waikato ประเทศนิวซีแลนด์ (http://www.cs.waikato.ac.nz/ml/weka/) โปรแกรม WEKA จะรับข้อมูลที่อยู่ในรูปแบบ ARFF (Attribute-Relation File Format) ซึ่งประกอบด้วยส่วนคำอธิบายข้อมูล และส่วนข้อมูล ดังตัวอย่างต่อไปนี้

```
% This is a toy example, the UCI weather dataset.
@relation weather.symbolic
@attribute outlook {sunny, overcast, rainy}
@attribute temperature {hot, mild, cool}
@attribute humidity {high, normal}
@attribute windy {TRUE, FALSE}
@attribute play {yes, no}
@data
sunny,hot,high,FALSE,no
sunny,hot,high,TRUE,no
overcast,hot,high,FALSE,yes
rainy,mild,high,FALSE,yes
rainy,cool,normal,FALSE,yes
rainy,cool,normal,TRUE,no
overcast,cool,normal,TRUE,yes
sunny,mild,high,FALSE,no
sunny,cool,normal,FALSE,yes
rainy,mild,normal,FALSE,yes
sunny,mild,normal,TRUE,yes
overcast,mild,high,TRUE,yes
overcast,hot,normal,FALSE,yes
rainy,mild,high,TRUE,no
```

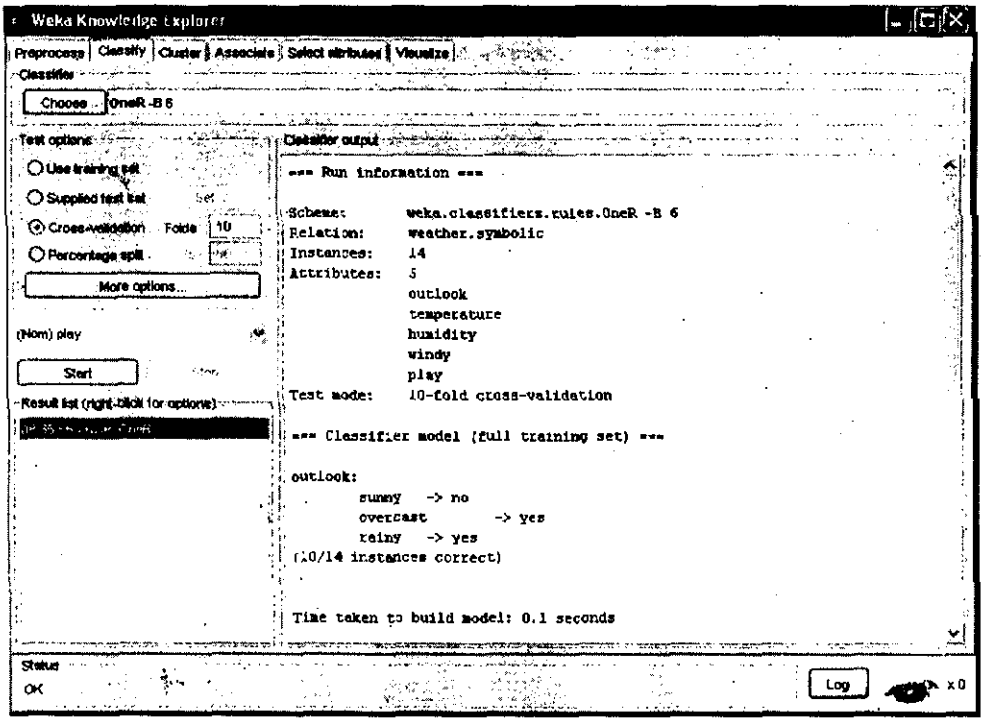
รูปที่ 3.1 ตัวอย่างข้อมูลในรูปแบบ ARFF

ชุดข้อมูลทั้งหมดที่ใช้ในการวิเคราะห์เปรียบเทียบประสิทธิภาพอัลกอริทึมสังเคราะห์โมเดล จะต้องถูกเปลี่ยนให้อยู่ในรูปแบบ ARFF จากนั้นโหลดข้อมูลเข้าสู่โปรแกรม WEKA ด้วยการเลือกแท็บคำสั่ง "Preprocess" และตามด้วยแท็บคำสั่ง "Open file..." ดังรูปที่ 3.2



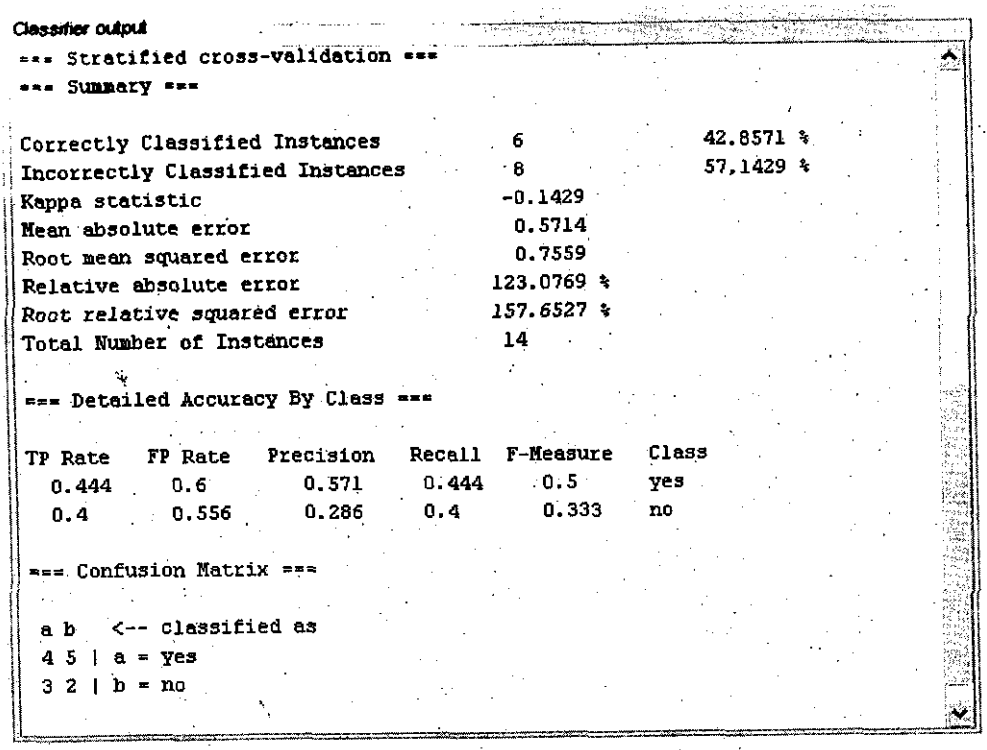
รูปที่ 3.2 แสดงการโหลดข้อมูลในรูปแบบ ARFF เข้าสู่โปรแกรม WEKA

การสังเคราะห์โมเดลทำได้โดยการเลือกแท็บคำสั่ง "Classify" ตามด้วยการเลือกอัลกอริทึมในการ classify และเลือก test options ที่จะใช้ทดสอบโมเดล (หรือ classifier) ดังตัวอย่างในรูปที่ 3.3



รูปที่ 3.3 แสดงการ classify ข้อมูลด้วยอัลกอริทึม OneR

ผลลัพธ์ที่ได้จะปรากฏในกรอบจอภาพ Classifier output ซึ่งจะแสดงโมเดล เวลาที่ใช้ในการสร้างโมเดล และรายละเอียดการทดสอบโมเดล (แสดงในรูปที่ 3.4) เช่นค่า TP Rate, FP Rate, Precision, Recall, F-Measure จำแนกตามคลาส รวมทั้ง confusion matrix แสดงจำนวนข้อมูลที่โมเดลทำนายถูกและทำนายผิดในลักษณะของตารางสองมิติ



รูปที่ 3.4 แสดงหน้าจอผลลัพธ์ของการทำ classification

ตาราง confusion matrix เป็นข้อมูลสำคัญที่จะใช้ในการวัดความแม่นตรงของโมเดล ความแม่นตรง (accuracy) คือ อัตราการทำนายข้อมูลทดสอบได้ถูกต้อง หรือบางครั้งเรียกว่า success rate หรืออัตราความสำเร็จของการทำนายคลาสข้อมูลได้ตรงกับคลาสที่แท้จริง ในกรณีที่ทำนายผิดจะเรียกว่าเป็น error แต่ความผิดพลาดนี้จำแนกย่อยได้เป็นสองประเภท คือ False positive และ False negative แสดงการจำแนก error นี้ได้ดังรูปที่ 3.5 ตาราง confusion matrix ในรูปที่ 3.5 เป็นการทำนายข้อมูลที่มีสองคลาส คือ *คลาส positive และ คลาส negative*

	ทำนายคลาส = "positive"	ทำนายคลาส = "negative"
ลาสที่แท้จริง = "positive"	True positive (**TP**)	False negative (**FN**)
ลาสที่แท้จริง = "negative"	False positive (**FP**)	True neagative (**TN**)

รูปที่ 3.5 เมตริกซ์จำแนกประเภทการทำนายถูก (True) และการทำนายผิด (False)

ในการทำนายเพื่อการวินิจฉัยทางการแพทย์ ค่าความแม่นตรงเพียงอย่างเดียวยังไม่ละเอียดเพียงพอต่อการตัดสินว่า โมเดล หรือ classifier ที่สร้างขึ้นมีประสิทธิภาพเพียงใด จำเป็นต้องวัดเจาะจงให้มากขึ้นว่า การทำนายถูกต้องนั้นเป็น True positive rate เท่าใด และเป็น True negative rate เท่าใด จึงจะตัดสินได้ว่าผลการทำนายเที่ยงตรงเพียงใด (precision) สุดท้ายจึงได้เป็นค่าความแม่นตรง (accuracy) ของโมเดล ตัววัดต่างๆเหล่านี้สามารถคำนวณค่าได้จาก confusion matrix ดังนี้

True positive rate (or sensitivity)	=	TP / (TP + FN)
True negative rate (or specificity)	=	TN / (TN + FP)
Precision	=	TP / (TP + FP)
Accuracy	=	(TP + TN) / (all samples)

ในการวิจัยนี้จะทดสอบเปรียบเทียบประสิทธิภาพของ classifier ที่ได้จากแต่ละอัลกอริทึม และแต่ละเทคนิคโดยวิเคราะห์จากค่าต่างๆ ดังนี้

(1) เวลาที่ใช้ในการสร้างโมเดล วัดในหน่วยของวินาที

(2) ค่า Sensitivity ซึ่งเป็นค่า True positive rate ของคลาสหลัก

(3) ค่า Specificity ซึ่งเป็นค่า True negative rate ของคลาสอื่นๆที่เหลือ

(4) ค่า Precision

(5) ค่า Accuracy

บทที่ 4
ผลการวิเคราะห์เปรียบเทียบประสิทธิภาพการสังเคราะห์โมเดล

การสังเคราะห์โมเดลเพื่อการวินิจฉัยโรค ใช้อัลกอริทึมพื้นฐาน 4 อัลกอริทึม ได้แก่ OneR, J48 (decision-tree induction), naive Bayes, Instance-based (10-nearest neighbors) ประมวลผลด้วยโปรแกรม Weka version 3-2 บนเครื่องคอมพิวเตอร์ PC Pentium III ความเร็ว 700 MHz หน่วยความจำหลัก 256 MB ฮาร์ดดิสก์ความจุ 28 GB ผลการวิเคราะห์เปรียบเทียบประสิทธิภาพการจำแนกของ classifier ปรากฏรายละเอียดในหัวข้อ 4.1 เทคนิคการเพิ่มประสิทธิภาพการสังเคราะห์โมเดล ใช้เทคนิค Bagging และ Boosting กระทำกับอัลกอริทึมพื้นฐานทั้ง 4 อัลกอริทึม ผลการวิเคราะห์เปรียบเทียบการเพิ่มประสิทธิภาพการจำแนก ปรากฏรายละเอียดในหัวข้อ 4.2 หัวข้อ 4.3 เป็นการอภิปรายสรุป

4.1 ผลการวิเคราะห์เปรียบเทียบอัลกอริทึม

ตารางที่ 4.1 ถึง 4.4 ต่อไปนี้แสดงประสิทธิภาพของ classifier ที่ได้จากการสังเคราะห์โมเดลด้วยอัลกอริทึม OneR, J48, naive Bayes และ Instance-based ตามลำดับ ในการแสดงผลจะเรียงลำดับชุดข้อมูลตามจำนวนคลาส โดยเริ่มจากชุดข้อมูลที่เป็น binary class ไปจนถึงชุดข้อมูลที่เป็น multi-class

ตารางที่ 4.1 ประสิทธิภาพของการสังเคราะห์โมเดลด้วยอัลกอริทึม OneR

ชุดข้อมูล (แสดงการจำแนกคลาส)	เวลาที่ใช้สร้างโมเดล (วินาที)	ค่า Sensitivity	ค่า Specificity	ค่า Precision	ค่า Accuracy
Heart disease	0.05				75.04%
Diameter < 50		0.782		0.793	
Diameter ≥ 50			0.705	0.691	
Breast cancer	0				69.23%
No recurrence		0.9		0.727	
Recurrence			0.2	0.459	
Diabetes	0.01				71.48%
Positive		0.466		0.622	
Negative			0.848	0.748	
Heart (Statlog)	0				71.48%
Absent		0.733		0.748	
Present			0.692	0.675	

ตารางที่ 4.1 (ต่อ)

ชุดข้อมูล (แสดงการจำแนกคลาส)	เวลาที่ใช้สร้างโมเดล (วินาที)	ค่า Sensitivity	ค่า Specificity	ค่า Precision	ค่า Accuracy
Liver disorder	0				55.94%
Class 1		0.421		0.473	
Class 2			0.66	0.611	
Wisconsin breast cancer	0.01				91.85%
Benign		0.954		0.924	
Malignant			0.851	0.907	
Hepatitis	0.04				84.52%
Die		0.4		0.667	
Live			0.952	0.869	
Post operative patient	0				68.97%
ICU		0		0	
Stable			0.042	0.333	
Admitted			0.952	0.711	
Thyroid	0				91.16%
Normal		0.947		0.928	
Hyper-thyroid			0.857	0.833	
Hypo-thyroid			0.8	0.923	
Lung cancer	0.01				40.63%
Class 1		0.556		0.625	
Class 2			0.308	0.333	
Class 3			0.4	0.333	
Lymphography	0.03				74.32%
Normal		0		0	
Metastases			0.778	0.84	
Malignant lymph			0.77	0.671	
Fibrosis			0	0	

ตารางที่ 4.1 (ต่อ)

ชุดข้อมูล (แสดงการจำแนกคลาส)	เวลาที่ใช้สร้างโมเดล (วินาที)	ค่า Sensitivity	ค่า Specificity	ค่า Precision	ค่า Accuracy
Primary tumor	0				27.43%
Lung			0.881	0.261	
Head and neck			0.2	0.364	
Esophagus			0	0	
Thyroid			0	0	
Stomach			0.026	0.053	
Duoden.and sm. int			0	0	
Colon			0	0	
Rectum			0	0	
Anus			0	0	
Salivary glands			0	0	
Pancreas			0	0	
Gall bladder			0	0	
Liver			0	0	
Kidney			0	0	
Bladder			0	0	
Testis			0	0	
Prostate			0	0	
Ovary			0	0	
Corpus uteri			0	0	
Cervix uteri			0	0	
Vagina			0	0	
Breast			0.583	0.56	

ตารางที่ 4.2 ประสิทธิภาพของการสังเคราะห์โมเดลด้วยอัลกอริทึม J48

ชุดข้อมูล (แสดงการจำแนกคลาส)	เวลาที่ใช้สร้างโมเดล (วินาที)	ค่า Sensitivity	ค่า Specificity	ค่า Precision	ค่า Accuracy
Heart disease	0.11				80.07%
Diameter < 50		0.864		0.811	
Diameter ≥ 50			0.709	0.783	
Breast cancer	0.01				75.17%
No recurrence		0.965		0.752	
Recurrence			0.247	0.75	
Diabetes	0.1				74.48%
Positive		0.619		0.638	
Negative			0.812	0.799	
Heart (Statlog)	0.04				81.48%
Absent		0.86		0.816	
Present			0.758	0.813	
Liver disorder	0.04				68.70%
Class 1		0.545		0.653	
Class 2			0.79	0.705	
Wisconsin breast cancer	0.06				94.56%
Benign		0.948		0.969	
Malignant			0.942	0.904	
Hepatitis	0.17				81.29%
Die		0.5		0.517	
Live			0.888	0.881	
Post operative patient	0				70.11%
ICU		0		0	
Stable			0	0	
Admitted			0.984	0.709	
Thyroid	0.02				94.42%
Normal		0.967		0.954	
Hyper-thyroid			0.943	0.943	
Hypo-thyroid			0.833	0.893	

ตารางที่ 4.2 (ต่อ)

ชุดข้อมูล (แสดงการจำแนกคลาส)	เวลาที่ใช้สร้างโมเดล (วินาที)	ค่า Sensitivity	ค่า Specificity	ค่า Precision	ค่า Accuracy
Lung cancer	0.02				43.75%
Class 1		0.444		0.4	
Class 2			0.462	0.429	
Class 3			0.4	0.5	
Lymphography	0.37				76.35%
Normal		0.5		0.5	
Metastases			0.815	0.825	
Malignant lymph			0.721	0.733	
Fibrosis			0.5	0.333	
Primary tumor	0.08				41.00%
Lung		0.679		0.483	
Head and neck			0.9	0.692	
Esophagus			0	0	
Thyroid			0.357	0.385	
Stomach			0.051	0.077	
Duoden.and sm. int			0	0	
Colon			0	0	
Rectum			0	0	
Anus			0	0	
Salivary glands			0	0	
Pancreas			0.107	0.1	
Gall bladder			0.563	0.375	
Liver			0	0	
Kidney			0.25	0.4	
Bladder			0	0	
Testis			0	0	
Prostate			0.2	0.5	
Ovary			0.724	0.538	
Corpus uteri			0	0	
Cervix uteri			0	0	
Vagina			0	0	
Breast			0.667	0.8	

ตารางที่ 4.3 ประสิทธิภาพของการสังเคราะห์โมเดลด้วยอัลกอริทึม naive Bayes

ชุดข้อมูล (แสดงการจำแนกคลาส)	เวลาที่ใช้สร้างโมเดล (วินาที)	ค่า Sensitivity	ค่า Specificity	ค่า Precision	ค่า Accuracy
Heart disease	0.03				83.75%
Diameter < 50		0.87		0.858	
Diameter ≥ 50			0.791	0.808	
Breast cancer	0				74.13%
No recurrence		0.866		0.787	
Recurrence			0.447	0.585	
Diabetes	0.01				76.04%
Positive		0.612		0.672	
Negative			0.84	0.802	
Heart (Statlog)	0.01				85.56%
Absent		0.893		0.854	
Present			0.808	0.858	
Liver disorder	0				55.65%
Class 1		0.759		0.482	
Class 2			0.41	0.701	
Wisconsin breast cancer	0.01				95.99%
Benign		0.954		0.984	
Malignant			0.971	0.918	
Hepatitis	0.03				84.52%
Die		0.733		0.579	
Live			0.872	0.932	
Post operative patient	0				70.11%
ICU		0		0	
Stable			0.042	0.333	
Admitted			0.968	0.714	
Thyroid	0				96.74%
Normal		0.987		0.967	
Hyper-thyroid			0.971	0.971	
Hypo-thyroid			0.867	0.963	

ตารางที่ 4.3(ต่อ)

ชุดข้อมูล (แสดงการจำแนกคลาส)	เวลาที่ใช้สร้างโมเดล (วินาที)	ค่า Sensitivity	ค่า Specificity	ค่า Precision	ค่า Accuracy
Lung cancer	0				53.13%
Class 1		0.444		0.5	
Class 2			0.615	0.444	
Class 3			0.5	0.833	
Lymphography	0.02				81.76%
Normal		0.5		0.5	
Metastases			0.901	0.811	
Malignant lymph			0.721	0.846	
Fibrosis			0.75	0.75	
Primary tumor	0				49.56%
Lung		0.702		0.678	
Head and neck			0.95	0.76	
Esophagus			0	0	
Thyroid			0.214	0.333	
Stomach			0.308	0.378	
Duoden.and sm. int			0	0	
Colon			0	0	
Rectum			0	0	
Anus			0	0	
Salivary glands			0	0	
Pancreas			0.357	0.256	
Gall bladder			0.5	0.276	
Liver			0	0	
Kidney			0.458	0.324	
Bladder			0	0	
Testis			0	0	
Prostate			0.2	0.5	
Ovary			0.862	0.543	
Corpus uteri			0	0	
Cervix uteri			0	0	
Vagina			0	0	
Breast			0.792	0.76	

ตารางที่ 4.4 ประสิทธิภาพของการสังเคราะห์โมเดลด้วยอัลกอริทึม Instance-based (10-nearest neighbors)

ชุดข้อมูล (แสดงการจำแนกคลาส)	เวลาที่ใช้สร้างโมเดล (วินาที)	ค่า Sensitivity	ค่า Specificity	ค่า Precision	ค่า Accuracy
Heart disease	0.01				83.58%
Diameter < 50		0.836		0.881	
Diameter ≥ 50			0.836	0.779	
Breast cancer	0				73.43%
No recurrence		0.98		0.732	
Recurrence			0.153	0.765	
Diabetes	0.01				74.35%
Positive		0.601		0.641	
Negative			0.82	0.793	
Heart (Statlog)	0.01				82.96%
Absent		0.893		0.817	
Present			0.75	0.849	
Liver disorder	0				62.90%
Class 1		0.559		0.559	
Class 2			0.68	0.68	
Wisconsin breast cancer	0.01				96.71%
Benign		0.976		0.974	
Malignant			0.95	0.954	
Hepatitis	0.01				83.23%
Die		0.467		0.583	
Live			0.92	0.878	
Post operative patient	0				71.26%
ICU		0		0	
Stable			0	0	
Admitted			1	0.713	
Thyroid	0				92.09%
Normal		1		0.898	
Hyper-thyroid			0.714	1	
Hypo-thyroid			0.767	1	

ตารางที่ 4.4(ต่อ)

ชุดข้อมูล (แสดงการจำแนกคลาส)	เวลาที่ใช้สร้างโมเดล (วินาที)	ค่า Sensitivity	ค่า Specificity	ค่า Precision	ค่า Accuracy
Lung cancer	0				46.88%
Class 1		0.556		0.5	
Class 2			0.538	0.389	
Class 3			0.3	0.75	
Lymphography	0				80.41%
Normal		0		0	
Metastases			0.901	0.793	
Malignant lymph			0.754	0.821	
Fibrosis			0	0	
Primary tumor	0.01				48.67%
Lung		0.726		0.635	
Head and neck			1	0.714	
Esophagus			0	0	
Thyroid			0.286	0.4	
Stomach			0.205	0.4	
Duoden.and sm. int			0	0	
Colon			0	0	
Rectum			0	0	
Anus			0	0	
Salivary glands			0	0	
Pancreas			0.321	0.25	
Gall bladder			0.688	0.344	
Liver			0	0	
Kidney			0.333	0.364	
Bladder			0	0	
Testis			0	0	
Prostate			0	0	
Ovary			0.897	0.413	
Corpus uteri			0	0	
Cervix uteri			0	0	
Vagina			0	0	
Breast			0.75	0.75	

ตารางที่ 4.5 แสดงการเปรียบเทียบเวลาที่ใช้ในการสังเคราะห์โมเดลของทั้งสี่อัลกอริทึม ตารางที่ 4.6 แสดงการเปรียบเทียบค่า Sensitivity และ Specificity ของแต่ละอัลกอริทึมโดยแสดงค่าที่ใช้เปรียบเทียบในลักษณะของ True rate จำแนกตามคลาส, ตารางที่ 4.7 และ 4.8 แสดงการเปรียบเทียบค่า Precision และ Accuracy ของแต่ละอัลกอริทึมตามลำดับ

ตารางที่ 4.5 เปรียบเทียบเวลาที่ใช้ในการสังเคราะห์โมเดลของทั้งสี่อัลกอริทึม

ชื่อชุดข้อมูล	เวลาที่ใช้ (วินาที)			
	OneR	J48	naive Bayes	Instance-based (10-NN)
Heart disease	0.05	0.11	0.03	0.01
Breast cancer	0	0.01	0	0
Diabetes	0.01	0.1	0.01	0.01
Heart (Statlog)	0	0.04	0.01	0.01
Liver disorders	0	0.04	0	0
Wisconsin breast cancer	0.01	0.06	0.01	0.01
Hepatitis	0.04	0.17	0.03	0.01
Post operative	0	0	0	0
Thyroid	0	0.02	0	0
Lung cancer	0.01	0.02	0	0
Lymphography	0.03	0.37	0.02	0
Primary tumor	0	0.08	0	0.01

ตารางที่ 4.6 เปรียบเทียบค่า Sensitivity และ Specificity ในรูปแบบ True rate ของทั้งสี่อัลกอริทึม

ชุดข้อมูล (แสดงการจำแนกคลาส)	ค่า True rate OneR	ค่า True rate J48	ค่า True rate Naive Bayes	ค่า True rate 10-NN
Heart disease				
Diameter < 50	0.782	0.864	0.87	0.836
Diameter ≥ 50	0.705	0.709	0.791	0.836
Breast cancer				
No recurrence	0.9	0.965	0.866	0.98
Recurrence	0.2	0.247	0.447	0.153
Diabetes				
Positive	0.466	0.619	0.612	0.601
Negative	0.848	0.812	0.84	0.82
Heart (Statlog)				
Absent	0.733	0.86	0.893	0.893
Present	0.692	0.758	0.808	0.75
Liver disorder				
Class 1	0.421	0.545	0.759	0.559
Class 2	0.66	0.79	0.41	0.68
Wisconsin breast cancer				
Benign	0.954	0.948	0.954	0.976
Malignant	0.851	0.942	0.971	0.95
Hepatitis				
Die	0.4	0.5	0.733	0.467
Live	0.952	0.888	0.872	0.92
Post operative patient				
ICU	0	0	0	0
Stable	0.042	0	0.042	0
Admitted	0.952	0.984	0.968	1
Thyroid				
Normal	0.947	0.967	0.987	1
Hyper-thyroid	0.857	0.943	0.971	0.714
Hypo-thyroid	0.8	0.833	0.867	0.767

ตารางที่ 4.6 (ต่อ)

ชุดข้อมูล (แสดงการจำแนกคลาส)	ค่า True rate OneR	ค่า True rate J48	ค่า True rate Naive Bayes	ค่า True rate 10-NN
Lung cancer				
Class 1	0.556	0.444	0.444	0.556
Class 2	0.308	0.462	0.615	0.538
Class 3	0.4	0.4	0.5	0.3
Lymphography				
Normal	0	0.5	0.5	0
Metastases	0.778	0.815	0.901	0.901
Malignant lymph	0.77	0.721	0.721	0.754
Fibrosis	0	0.5	0.75	0
Primary tumor				
Lung	0.881	0.679	0.702	0.726
Head and neck	0.2	0.9	0.95	1
Esophagus	0	0	0	0
Thyroid	0	0.357	0.214	0.286
Stomach	0.026	0.051	0.308	0.205
Duoden.and sm. int	0	0	0	0
Colon	0	0	0	0
Rectum	0	0	0	0
Anus	0	0	0	0
Salivary glands	0	0	0	0
Pancreas	0	0.107	0.357	0.321
Gall bladder	0	0.563	0.5	0.688
Liver	0	0	0	0
Kidney	0	0.25	0.458	0.333
Bladder	0	0	0	0
Testis	0	0	0	0
Prostate	0	0.2	0.2	0
Ovary	0	0.724	0.862	0.857
Corpus uteri	0	0	0	0
Cervix uteri	0	0	0	0
Vagina	0	0	0	0
Breast	0.583	0.667	0.792	0.75

ตารางที่ 4.7 เปรียบเทียบค่า Precision ของทั้งสี่อัลกอริทึม

ชุดข้อมูล (แสดงการจำแนกคลาส)	ค่า Precision **OneR**	ค่า Precision **J48**	ค่า Precision **Naive Bayes**	ค่า Precision **10-NN**
Heart disease				
Diameter < 50	0.793	0.811	0.858	0.881
Diameter ≥ 50	0.691	0.783	0.808	0.779
Breast cancer				
No recurrence	0.727	0.752	0.787	0.732
Recurrence	0.459	0.75	0.585	0.765
Diabetes				
Positive	0.622	0.638	0.672	0.641
Negative	0.748	0.799	0.802	0.793
Heart (Statlog)				
Absent	0.748	0.816	0.854	0.817
Present	0.675	0.813	0.858	0.849
Liver disorder				
Class 1	0.473	0.653	0.482	0.559
Class 2	0.611	0.705	0.701	0.68
Wisconsin breast cancer				
Benign	0.924	0.969	0.984	0.974
Malignant	0.907	0.904	0.918	0.954
Hepatitis				
Die	0.667	0.517	0.579	0.583
Live	0.869	0.881	0.932	0.878
Post operative patient				
ICU	0	0	0	0
Stable	0.333	0	0.333	0
Admitted	0.711	0.709	0.714	0.713
Thyroid				
Normal	0.928	0.954	0.967	0.898
Hyper-thyroid	0.833	0.943	0.971	1
Hypo-thyroid	0.923	0.893	0.963	1

ตารางที่ 4.7 (ต่อ)

ชุดข้อมูล (แสดงการจำแนกคลาส)	ค่า Precision OneR	ค่า Precision J48	ค่า Precision Naive Bayes	ค่า Precision 10-NN
Lung cancer				
Class 1	0.625	0.4	0.5	0.5
Class 2	0.333	0.429	0.444	0.389
Class 3	0.333	0.5	0.833	0.75
Lymphography				
Normal	0	0.5	0.5	0
Metastases	0.84	0.825	0.811	0.793
Malignant lymph	0.671	0.733	0.846	0.821
Fibrosis	0	0.333	0.75	0
Primary tumor				
Lung	0.261	0.483	0.678	0.635
Head and neck	0.364	0.692	0.76	0.714
Esophagus	0	0	0	0
Thyroid	0	0.385	0.333	0.4
Stomach	0.053	0.077	0.378	0.4
Duoden.and sm. int	0	0	0	0
Colon	0	0	0	0
Rectum	0	0	0	0
Anus	0	0	0	0
Salivary glands	0	0	0	0
Pancreas	0	0.1	0.256	0.25
Gall bladder	0	0.375	0.276	0.344
Liver	0	0	0	0
Kidney	0	0.4	0.324	0.364
Bladder	0	0	0	0
Testis	0	0	0	0
Prostate	0	0.5	0.5	0
Ovary	0	0.538	0.543	0.413
Corpus uteri	0	0	0	0
Cervix uteri	0	0	0	0
Vagina	0	0	0	0
Breast	0.56	0.8	0.76	0.75

ตารางที่ 4.8 เปรียบเทียบค่า Accuracy ของทั้งสี่อัลกอริทึม

ชุดข้อมูล	ค่า Accuracy **OneR**	ค่า Accuracy **J48**	ค่า Accuracy **Naive Bayes**	ค่า Accuracy **10-NN**
Heart disease	75.04%	80.07%	83.75%	83.58%
Breast cancer	69.23%	75.17%	74.13%	73.43%
Diabetes	71.48%	74.48%	76.04%	74.35%
Heart (Statlog)	71.48%	81.48%	85.56%	82.96%
Liver disorder	55.94%	68.70%	55.65%	62.90%
Wisconsin breast cancer	91.85%	94.56%	95.99%	96.71%
Hepatitis	84.52%	81.29%	84.52%	83.23%
Post operative patient	68.97%	70.11%	70.11%	71.26%
Thyroid	91.16%	94.42%	94.76%	92.09%
Lung cancer	40.63%	43.75%	53.13%	46.88%
Lymphography	74.32%	76.35%	81.76%	80.41%
Primary tumor	27.43%	41.00%	49.56%	48.67%

4.2 ผลการใช้เทคนิค Bagging และ Boosting

เทคนิค Bagging และ Boosting เป็นการสังเคราะห์โมเดลหลายครั้ง (multiple learning) เพื่อหวังผลเพิ่มประสิทธิภาพการทำนายคลาสของข้อมูล ในการวิจัยนี้ทดสอบการทำ Bagging กับอัลกอริทึม OneR, J48, naive Bayes และ Instance-based ปรากฏผลสรุปได้ดังตารางที่ 4.9 ถึง 4.12

การทำ Boosting กับอัลกอริทึม OneR, J48, naive Bayes และ Instance-based ปรากฏผลสรุปได้ดังตารางที่ 4.13 ถึง 4.16

ตารางที่ 4.9 ประสิทธิภาพการทำ **Bagging** กับอัลกอริทึม OneR

ชุดข้อมูล (แสดงการจำแนกคลาส)	เวลาที่ใช้สร้างโมเดล		ค่า True rate (sensitivity and specificity)		ค่า Precision		ค่า Accuracy	
	OneR	Bagging OneR	OneR	Bagging OneR	OneR	Bagging OneR	OneR	Bagging OneR
Heart disease	0.05	0.12					75.04%	77.39%
Diameter < 50			0.782	0.839	0.793	0.791		
Diameter ≥ 50			0.705	0.68	0.691	0.744		
Breast cancer	0	0.01					69.23%	72.38%
No recurrence			0.9	0.92	0.727	0.746		
Recurrence			0.2	0.259	0.459	0.579		
Diabetes	0.01	0.1					71.48%	72.56%
Positive			0.466	0.496	0.622	0.636		
Negative			0.848	0.848	0.748	0.758		
Heart (Statlog)	0	0.04					71.48%	74.07%
Absent			0.733	0.78	0.748	0.76		
Present			0.692	0.692	0.675	0.716		
Liver disorder	0	0.03					55.94%	58.26%
Class 1			0.421	0.414	0.473	0.504		
Class 2			0.66	0.705	0.611	0.624		
Wisconsin breast cancer	0.01	0.11					91.85%	93.13%
Benign			0.954	0.963	0.924	0.934		
Malignant			0.851	0.871	0.907	0.925		
Hepatitis	0.04	0.06					84.52%	87.10%
Die			0.4	0.433	0.667	0.813		
Live			0.952	0.976	0.869	0.878		
Post operative patient	0	0					68.97%	70.11%
ICU			0	0	0	0		
Stable			0.042	0.042	0.333	0.333		
Admitted			0.952	0.968	0.711	0.714		
Thyroid	0	0.03					91.16%	90.23%
Normal			0.947	0.947	0.928	0.916		
Hyper-thyroid			0.857	0.8	0.833	0.824		
Hypo-thyroid			0.8	0.8	0.923	0.923		

ตารางที่ 4.9 (ต่อ)

ชุดข้อมูล	เวลาที่ใช้สร้างโมเดล		ค่า True rate		ค่า Precision		ค่า Accuracy	
	OneR	Bagging	OneR	Bagging	OneR	Bagging	OneR	Bagging
Lung cancer	0.01	0.01					40.63%	46.88%
Class 1			0.556	0.444	0.625	0.5		
Class 2			0.308	0.538	0.333	0.438		
Class 3			0.4	0.4	0.333	0.5		
Lymphography	0.03	0.02					74.32%	70.27%
Normal			0	0	0	0		
Metastases			0.778	0.79	0.84	0.762		
Malignant lymph			0.77	0.656	0.671	0.635		
Fibrosis			0	0	0	0		
Primary tumor	0	0.03					27.43%	27.14%
Lung			0.881	0.976	0.261	0.276		
Head and neck			0.2	0.05	0.364	0.5		
Esophagus			0	0	0	0		
Thyroid			0	0	0	0		
Stomach			0.026	0.128	0.053	0.156		
Duoden.and sm. int			0	0	0	0		
Colon			0	0	0	0		
Rectum			0	0	0	0		
Anus			0	0	0	0		
Salivary glands			0	0	0	0		
Pancreas			0	0	0	0		
Gall bladder			0	0	0	0		
Liver			0	0	0	0		
Kidney			0	0	0	0		
Bladder			0	0	0	0		
Testis			0	0	0	0		
Prostate			0	0	0	0		
Ovary			0	0	0	0		
Corpus uteri			0	0	0	0		
Cervix uteri			0	0	0	0		
Vagina			0	0	0	0		
Breast			0.583	0.167	0.56	0.5		

ตารางที่ 4.10 ประสิทธิภาพการทำ **Bagging** กับอัลกอริทึม J48

ชุดข้อมูล (แสดงการจำแนกคลาส)	เวลาที่ใช้สร้างโมเดล		ค่า True rate (sensitivity and specificity)		ค่า Precision		ค่า Accuracy	
	J48	Bagging J48	J48	Bagging J48	J48	Bagging J48	J48	Bagging J48
Heart disease	0.11	0.96					80.07%	79.73%
Diameter < 50			0.864	0.858	0.811	0.81		
Diameter ≥ 50			0.709	0.709	0.783	0.776		
Breast cancer	0.01	0.14					75.17%	73.08%
No recurrence			0.965	0.92	0.752	0.752		
Recurrence			0.247	0.282	0.75	0.6		
Diabetes	0.1	2.62					74.48%	76.56%
Positive			0.619	0.601	0.638	0.688		
Negative			0.812	0.854	0.799	0.8		
Heart (Statlog)	0.04	0.45					81.48%	79.26%
Absent			0.86	0.78	0.816	0.836		
Present			0.758	0.808	0.813	0.746		
Liver disorder	0.04	0.49					68.70%	68.99%
Class 1			0.545	0.586	0.653	0.644		
Class 2			0.79	0.765	0.705	0.718		
Wisconsin breast cancer	0.06	1.88					94.56%	96.42%
Benign			0.948	0.969	0.969	0.976		
Malignant			0.942	0.954	0.904	0.943		
Hepatitis	0.17	0.28					81.29%	86.45%
Die			0.5	0.533	0.517	0.696		
Live			0.888	0.944	0.881	0.894		
Post operative patient	0	0.07					70.11%	67.82%
ICU			0	0	0	0		
Stable			0	0.042	0	0.2		
Admitted			0.984	0.935	0.709	0.707		
Thyroid	0.02	0.13					94.42%	94.42%
Normal			0.967	0.967	0.954	0.954		
Hyper-thyroid			0.943	0.914	0.943	0.941		
Hypo-thyroid			0.833	0.867	0.893	0.897		

ตารางที่ 4.10 (ต่อ)

ชุดข้อมูล	เวลาที่ใช้สร้างโมเดล		ค่า True rate		ค่า Precision		ค่า Accuracy	
	J48	Bagging	J48	Bagging	J48	Bagging	J48	Bagging
Lung cancer	0.02	0.07					43.75%	50%
Class 1			0.444	0.556	0.4	0.5		
Class 2			0.462	0.462	0.429	0.462		
Class 3			0.4	0.5	0.5	0.556		
Lymphography	0.37	0.15					76.35%	77.70%
Normal			0.5	0	0.5	0		
Metastases			0.815	0.84	0.825	0.81		
Malignant lymph			0.721	0.738	0.733	0.738		
Fibrosis			0.5	0.5	0.333	0.5		
Primary tumor	0.08	0.73					41.00%	42.77%
Lung			0.679	0.679	0.483	0.576		
Head and neck			0.9	0.9	0.692	0.667		
Esophagus			0	0	0	0		
Thyroid			0.357	0.214	0.385	0.375		
Stomach			0.051	0.154	0.077	0.214		
Duoden.and sm. int			0	0	0	0		
Colon			0	0.071	0	0.077		
Rectum			0	0	0	0		
Anus			0	0	0	0		
Salivary glands			0	0.5	0	0.25		
Pancreas			0.107	0.179	0.1	0.179		
Gall bladder			0.563	0.438	0.375	0.35		
Liver			0	0	0	0		
Kidney			0.25	0.25	0.4	0.25		
Bladder			0	0	0	0		
Testis			0	0	0	0		
Prostate			0.2	0	0.5	0		
Ovary			0.724	0.793	0.538	0.548		
Corpus uteri			0	0	0	0		
Cervix uteri			0	0	0	0		
Vagina			0	0	0	0		
Breast			0.667	0.75	0.8	0.72		

ตารางที่ 4.11 ประสิทธิภาพการทำ **Bagging** กับอัลกอริทึม naive Bayes (NB)

ชุดข้อมูล (แสดงการจำแนกคลาส)	เวลาที่ใช้สร้างโมเดล		ค่า True rate (sensitivity and specificity)		ค่า Precision		ค่า Accuracy	
	NB	Bagging NB	NB	Bagging NB	NB	Bagging NB	NB	Bagging NB
Heart disease	0.03	0.11					83.75%	83.92%
Diameter < 50			0.87	0.867	0.858	0.862		
Diameter ≥ 50			0.791	0.799	0.808	0.806		
Breast cancer	0	0.01					74.13%	74.48%
No recurrence			0.866	0.866	0.787	0.791		
Recurrence			0.447	0.459	0.585	0.591		
Diabetes	0.01	0.13					76.04%	75.65%
Positive			0.612	0.612	0.672	0.664		
Negative			0.84	0.834	0.802	0.8		
Heart (Statlog)	0.01	0.07					85.56%	85.19%
Absent			0.893	0.887	0.854	0.853		
Present			0.808	0.808	0.858	0.851		
Liver disorder	0	0.05					55.65%	57.68%
Class 1			0.759	0.703	0.482	0.498		
Class 2			0.41	0.485	0.701	0.693		
Wisconsin breast cancer	0.01	0.13					95.99%	95.99%
Benign			0.954	0.954	0.984	0.984		
Malignant			0.971	0.971	0.918	0.918		
Hepatitis	0.03	0.05					84.52%	85.16%
Die			0.733	0.733	0.579	0.595		
Live			0.872	0.88	0.932	0.932		
Post operative patient	0	0.01					70.11%	71.26%
ICU			0	0	0	0		
Stable			0.042	0	0.333	0		
Admitted			0.968	1	0.714	0.713		
Thyroid	0	0.02					96.74%	96.74%
Normal			0.987	0.993	0.967	0.961		
Hyper-thyroid			0.971	0.943	0.971	1		
Hypo-thyroid			0.867	0.867	0.963	0.963		

ตารางที่ 4.11 (ต่อ)

ชุดข้อมูล	เวลาที่ใช้สร้างโมเดล		ค่า True rate		ค่า Precision		ค่า Accuracy	
	NB	Bagging	NB	Bagging	NB	Bagging	NB	Bagging
Lung cancer	0	0.01					53.13%	59.38%
Class 1			0.444	0.444	0.5	0.571		
Class 2			0.615	0.692	0.444	0.5		
Class 3			0.5	0.6	0.833	0.857		
Lymphography	0.02	0.02					81.76%	83.11%
Normal			0.5	0	0.5	0		
Metastases			0.901	0.914	0.811	0.813		
Malignant lymph			0.721	0.754	0.846	0.868		
Fibrosis			0.75	0.75	0.75	1		
Primary tumor	0	0.03					49.56%	51.03%
Lung			0.702	0.726	0.678	0.693		
Head and neck			0.95	0.95	0.76	0.76		
Esophagus			0	0	0	0		
Thyroid			0.214	0.214	0.333	0.429		
Stomach			0.308	0.308	0.387	0.353		
Duoden.and sm. int			0	0	0	0		
Colon			0	0	0	0		
Rectum			0	0	0	0		
Anus			0	0	0	0		
Salivary glands			0	0.5	0	0.5		
Pancreas			0.357	0.357	0.256	0.303		
Gall bladder			0.5	0.625	0.276	0.345		
Liver			0	0	0	0		
Kidney			0.458	0.5	0.324	0.316		
Bladder			0	0	0	0		
Testis			0	0	0	0		
Prostate			0.2	0.1	0.5	0.5		
Ovary			0.862	0.862	0.543	0.532		
Corpus uteri			0	0	0	0		
Cervix uteri			0	0	0	0		
Vagina			0	0	0	0		
Breast			0.792	0.792	0.76	0.76		

ตารางที่ 4.12 ประสิทธิภาพการทำ **Bagging** กับอัลกอริทึม Instance-based (10-NN)

ชุดข้อมูล (แสดงการจำแนกคลาส)	เวลาที่ใช้สร้างโมเดล		ค่า True rate (sensitivity and specificity)		ค่า Precision		ค่า Accuracy	
	10-NN	Bagging 10-NN	10-NN	Bagging 10-NN	10-NN	Bagging 10-NN	10-NN	Bagging 10-NN
Heart disease	0.01	0.03					83.58%	82.08%
Diameter < 50			0.836	0.793	0.881	0.892		
Diameter ≥ 50			0.836	0.861	0.779	0.742		
Breast cancer	0	0.01					73.43%	74.83%
No recurrence			0.98	0.975	0.732	0.745		
Recurrence			0.153	0.212	0.765	0.783		
Diabetes	0.01	0.02					74.35%	73.96%
Positive			0.601	0.545	0.641	0.652		
Negative			0.82	0.844	0.793	0.776		
Heart (Statlog)	0.01	0.01					82.96%	82.59%
Absent			0.893	0.867	0.817	0.828		
Present			0.75	0.775	0.849	0.823		
Liver disorder	0	0.01					62.90%	61.45%
Class 1			0.559	0.428	0.559	0.554		
Class 2			0.68	0.75	0.68	0.644		
Wisconsin breast cancer	0.01	0.03					96.71%	95.57%
Benign			0.976	0.972	0.974	0.976		
Malignant			0.95	0.954	0.954	0.947		
Hepatitis	0.01	0.01					83.23%	85.16%
Die			0.467	0.5	0.583	0.652		
Live			0.92	0.936	0.878	0.886		
Post operative patient	0	0.01					71.26%	71.26%
ICU			0	0	0	0		
Stable			0	0	0	0		
Admitted			1	1	0.713	0.713		
Thyroid	0	0					92.09%	93.23%
Normal			1	1	0.898	0.909		
Hyper-thyroid			0.714	0.771	1	1		
Hypo-thyroid			0.767	0.767	1	1		

ตารางที่ 4.12 (ต่อ)

ชุดข้อมูล	เวลาที่ใช้สร้างโมเดล		ค่า True rate		ค่า Precision		ค่า Accuracy	
	10-NN	Bagging	10-NN	Bagging	10-NN	Bagging	10-NN	Bagging
Lung cancer	0	0.01					46.88%	50%
Class 1			0.556	0.222	0.5	0.4		
Class 2			0.538	0.769	0.389	0.435		
Class 3			*0.3*	*0.4*	*0.75*	*1*		
Lymphography	0	0.01					80.41%	81.76%
Normal			0	0	0	0		
Metastases			0.901	0.889	0.793	0.818		
Malignant lymph			0.754	0.803	0.821	0.817		
Fibrosis			0	0	0	0		
Primary tumor	0.01	0.02					48.67%	47.49%
Lung			0.726	0.738	0.635	0.681		
Head and neck			1	1	0.714	0.741		
Esophagus			0	0	0	0		
Thyroid			0.286	0.214	0.4	0.375		
Stomach			0.205	0.154	0.4	0.261		
Duoden.and sm. int			0	0	0	0		
Colon			0	0	0	0		
Rectum			0	0	0	0		
Anus			0	0	0	0		
Salivary glands			0	0	0	0		
Pancreas			0.321	0.143	0.25	0.16		
Gall bladder			0.688	0.813	0.344	0.295		
Liver			0	0	0	0		
Kidney			*0.333*	*0.375*	*0.364*	*0.321*		
Bladder			0	0	0	0		
Testis			0	0	0	0		
Prostate			0	0	0	0		
Ovary			0.897	0.897	0.413	0.433		
Corpus uteri			0	0	0	0		
Cervix uteri			0	0	0	0		
Vagina			0	0	0	0		
Breast			0.75	0.75	0.75	0.783		

ตารางที่ 4.13 ประสิทธิภาพการทำ **Boosting** กับอัลกอริทึม OneR

ชุดข้อมูล (แสดงการจำแนกคลาส)	เวลาที่ใช้สร้างโมเดล		ค่า True rate (sensitivity and specificity)		ค่า Precision		ค่า Accuracy	
	OneR	Boost OneR	OneR	Boost OneR	OneR	Boost OneR	OneR	Boost OneR
Heart disease	0.05	0.22					75.04%	76.38%
Diameter < 50			0.782	0.822	0.793	0.788		
Diameter ≥ 50			0.705	0.68	0.691	0.725		
Breast cancer	0	0.03					69.23%	70.63%
No recurrence			0.9	0.881	0.727	0.747		
Recurrence			0.2	0.294	0.459	0.51		
Diabetes	0.01	0.18					71.48%	69.40%
Positive			0.466	0.489	0.622	0.572		
Negative			0.848	0.804	0.748	0.746		
Heart (Statlog)	0	0.11					71.48%	76.67%
Absent			0.733	0.78	0.748	0.796		
Present			0.692	0.75	0.675	0.732		
Liver disorder	0	0.06					55.94%	64.35%
Class 1			0.421	0.531	0.473	0.583		
Class 2			0.66	0.725	0.611	0.681		
Wisconsin breast cancer	0.01	0.16					91.85%	95.28%
Benign			0.954	0.965	0.924	0.963		
Malignant			0.851	0.929	0.907	0.933		
Hepatitis	0.04	0.13					84.52%	79.35%
Die			0.4	0.4	0.667	0.462		
Live			0.952	0.888	0.869	0.86		
Post operative patient	0						68.97%	68.97%
ICU		0	0	0	0	0		
Stable			0.042	0.125	0.333	0.429		
Admitted			0.952	0.919	0.711	0.722		
Thyroid	0	0.03					91.16%	95.35%
Normal			0.947	0.98	0.928	0.955		
Hyper-thyroid			0.857	0.914	0.833	0.97		
Hypo-thyroid			0.8	0.867	0.923	0.929		

ตารางที่ 4.13 (ต่อ)

ชุดข้อมูล	เวลาที่ใช้สร้างโมเดล		ค่า True rate		ค่า Precision		ค่า Accuracy	
	OneR	Boost	OneR	Boost	OneR	Boost	OneR	Boost
Lung cancer	0.01	0					40.63%	50%
Class 1			0.556	0.444	0.625	0.667		
Class 2			0.308	0.538	0.333	0.412		
Class 3			0.4	0.5	0.333	0.556		
Lymphography	0.03	0.07					74.32%	79.05%
Normal			0	0	0	0		
Metastases			0.778	0.84	0.84	0.84		
Malignant lymph			0.77	0.803	0.671	0.731		
Fibrosis			0	0	0	0		
Primary tumor	0	0.01					27.43%	28.02%
Lung			0.881	0.714	0.261	0.536		
Head and neck			0.2	0.05	0.364	0.143		
Esophagus			0	0	0	0		
Thyroid			0	0	0	0		
Stomach			0.026	0.872	0.053	0.155		
Duoden.and sm. int			0	0	0	0		
Colon			0	0	0	0		
Rectum			0	0	0	0		
Anus			0	0	0	0		
Salivary glands			0	0	0	0		
Pancreas			0	0	0	0		
Gall bladder			0	0	0	0		
Liver			0	0	0	0		
Kidney			0	0	0	0		
Bladder			0	0	0	0		
Testis			0	0	0	0		
Prostate			0	0	0	0		
Ovary			0	0	0	0		
Corpus uteri			0	0	0	0		
Cervix uteri			0	0	0	0		
Vagina			0	0	0	0		
Breast			0.583	0	0.56	0		

ตารางที่ 4.14 ประสิทธิภาพการทำ **Boosting** กับอัลกอริทึม J48

ชุดข้อมูล (แสดงการจำแนกคลาส)	เวลาที่ใช้สร้างโมเดล		ค่า True rate (sensitivity and specificity)		ค่า Precision		ค่า Accuracy	
	J48	Boost J48	J48	Boost J48	J48	Boost J48	J48	Boost J48
Heart disease	0.11	1.94					80.07%	80.07%
Diameter < 50			0.864	0.836	0.811	0.829		
Diameter ≥ 50			0.709	0.75	0.783	0.759		
Breast cancer	0.01	0.2					75.17%	69.58%
No recurrence			0.965	0.801	0.752	0.774		
Recurrence			0.247	0.447	0.75	0.487		
Diabetes	0.1	2.08					74.48%	73.31%
Positive			0.619	0.612	0.638	0.619		
Negative			0.812	0.798	0.799	0.793		
Heart (Statlog)	0.04	0.64					81.48%	80.00%
Absent			0.86	0.833	0.816	0.812		
Present			0.758	0.758	0.813	0.784		
Liver disorder	0.04	0.28					68.70%	71.30%
Class 1			0.545	0.641	0.653	0.664		
Class 2			0.79	0.765	0.705	0.746		
Wisconsin breast cancer	0.06	1.02					94.56%	95.85%
Benign			0.948	0.961	0.969	0.976		
Malignant			0.942	0.954	0.904	0.927		
Hepatitis	0.17	0.4					81.29%	83.23%
Die			0.5	0.633	0.517	0.559		
Live			0.888	0.88	0.881	0.909		
Post operative patient	0	0.08					70.11%	58.62%
ICU			0	0	0	0		
Stable			0	0.083	0	0.125		
Admitted			0.984	0.79	0.709	0.7		
Thyroid	0.02	0.2					94.42%	94.88%
Normal			0.967	0.973	0.954	0.954		
Hyper-thyroid			0.943	0.914	0.943	0.941		
Hypo-thyroid			0.833	0.867	0.893	0.929		

ตารางที่ 4.14 (ต่อ)

ชุดข้อมูล	เวลาที่ใช้สร้างโมเดล		ค่า True rate		ค่า Precision		ค่า Accuracy	
	J48	Boost	J48	Boost	J48	Boost	J48	Boost
Lung cancer	0.02	0.09					43.75%	50.00%
Class 1			0.444	0.556	0.4	0.417		
Class 2			0.462	0.308	0.429	0.444		
Class 3			0.4	0.7	0.5	0.636		
Lymphography	0.37	0.19					76.35%	83.78%
Normal			0.5	0	0.5	0		
Metastases			0.815	0.889	0.825	0.847		
Malignant lymph			0.721	0.787	0.733	0.842		
Fibrosis			0.5	1	0.333	0.667		
Primary tumor	0.08	0.41					41.00%	41.00%
Lung			0.679	0.667	0.483	0.479		
Head and neck			0.9	0.9	0.692	0.692		
Esophagus			0	0	0	0		
Thyroid			0.357	0.357	0.385	0.385		
Stomach			0.051	0.051	0.077	0.087		
Duoden.and sm. int			0	0	0	0		
Colon			0	0	0	0		
Rectum			0	0	0	0		
Anus			0	0	0	0		
Salivary glands			0	0	0	0		
Pancreas			0.107	0.143	0.1	0.129		
Gall bladder			0.563	0.563	0.375	0.36		
Liver			0	0	0	0		
Kidney			0.25	0.25	0.4	0.375		
Bladder			0	0	0	0		
Testis			0	0	0	0		
Prostate			0.2	0.2	0.5	0.5		
Ovary			0.724	0.724	0.538	0.538		
Corpus uteri			0	0	0	0		
Cervix uteri			0	0	0	0		
Vagina			0	0	0	0		
Breast			0.667	0.667	0.8	0.8		

ตารางที่ 4.15 ประสิทธิภาพการทำ **Boosting** กับอัลกอริทึม naive Bayes (NB)

ชุดข้อมูล (แสดงการจำแนกคลาส)	เวลาที่ใช้สร้างโมเดล		ค่า True rate (sensitivity and specificity)		ค่า Precision		ค่า Accuracy	
	NB	Boost NB	NB	Boost NB	NB	Boost NB	NB	Boost NB
Heart disease	0.03	0.35					83.75%	82.58%
Diameter < 50			0.87	0.858	0.858	0.849		
Diameter ≥ 50			0.791	0.779	0.808	0.792		
Breast cancer	0	0.09					74.13%	69.23%
No recurrence			0.866	0.811	0.787	0.765		
Recurrence			0.447	0.412	0.585	0.479		
Diabetes	0.01	0.44					76.04%	76.69%
Positive			0.612	0.612	0.672	0.686		
Negative			0.84	0.85	0.802	0.803		
Heart (Statlog)	0.01	0.17					85.56%	86.30%
Absent			0.893	0.873	0.854	0.879		
Present			0.808	0.85	0.858	0.843		
Liver disorder	0	0.09					55.65%	68.41%
Class 1			0.759	0.517	0.482	0.658		
Class 2			0.41	0.805	0.701	0.697		
Wisconsin breast cancer	0.01	0.29					95.99%	95.14%
Benign			0.954	0.965	0.984	0.961		
Malignant			0.971	0.925	0.918	0.933		
Hepatitis	0.03	0.11					84.52%	85.81%
Die			0.733	0.667	0.579	0.625		
Live			0.872	0.904	0.932	0.919		
Post operative patient	0	0.02					70.11%	68.97%
ICU			0	0	0	0		
Stable			0.042	0.083	0.333	0.333		
Admitted			0.968	0.935	0.714	0.716		
Thyroid	0	0.04					96.74%	69.77%
Normal			0.987	1	0.967	0.698		
Hyper-thyroid			0.971	0	0.971	0		
Hypo-thyroid			0.867	0	0.963	0		

ตารางที่ 4.15 (ต่อ)

ชุดข้อมูล	เวลาที่ใช้สร้างโมเดล		ค่า True rate		ค่า Precision		ค่า Accuracy	
	NB	Boost	NB	Boost	NB	Boost	NB	Boost
Lung cancer	0	0.05					53.13%	50%
Class 1			0.444	0.556	0.5	0.5		
Class 2			0.615	0.538	0.444	0.412		
Class 3			0.5	0.4	0.833	0.8		
Lymphography	0.02	0.13					81.76%	78.38%
Normal			0.5	0	0.5	0		
Metastases			0.901	0.815	0.811	0.815		
Malignant lymph			0.721	0.77	0.846	0.758		
Fibrosis			0.75	0.75	0.75	0.75		
Primary tumor	0	0.19					49.56%	49.56%
Lung			0.702	0.702	0.678	0.678		
Head and neck			0.95	0.95	0.76	0.76		
Esophagus			0	0	0	0		
Thyroid			0.214	0.214	0.333	0.333		
Stomach			0.308	0.308	0.387	0.387		
Duoden.and sm. int			0	0	0	0		
Colon			0	0	0	0		
Rectum			0	0	0	0		
Anus			0	0	0	0		
Salivary glands			0	0	0	0		
Pancreas			0.357	0.357	0.256	0.256		
Gall bladder			0.5	0.5	0.276	0.276		
Liver			0	0	0	0		
Kidney			0.458	0.458	0.324	0.324		
Bladder			0	0	0	0		
Testis			0	0	0	0		
Prostate			0.2	0.2	0.5	0.5		
Ovary			0.862	0.862	0.543	0.543		
Corpus uteri			0	0	0	0		
Cervix uteri			0	0	0	0		
Vagina			0	0	0	0		
Breast			0.792	0.792	0.76	0.76		

ตารางที่ 4.16 ประสิทธิภาพการทำ **Boosting** กับอัลกอริทึม Instance-based (10-NN)

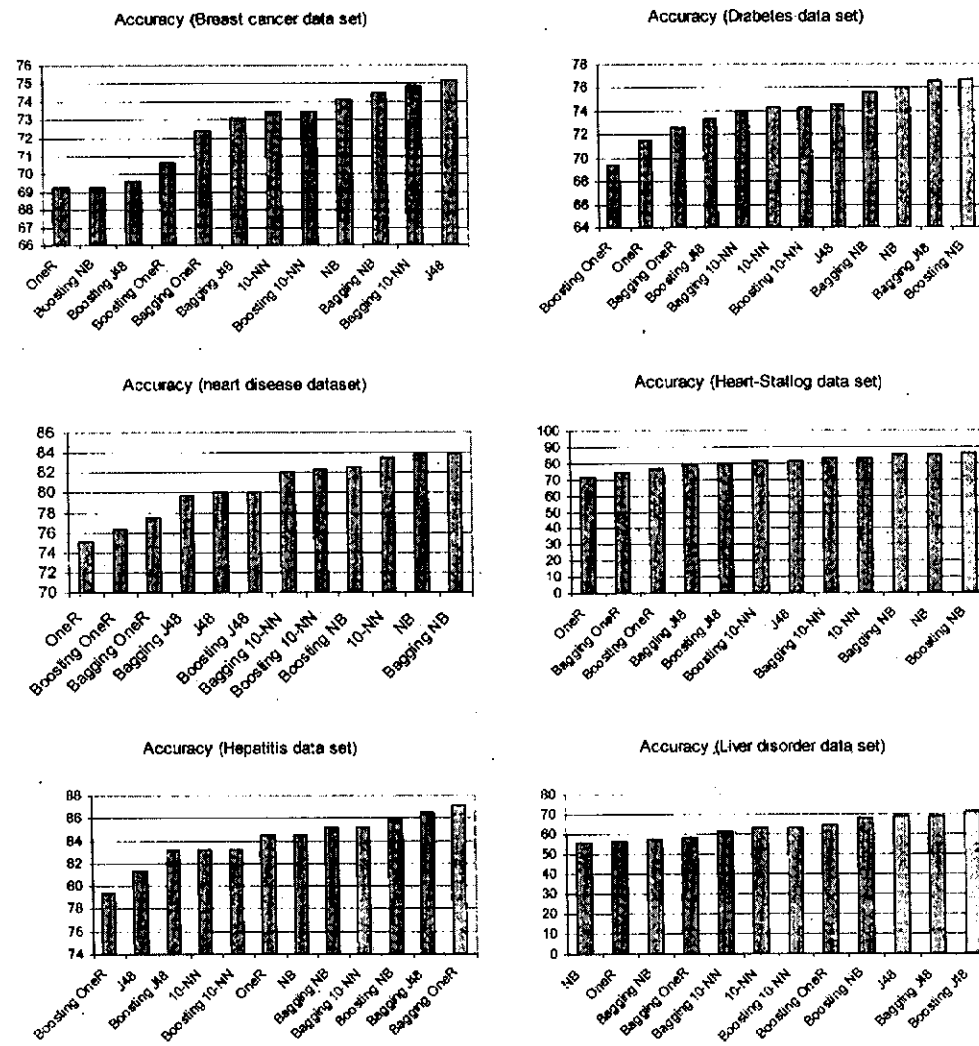
ชุดข้อมูล (แสดงการจำแนกคลาส)	เวลาที่ใช้สร้างโมเดล		ค่า True rate (sensitivity and specificity)		ค่า Precision		ค่า Accuracy	
	10-NN	Boost 10-NN	10-NN	Boost 10-NN	10-NN	Boost 10-NN	10-NN	Boost 10-NN
Heart disease	0.01	21.24					83.58%	82.25%
Diameter < 50			0.836	0.824	0.881	0.869		
Diameter ≥ 50			0.836	0.82	0.779	0.763		
Breast cancer	0	1.72					73.43%	73.43%
No recurrence			0.98	0.98	0.732	0.732		
Recurrence			0.153	0.153	0.765	0.765		
Diabetes	0.01	18.55					74.35%	74.35%
Positive			0.601	0.601	0.641	0.641		
Negative			0.82	0.82	0.793	0.793		
Heart (Statlog)	0.01	5.34					82.96%	81.11%
Absent			0.893	0.86	0.817	0.811		
Present			0.75	0.75	0.849	0.811		
Liver disorder	0	4.94					62.90%	62.90%
Class 1			0.559	0.559	0.559	0.559		
Class 2			0.68	0.68	0.68	0.68		
Wisconsin breast cancer	0.01	24.82					96.71%	96.14%
Benign			0.976	0.972	0.974	0.969		
Malignant			0.95	0.942	0.954	0.946		
Hepatitis	0.01	1.65					83.23%	83.23%
Die			0.467	0.467	0.583	0.583		
Live			0.92	0.92	0.878	0.878		
Post operative patient	0	0.16					71.26%	71.27%
ICU			0	0	0	0		
Stable			0	0	0	0		
Admitted			1	1	0.713	0.713		
Thyroid	0	4.44					92.09%	93.02%
Normal			1	0.953	0.898	0.947		
Hyper-thyroid			0.714	0.914	1	0.941		
Hypo-thyroid			0.767	0.833	1	0.833		

ตารางที่ 4.16 (ต่อ)

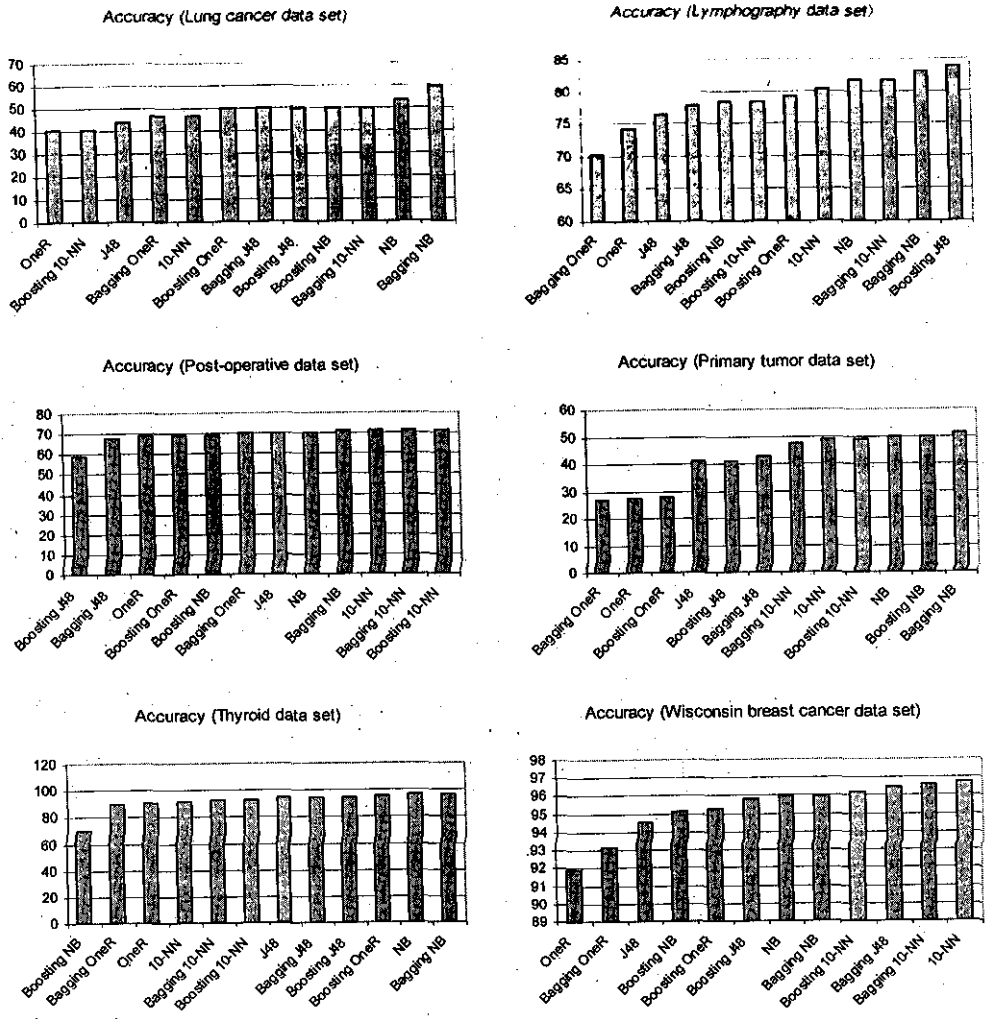
ชุดข้อมูล	เวลาที่ใช้สร้างโมเดล		ค่า True rate		ค่า Precision		ค่า Accuracy	
	10-NN	Boost	10-NN	Boost	10-NN	Boost	10-NN	Boost
Lung cancer	0	0.16					46.88%	40.63%
Class 1			0.556	0.333	0.5	0.375		
Class 2			0.538	0.538	0.389	0.35		
Class 3			0.3	0.3	0.75	0.75		
Lymphography	0	1.57					80.41%	78.38%
Normal			0	0	0	0		
Metastases			0.901	0.852	0.793	0.793		
Malignant lymph			0.754	0.754	0.821	0.767		
Fibrosis			0	0.25	0	1		
Primary tumor	0.01	2.46					48.67%	48.67%
Lung			0.726	0.726	0.635	0.649		
Head and neck			1	1	0.714	0.714		
Esophagus			0	0	0	0		
Thyroid			0.286	0.286	0.4	0.4		
Stomach			0.205	0.205	0.4	0.4		
Duoden.and sm. int			0	0	0	0		
Colon			0	0	0	0		
Rectum			0	0	0	0		
Anus			0	0	0	0		
Salivary glands			0	0	0	0		
Pancreas			0.321	0.321	0.25	0.25		
Gall bladder			0.688	0.688	0.344	0.344		
Liver			0	0	0	0		
Kidney			0.333	0.333	0.364	0.364		
Bladder			0	0	0	0		
Testis			0	0	0	0		
Prostate			0	0	0	0		
Ovary			0.897	0.897	0.413	0.413		
Corpus uteri			0	0	0	0		
Cervix uteri			0	0	0	0		
Vagina			0	0	0	0		
Breast			0.75	0.75	0.75	0.75		

4.3 อภิปรายผล

ผลการทดสอบอัลกอริทึมพื้นฐานในการสังเคราะห์โมเดลเพื่อการวินิจฉัยโรค ทั้งสี่อัลกอริทึม รวมทั้งเทคนิค bagging และ boosting ที่ใช้เพิ่มความแม่นตรงของโมเดล สามารถสรุปเป็นภาพเปรียบเทียบความแม่นตรง (accuracy) ในการทำนายประเภทข้อมูลได้ดังรูปที่ 4.1 ซึ่งแสดงการเปรียบเทียบในลักษณะของกราฟแท่ง จำแนกตามชุดข้อมูล



รูปที่ 4.1 กราฟเปรียบเทียบค่าความแม่นตรงของโมเดลแยกตามชุดข้อมูล



รูปที่ 4.1 กราฟเปรียบเทียบค่าความแม่นตรงของโมเดลแยกตามชุดข้อมูล (ต่อ)

จากข้อมูลในภาพกราฟข้างต้น สามารถสรุปอัลกอริทึมที่ทำงานได้ดีที่สุดกับข้อมูลแต่ละชุดดังแสดงด้วยตารางที่ 4.17 โดยในตารางแสดงคุณสมบัติของข้อมูลร่วมด้วยเพื่อความชัดเจนในการแปลผลความสัมพันธ์ว่าแต่ละอัลกอริทึมและแต่ละเทคนิคเหมาะสมกับข้อมูลประเภทใดบ้าง

ตารางที่ 4.17 ชุดข้อมูลและอัลกอริทึมที่จำแนกข้อมูลได้แม่นตรงที่สุด

ชื่อข้อมูล	จำนวนข้อมูล	จำนวนแอททริบิวต์	จำนวนคลาส	ข้อมูลตัวเลข	ข้อมูลข้อความ	ข้อมูลสูญหาย	อัลกอริทึมแม่นตรงที่สุด	เทคนิคเพิ่มความแม่นตรง
1. Lymphography	148	19	4		✓		NB	Boost J48
2. Post operative	87	9	3		✓		10NN	--
3. Primary tumor	339	18	22		✓	✓	NB	Bag NB
4. Heart disease	597	14	5		✓	✓	J48	Bag J48
5. Breast cancer	286	10	2		✓	✓	J48	--
6. Diabetes	768	9	2	✓			NB	Boost NB
7. Heart (Statlog)	270	14	2	✓			NB	Boost NB
8. Thyroid	215	6	3	✓			NB	--
9. Liver disorder	345	7	2	✓			J48	Boost J48
10.Wisconsin breast cancer	699	10	2	✓		✓	10NN	--
11.Hepatitis	155	20	2	✓		✓	NB, OneR	Bag OneR
12.Lung cancer	32	57	4	✓		✓	NB	Bag NB

จากข้อมูลตามตารางที่ 4.17 สามารถสรุปผลการทดสอบเปรียบเทียบในสามประเด็นหลักได้ดังนี้

<u>การเปรียบเทียบประสิทธิภาพของอัลกอริทึมพื้นฐาน</u>

(1) อัลกอริทึม naive Bayes (NB) ทำงานได้ดีกับข้อมูลที่เป็นตัวเลข (numeric data) และไม่ว่าชุดข้อมูลนั้นจะมีข้อมูลสูญหายปะปนอยู่หรือไม่ ก็ไม่มีผลกระทบต่อความเที่ยงตรงของอัลกอริทึม จะเห็นได้จากในชุดข้อมูลตัวเลขทั้งหมด 7 ชุด อัลกอริทึม naive Bayes ให้ผลการจำแนกข้อมูลเที่ยงตรงที่สุดถึง 5 ชุดข้อมูล ในกรณีที่ชุดข้อมูลประกอบขึ้นจากข้อความหรือสัญลักษณ์ (nominal data) อัลกอริทึมทำงานได้ดีเทียบเท่ากับอัลกอริทึม J48

ข้อสังเกตจากกราฟในรูปที่ 4.1 อัลกอริทึม naive Bayes ให้ความแม่นตรงในการทำนายต่ำที่สุดในชุดข้อมูล Liver disorders ที่เป็นข้อมูลประเภทตัวเลข มีจำนวนแอททริบิวต์น้อย (7 แอททริบิวต์) และมีข้อมูลเพียงสองคลาส

(2) อัลกอริทึม J48 ซึ่งใช้ลักษณะการสร้างต้นไม้ตัดสินใจ (decision-tree induction) ในการสร้างโมเดลเพื่อการจำแนก ทำงานได้ดีกับข้อมูลประเภทข้อความหรือสัญลักษณ์ที่มีจำนวน

คลาสของข้อมูลไม่เกินสองคลาส ประสิทธิภาพของอัลกอริทึม J48 ไม่ได้รับผลกระทบจากกรณีข้อมูลสูญหาย แต่ในกรณีที่ข้อมูลมีจำนวนแอททริบิวต์มาก (high-dimensional data) เช่น ชุดข้อมูล Lung cancer (57 แอททริบิวต์), Hepatitis (20 แอททริบิวต์), Lymphography (19 แอททริบิวต์) และ Primary tumor (18 แอททริบิวต์) อัลกอริทึม J48 ให้ประสิทธิภาพความแม่นตรงในการจำแนกค่อนข้างต่ำมาก

(3) อัลกอริทึม Instance-based หรือ 10-nearest neighbors ให้ผลการจำแนกข้อมูลที่แม่นตรงที่สุดในสองชุดข้อมูล คือ ข้อมูล Post operative patient ซึ่งเป็นข้อมูลประเภทข้อความที่ไม่มีข้อมูลส่วนใดสูญหาย และข้อมูล Wisconsin breast cancer ที่เป็นประเภทตัวเลขและมีบางส่วนของข้อมูลสูญหย ทำให้สรุปในเบื้องต้นได้เพียงว่าอัลกอริทึม Instance-based มีประสิทธิภาพในการสร้างโมเดลที่มีความแม่นตรงในการจำแนก อยู่ในเกณฑ์ปานกลางไปจนถึงดีโดยไม่ขึ้นอยู่กับลักษณะข้อมูล

(4) อัลกอริทึม OneR ซึ่งเป็นอัลกอริทึมที่มีขั้นตอนการสร้างโมเดลค่อนข้างง่ายและให้โมเดลที่ไม่ซับซ้อน แสดงประสิทธิภาพสูงที่สุด (เทียบเท่ากับอัลกอริทึม naive Bayes) ในชุดข้อมูล Hepatitis ที่เป็นข้อมูลประเภทตัวเลข โดยทั่วไปอัลกอริทึม OneR จะไม่ใช่อัลกอริทึมที่ให้ประสิทธิภาพการจำแนกสูงที่สุด แต่นิยมใช้เป็นอัลกอริทึมฐานเพื่อเปรียบเทียบกับอัลกอริทึมชนิดใหม่ที่มีการคิดค้นและพัฒนาขึ้น ผลการทดสอบนี้จัดว่าสอดคล้องกับข้อสังเกตดังกล่าว

<u>การเปรียบเทียบประสิทธิภาพ single learning กับ multiple learning</u>

การเพิ่มประสิทธิภาพการจำแนกของโมเดลด้วยเทคนิค multiple learning (การทำ bagging และ boosting) เมื่อเปรียบเทียบกับ single learning (อัลกอริทึมพื้นฐาน OneR, naive Bayes, J48, Instance-based) สรุปผลการทดสอบได้ดังนี้

(1) จาก 12 ชุดข้อมูล จะเห็นได้ว่าการทำ multiple learning ให้ประสิทธิภาพการทำนายที่แม่นตรงสูงขึ้นถึง 8 ชุดข้อมูล มีสองชุดข้อมูลที่การใช้เทคนิค multiple learning ไม่มีผลต่อการเพิ่มความแม่นตรง (ได้แก่ชุดข้อมูล Post operative patient และข้อมูล Thyroid) และมีสองชุดข้อมูล (ได้แก่ชุดข้อมูล Breast cancer และข้อมูล Wisconsic breast cancer) ที่การทำ multiple learning ให้ผลการทำนายที่มีความแม่นตรงต่ำกว่าอัลกอริทึมพื้นฐานที่เป็น single learning

สังเกตได้ว่าชุดข้อมูลที่การทำ multiple learning ไม่ให้ผลการจำแนกที่ดีขึ้น เป็นข้อมูลที่มีการกระจายในแต่ละคลาสไม่เป็นสัดส่วนที่ใกล้เคียงกัน หรือค่อนข้างจะไม่เป็น uniform distribution ดังแสดงสัดส่วนข้อมูลในแต่ละคลาสของชุดข้อมูลทั้งสี่ชุดได้ดังนี้

ข้อมูล	อัตราเพิ่มของความแม่นตรง	สัดส่วนของข้อมูลในแต่ละคลาสเทียบเป็นร้อยละ
Breast cancer	-2.78%	70.3 : 29.7
Wisconsic breast cancer	-0.14%	65.5 : 34.5
Post operative	0%	1.1 : 27.6 : 71.3
Thyroid	0%	69.8 : 16.3 : 13.9

ในขณะที่ชุดข้อมูลที่เทคนิค multiple learning ใช้เพิ่มอัตราความแม่นตรงในการทำนายข้อมูลอย่างได้ผลดีนั้น การกระจายของข้อมูลในแต่ละคลาสค่อนข้างใกล้เคียงกัน (ยกเว้นเพียงชุดข้อมูลเดียว คือ ข้อมูล Hepatitis ที่ข้อมูลในแต่ละคลาสมีปริมาณที่แตกต่างกันมาก) รายละเอียดของสัดส่วนการกระจายข้อมูลในแต่ละคลาสของทั้งแปดชุดข้อมูลแสดงได้ดังนี้

ข้อมูล	อัตราเพิ่มของความแม่นตรง	สัดส่วนของข้อมูลในแต่ละคลาสเทียบเป็นร้อยละ
Lung cancer	11.76%	28.1 : 40.6 : 31.3
Liver disorders	3.78%	42.1 : 579
Hepatitis	3.05%	20.6 : 79.4
Primary tumor	2.97%	24.8 : 5.9 : 2.7 : 4.1 : 11.5 : 0.3 : 4.1 : 1.8: 0 : 0.6 : 8.3 : 4.7 : 2.1 : 7.1 : 0.6 : 0.3 : 2.9 : 8.6 : 1.8 : 0.6 : 0.3 : 7.1
Lymphography	2.47%	1.4 : 54.7 : 41.2 : 2.7
Heart (Statlog)	0.86%	55.6 : 44.4
Diabetes	0.85%	34.9 : 65.1
Heart disease	0.2%	59.1 : 40.9

(2) เมื่อพิจารณาการทำ multiple learning กับอัลกอริทึมพื้นฐานแต่ละอัลกอริทึม สรุปผลการทดสอบได้ว่า

อัลกอริทึม OneR ใช้ร่วมกับเทคนิค boosting ให้อัตราความแม่นตรงโดยเฉลี่ยเพิ่มขึ้น 4.75% ในขณะที่ใช้เวลาเฉลี่ยเพิ่มขึ้น 0.047 วินาที ให้ผลดีกว่าใช้ร่วมกับเทคนิค bagging ที่ให้อัตราความแม่นตรงโดยเฉลี่ยเพิ่มขึ้น 2.57% และใช้เวลาเฉลี่ยเพิ่มขึ้น 0.08 วินาที

อัลกอริทึม J48 ใช้ร่วมกับเทคนิค bagging ให้อัตราความแม่นตรงโดยเฉลี่ยเพิ่มขึ้น 1.89% ในขณะที่ใช้เวลาเฉลี่ยเพิ่มขึ้น 0.66 วินาที ให้ผลดีกว่าใช้ร่วมกับเทคนิค

boosting ที่ให้อัตราความแม่นตรงโดยเฉลี่ยเพิ่มขึ้น 0.40% และใช้เวลาเฉลี่ยเพิ่มขึ้น 0.63 วินาที

- อัลกอริทึม naive Bayes ใช้ร่วมกับเทคนิค bagging ให้อัตราความแม่นตรงโดยเฉลี่ยเพิ่มขึ้น 1.85% ในขณะที่ใช้เวลาเฉลี่ยเพิ่มขึ้น 0.05 วินาที ให้ผลดีกว่าใช้ร่วมกับเทคนิค boosting ที่ให้อัตราความแม่นตรงโดยเฉลี่ยลดลงเป็น -1.85% และใช้เวลาเฉลี่ยเพิ่มขึ้น 0.16 วินาที
- อัลกอริทึม Instance-based ใช้ร่วมกับเทคนิค bagging ให้อัตราความแม่นตรงโดยเฉลี่ยเพิ่มขึ้น 0.5% ในขณะที่ใช้เวลาเฉลี่ยเพิ่มขึ้น 0.014 วินาที ให้ผลดีกว่าใช้ร่วมกับเทคนิค boosting ที่ให้อัตราความแม่นตรงโดยเฉลี่ยลดลงเป็น -1.61% และใช้เวลาเฉลี่ยเพิ่มขึ้นถึง 7.254 วินาที

จะเห็นได้ว่าเทคนิค multiple learning ใช้งานได้ดีมากกับอัลกอริทึมพื้นฐาน *OneR* ที่มีขั้นตอนการสร้างโมเดลที่ไม่ซับซ้อน ในขณะที่อัลกอริทึม Instance-based ไม่มีผลตอบสนองต่อเทคนิค multiple learning มากนัก

พิจารณาเวลาที่ใช้ในการสร้างโมเดล จะเห็นว่าการทำ boosting กับอัลกอริทึม instance-based จะใช้เวลาเพิ่มขึ้นมากกว่าการทำ bagging อย่างชัดเจน ทั้งนี้เนื่องมาจากลักษณะ lazy evaluation ของอัลกอริทึม ที่จะไม่สร้างโมเดลไว้ล่วงหน้า แต่จะสร้างเมื่อมีข้อมูลที่ต้องการจำแนกเกิดขึ้น เมื่อทำนายข้อมูลด้วยเทคนิค boosting ที่ต้องทำต่อเนื่องกันหลายรอบและปรับค่าน้ำหนักของข้อมูลในแต่ละรอบ จึงทำให้เวลาที่ใช้โดยรวมสูงขึ้น

การพิจารณาผลกระทบของลักษณะข้อมูลที่มีต่อประสิทธิภาพของอัลกอริทึม

(1) ในกรณีข้อมูลที่มีเพียงสองคลาส อัลกอริทึม J48 ทำงานได้ดีเทียบเท่ากับอัลกอริทึม naive Bayes แต่เมื่อข้อมูลมีจำนวนคลาสเพิ่มขึ้น (multi-class) ความแม่นตรงในการจำแนกของอัลกอริทึม naive Bayes จะดีกว่า

(2) ข้อมูลที่มีจำนวนแอททริบิวต์มาก (high-diemensional data) ซึ่งในการทดสอบนี้กำหนดเกณฑ์ที่จำนวนแอททริบิวต์มากกว่า 15 แอททริบิวต์ อัลกอริทึม naive Bayes ทำงานได้ดีในทุกชุดข้อมูลที่เป็น high-dimensional data ในกรณีที่ข้อมูลมีจำนวนแอททริบิวต์ต่ำกว่า 15 แอททริบิวต์ ทั้งอัลกอริทึม naive Bayes, J48 และ Instance-based ทำงานได้ดีใกล้เคียงกัน

(3) ข้อมูลที่ไม่สมบูรณ์ มีบางแอททริบิวต์ของบางเรคคอร์ดที่ไม่ทราบค่า (missing values) ไม่มีผลกระทบต่อประสิทธิภาพของอัลกอริทึม

(4) ข้อมูลที่เป็นตัวเลข (numeric data) หรือข้อมูลที่เป็นค่าไบนารี่ (เช่น true/false, 0/1, yes/no) อัลกอริทึม naive Bayes จะให้ผลการทำงานที่ดีกว่า J48 และ Instance-based อย่างชัดเจน นั่นคือให้ความแม่นตรงที่สูงที่สุดใน 7 ชุดข้อมูลจากจำนวน 9 ชุดข้อมูล

(5) ข้อมูลที่จำแนกได้ยาก เนื่องจากมีจำนวนข้อมูลปรากฏอยู่ในสัดส่วนที่ต่ำ เช่น น้อยกว่า 30% (เรียกว่า rare case) หรืออาจจะเนื่องมาจากไม่ปรากฏรูปแบบหรือแพทเทิร์นที่ชัดเจนในกลุ่มข้อมูล (เรียกว่า hard case) อัลกอริทึม naive Bayes และการใช้เทคนิค boosting ร่วมกับอัลกอริทึม J48 ช่วยให้การจำแนกข้อมูลประเภทนี้ถูกต้องได้มากขึ้น (สังเกตได้จากค่า sensitivity และ specificity ในตารางรายงานผลการทดสอบ)

(6) จำนวนข้อมูลไม่มีผลชี้ว่าอัลกอริทึมใดจะให้ผลการจำแนกที่แม่นตรงกว่าอัลกอริทึมอื่นๆ ทั้งนี้สรุปได้จากในกรณีข้อมูลน้อยกว่า 100 เรคคอร์ด อัลกอริทึม Instance-based และ naive Bayes ทำงานได้ดี ส่วนในกรณีข้อมูลมากกว่า 500 เรคคอร์ด อัลกอริทึม J48, Instance-based และ naive Bayes ทำงานได้ดี แต่ทั้งนี้มีข้อสังเกตว่าอัลกอริทึม J48 ใช้เวลาในการสร้างโมเดลมากกว่าอัลกอริทึมพื้นฐานอื่น จึงอาจจะมีผลกระทบต่อประสิทธิภาพของอัลกอริทึมถ้าข้อมูลมีปริมาณสูงมาก เช่น มากกว่า 10,000 เรคคอร์ด

บทที่ 5
บทสรุป

การวิจัยนี้มีจุดมุ่งหมายที่จะทดสอบประสิทธิภาพของอัลกอริทึมต่างๆ ที่ใช้ในการทำเหมืองข้อมูลประเภทสังเคราะห์โมเดลจำแนกข้อมูล (classification) เพื่อค้นหาอัลกอริทึมที่เหมาะสมที่สุดสำหรับสร้าง classification model ช่วยในการวินิจฉัยโรคทางการแพทย์ ความเหมาะสมของอัลกอริทึมจะพิจารณาที่ความแม่นตรงในการจำแนกเป็นประเด็นหลัก การทดสอบกระทำกับข้อมูลสิบสองชุด ด้วยอัลกอริทึมพื้นฐานสี่อัลกอริทึม และทดสอบกระตุ้นความแม่นตรงของโมเดลด้วยสองเทคนิค คือ bagging และ boosting

5.1 สรุปผลการวิจัย

ผลการทดลองสรุปในประเด็นที่สำคัญได้ดังนี้

(1) ประสิทธิภาพของอัลกอริทึมพื้นฐานที่เป็น single learning เมื่อพิจารณาจากค่า accuracy และค่า precision โดยเฉลี่ย จัดลำดับประสิทธิภาพจากสูงไปต่ำได้ดังนี้

naive Bayes > Instance-based > decision-tree induction > simple-rule induction (1R)

แต่เมื่อพิจารณาเจาะจงแต่ละชุดข้อมูล พบว่าอัลกอริทึม decision-tree induction ให้ประสิทธิภาพการทำงานสูงในชุดข้อมูลประเภทข้อความหรือสัญลักษณ์ และเป็นข้อมูลที่มีเพียงสองคลาส ในขณะที่ naive Bayes ให้ประสิทธิภาพการทำงานดีที่สุดในกรณีข้อมูลส่วนใหญ่เป็นตัวเลข หรือค่าไบนารี่และข้อมูลมีจำนวนคลาสมากกว่าสองคลาส (multi-class)

(2) เทคนิค multiple learning สามารถเพิ่มความแม่นตรงในการจำแนก เมื่อใช้ร่วมกับอัลกอริทึมพื้นฐาน โดยมีข้อจำกัดว่าสัดส่วนของข้อมูลในแต่ละคลาสจะต้องมีจำนวนที่ใกล้เคียงกัน หรือในกรณีที่ข้อมูลมีจำนวนคลาสมากกว่าสองคลาส จะต้องไม่มีข้อมูลในคลาสใดเพียงคลาสเดียวที่มีปริมาณสูงเกินกว่าคลาสอื่นๆที่เหลืออย่างมีนัยสำคัญ

ในกรณีที่การกระจายของข้อมูลในแต่ละคลาสมีสัดส่วนที่แตกต่างกันเกินกว่า 30% (เช่น จากข้อมูลทั้งหมด 100 เรคคอร์ด จำนวนข้อมูลในคลาสที่หนึ่งเป็น 66 เรคคอร์ด จำนวนข้อมูลในคลาสที่สองเป็น 34 เรคคอร์ด ทำให้ข้อมูลในคลาสที่หนึ่งมีมากกว่าข้อมูลในคลาสที่สองคิดเป็น 32%) เทคนิค multiple learning จะไม่ช่วยเพิ่มความแม่นตรงในการจำแนก

(3) ลักษณะของข้อมูลที่มีผลต่อประสิทธิภาพการทำงานของอัลกอริทึม ได้แก่

- จำนวนแอททริบิวต์ หรือ dimension

  อัลกอริทึม naive Bayes ทำงานได้ดีกับ high-dimensional data

- จำนวนคลาส

  อัลกอริทึม decision-tree induction ทำงานได้ดีถึงดีมากกับข้อมูลที่เป็น binary class แต่ทำงานได้ไม่ดีกับข้อมูล multi-class

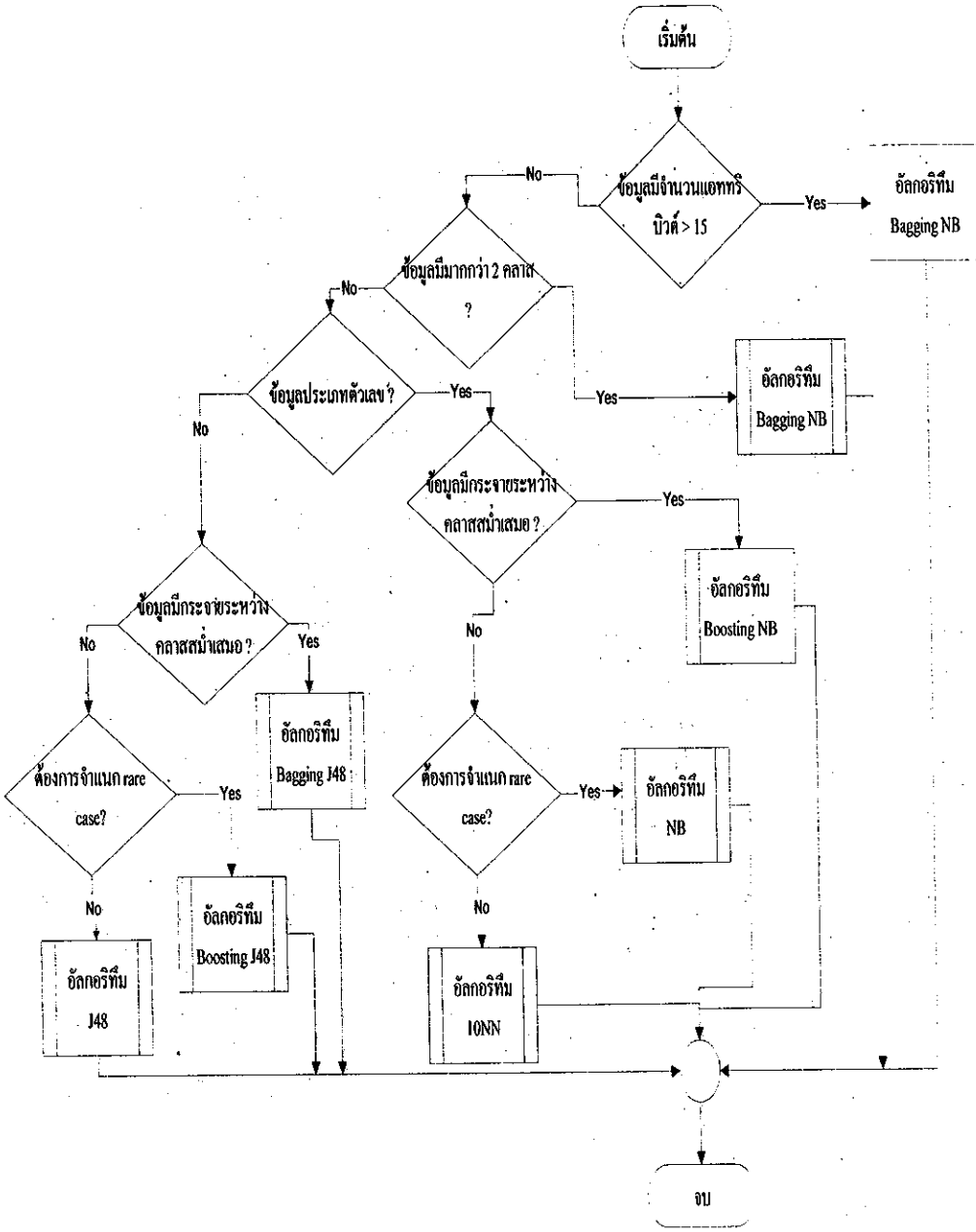
- ประเภทของข้อมูล

  อัลกอริทึม naive Bayes ทำงานได้ดีมากกับข้อมูลประเภทตัวเลข (numeric data) และทำงานได้ดีเทียบเท่ากับอัลกอริทึม decision-tree induction ในข้อมูลประเภทข้อความ (nominal data)

ในการทดสอบอัลกอริทึมพบว่าลักษณะข้อมูลไม่ครบถ้วน หรือ missing values ไม่มีผลกระทบต่อประสิทธิภาพการทำงานของอัลกอริทึม

5.2 ข้อเสนอแนะ

จากการทดสอบอัลกอริทึมสังเคราะห์โมเดลเพื่อการจำแนก (classification model or classifier) ในข้อมูลกลุ่มการวินิจฉัยโรค พบว่าไม่มีอัลกอริทึมเดียวที่ทำงานได้ดีที่สุดกับทุกข้อมูล นั่นคือ ลักษณะของข้อมูลมีผลต่อประสิทธิภาพการทำงานของอัลกอริทึม และจากข้อสังเกตที่ได้จากการทดลองสังเคราะห์โมเดลจากข้อมูลทั้ง 12 ชุด สามารถสรุปเป็นข้อเสนอแนะในการเลือกอัลกอริทึมและเทคนิคที่เหมาะสมกับลักษณะเฉพาะของข้อมูลได้ดังโฟลว์ชาร์ตต่อไปนี้


เริ่มต้น
ข้อมูลมีจำนวนแอททริบิวต์ > 15
Yes
อัลกอริทึม Bagging NB
No
ข้อมูลมีมากกว่า 2 คลาส ?
No
Yes
อัลกอริทึม Bagging NB
ข้อมูลประเภทตัวเลข ?
Yes
No
ข้อมูลมีกระจายระหว่างคลาสสม่ำเสมอ ?
Yes
อัลกอริทึม Boosting NB
ข้อมูลมีกระจายระหว่างคลาสสม่ำเสมอ ?
No
Yes
No
ต้องการจำแนก rare case?
Yes
อัลกอริทึม Bagging J48
ต้องการจำแนก rare case?
Yes
อัลกอริทึม NB
No
อัลกอริทึม J48
อัลกอริทึม Boosting J48
No
อัลกอริทึม IONN
จบ

โฟลว์ชาร์ตที่แสดงข้างต้นเป็นโมเดลที่ใช้ช่วยประกอบการตัดสินใจเลือกอัลกอริทึม โดยพิจารณาตามลักษณะของข้อมูล โมเดลดังกล่าวสามารถแสดงในลักษณะของกฎแบบมีเงื่อนไขได้ดังนี้

```
IF number of attributes > 15
THEN choose Bagging naive Bayes algorithm
ELSE
   IF binary-class
   THEN choose Bagging naive Bayes algorithm
   ELSE
        IF numeric data
        THEN   IF uniform class distribution
               THEN choose Boosting naive Bayes algorithm
               ELSE
                    IF classify rare case
                    THEN choose naive Bayes algorithm
                    ELSE choose 10-nearest neighbors algorithm
        ELSE
               IF uniform class distribution
               THEN choose Bagging decision-tree algorithm
               ELSE
                    IF classify rare case
                    THEN choose Boosting decision-tree algorithm
                    ELSE choose decision-tree induction algorithm
```

❑

โมเดลดังกล่าวมีค่าความคลาดเคลื่อน 0.25 หรือมีอัตราความเที่ยงตรง 75% ซึ่งการนำโมเดลนี้ไปใช้ประโยชน์อย่างได้ผลสมบูรณ์ ยังต้องได้รับการทดสอบและปรับปรุงประสิทธิภาพเพิ่มขึ้น ซึ่งเป็นแนวทางที่ผู้วิจัยมีจุดมุ่งหมายที่จะพัฒนาเพิ่มเติมต่อไปในอนาคต รวมถึงแนวทางการปรับปรุงโมเดลให้ใช้งานได้กับข้อมูลทุกกลุ่มไม่จำกัดอยู่เฉพาะข้อมูลทางการแพทย์เท่านั้น

บรรณานุกรม

Agrawal, R., Manilia, H., Srikant, R., Toivonen, H. and Vermkamo, A.I. (1996). Fast discovery of association rules. In U.M. Fayyad,G. Piatetsky-Shapiro,P. Smyth and R.Uthurusamy, Advances in Knowledge Discovery and Data Mining.pages 307-328. AAAI Press.

Aikins,J.S. (1997). Prototypes and production rules: An approach to knowledge representation for hypothesis formation. In Procedings 6th International Joint Conference on Artificial Intelligence. pages 1-3.

Bauer, E. and Kohavi, R. (1999). An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. Machine Learning, 36, 105-142.

Blake. C., Keogh, E. and Merz, C.J. (1998). UCI Repository of Machine Learning Databases [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Department of Information and Computer Science, University of California, Irvine, CA.

Bratko,I., Mozetic,I. and Lavrac, N. (1989). KARDIO: A Study in Deep and Qualitative Knowledge for Expert Systems. The MIT Press.

Cendrowka,J. (1987). PRISM: An algorithm for inducing modular rules. International Journal of Man-Machine Studies, 27:349-370.

Chandrasekaran, B. and Mittal, S. (1983). Conceptual representation of medical knowledge for diagnosis by computer: MDX and related systems. Advances in Computers, 22: 217-293.

DeClaris, N., Shalvi, D. and Tran-Luu, T.-D. (1996). Computational intelligence-based methodologies for population studies and laboratory medicine decision aids. In Proceedings of the International Neural Network Society 1996 World Congress on Neural Networks, September, San Diego, CA.

Dietterich, T. (2000). An experimental comparison of three methods for constructing ensembles of decision trees: Bagging boosting , and randomization. Machine Learning, *40*(2), 139-158.

Dzeroski, S. and Lavrac,N. (1996). Rule induction and instance-based learning applied in medical diagnosis. Technology and Health Care, 4(2): 203-221.

Fagen,L.M., Shortliffe, E.H. and Buchanan,B.G. (1980). Computer-based medical decision making: From MYCIN to VM. Automedica.

Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P. and Uthurusamy, R. (1996). Advances in Knowledge Discovery and Data Mining, AAAI Press/MIT Press.

Grzymala-Busse,J. (1998). Applications of the ruleinduction systems LERS. In L.Polkowski and A.Skowron, Rough Sets in Knowledge Discovery 1: Methodology and Applications. pages 366-375. Physica-Verlag.

Haiser, J.F., Brooks,R.E. and Ballard, J.P. (1978). Progress report: A computerized psychopharmology advisor. In Proceedings of 11th Collegium Internationale Neuro-Psychopharmatologicum,Vienna.

Han,J. and Kamber, M. (2001). Data Mining: Concepts and Techniques. Morgan Kaufmann.

Holt, R.C.(1993). Very simple classification rulesperform wellon most commonly used datasets. Machine Learning,11: 63-90.

Hubbard, S.M., Martin, N.B. and Thurn, A.L. (1995). NCI's cancer information systems – Bringing medical knowledge to clinicians. Oncology (April) *9*(4), 302-314.

Joshi, M.V., Kumar, V. and Agarwal, R.C. (2001). Evaluating boosting algorithms to classify rare classes: Comparison and improvements. Technical Report RC-22147, IBM Research Division.

Komorowski,J. and Ohrn, A. (1998). Modelling prognostic power of cardiac tests using rough sets. Artificial Intelligence in Medicine.

Kunz,J.C., et al. (1987). A physiological rule-based system for interpreting pulmonary function test results. Technical Report Stanford HPP Memo HPP-78-19.

Lim, T.-S., Loh, W.-Y. and Shih, Y.-S. (2000). A comparison of prediction accuracy, complexity, and training time of thirty-three old and new algorithms. Machine Learning, *40*, 203-229.

Michalski, R., Mozetic,I., Hong,J. and Lavrac,N. (1986). The multi-purpose incremental learning system AQ15 and its testing application on three medicaldomains. In Proceedings 5th National Conference on Artificial Intelligence. pages 1041-1045.

Miller, R.A.(1984). Internist-I/CADUCEUS: Problems facing expert consultant programs. Meth. Inform. Med, 23: 9-14.

Miller, R.A., Pople, H.E. and Myers,J.D. (1982). Internist-I, An experimental computer-based diagnostic consultant for general internal medicine. The New England Journal of Medicine, 29(8): 468-476.

Patil, R.S., Szolovits, P. and Schwartz, W.B. (1982). Modelling knowledge of the patient in acid-base and electrolyte disorders. In P.Szolovits, Artificial Intelligence in Medicine. pages 345-348, AAAS Selected Symposium Series, West View Press.

Pauker, S.G., Gorry, G.A., Kassirer, J.P. and Schwartz, W.B. (1976). Towards the simulation of clinical cognition: Taking a present illness by computer. The American Journal of Medicine, 60: 981-995.

Pople, H.E. (1982). Heuristic methods forimposing structure on ill structured problems: The structuring of medical diagnosis. In P.Szolovits, Artificial Intelligence in Medicine. pages 119-185. AAAI Selected Symposium, West View Press.

Quinlan,J.R. (1986). Induction of decision trees. Machine Learning, 1(1): 81-106.

Quinlan, J.R. (1993). C4.5:Programs for Machine Learning. Morgan Kaufmann.

Safrans, C., Desforges, J. and Tsichlis, P.(1976). Diagnosis planning and cancer management. Report Number TR-169. Technical Report, Laboratory for Computer Science, MIT.

Shortliffe, E.H. (1976). Computer-Based Medical Consultations: MYCIN. Elsevier.

Shortliffe, E.H., Scott, C.A. and Bischoff, M.B. (1981). ONCOCIN: An expert system for oncology protocol management. In Proceedings 7th International Joint Conference on Artificial Intelligence. pages 876-881.

Szolovits,P. and Pauker, S.G. (1978). Categorical and probabilistic reasoning in medical diagnosis. Artificial Intelligence, 11.

Thompson, W.B., Johnson, P.E. and Moen,J.B. (1983). Recognition-based diagnostic reasoning. In Proceedings 8th International Joint Conference on Artificial Intelligence. pages 236-238.

Ting, K.M. (2000). A comparative study of cost-sensitive boosting algorithms. In Proceedings of 17th International Conference on Machine Learning, pages 983-990, Stanford University, CA.

Vernado, S. (1995). The role of information technology in reducing health care cost. In Proceedings of SPIE – The International Society for Optical Engineering volume 2618: Health Care Information Infrastructure, pages 36-40, Philadelphia, PA.

Weiss, S.M., Kulikowski,C.A., Amarel, S. and Safir, A. (1978). A model-based method for computer-aided medical decision making. Artificial Intelligence, 11: 145-172.

Witten,I.H. and Frank, F.(2000). Data Mining: Practical Machine Learning Tools and Techniques with Java Implementation. Morgan Kaufmann.

Zenko, B., Todorovski, L. and Dzeroski, S. (2001). A comparison of stacking with meta decision trees to bagging, boosting, and stacking with other methods. In Proceedings of the 2001 IEEE International Conference on Data Mining, pages 669-670, November, San Jose, CA.

ภาคผนวก

ภาคผนวก ก

บทความผลงานวิจัยนำเสนอในการประชุมวิชาการ

(การประชุมคณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์ ปี 2545)

DATA CLASSIFICATION TECHNIQUES FOR CANCER DATASET*

Nittaya Kerdprasop, Kittisak Kerdprasop, Prathan Saithong, and Surasiri Noppakan

School of Computer Engineering
Suranaree University of Technology
111 University Ave., Muang District,
Nakorn Ratchasima 30000
Phone (044) 224432, Fax (044) 224165
nittaya@ccs.sut.ac.th, kerdpras@ccs.sut.ac.th, thanlove@msn.com, surasiri@hotmail.com

ABSTRACT

We are flooded with a huge volume of data and information. The tremendous amount of data, collected and stored in large databases, has far exceeded the human ability to analyze and extract valuable information for the purpose of decision-making support. Data mining has emerged as a new technology that can intelligently transform the vast amount of data into useful information and knowledge. Data mining tasks can vary from classification, association, to deviation detection. We focus on the classification technique. The objective of this research is to analyze the different techniques and algorithms of data classification with the intention of discovering the appropriate technique for the cancer dataset. The discovered technique must generate the most accurate classifier with the lowest error rate on predicting the class of unseen data.

1. INTRODUCTION

Enormous amounts of data are being collected daily from scientific projects, stocks trading, hospital information systems, computerized sales records and many other sources. A huge volume of data has far exceeded the human ability to analyze and extract valuable information. This situation has urged for new techniques and automated tools that can intelligently transform the pile of data into useful information and knowledge. Data mining is such an imminent promising technology. The benefit of data mining is to turn the newfound knowledge into actionable results such as increasing a customer's likelihood to buy, or improving the ability to identify patterns of cancer recurrence of patients. We focus on the task of classification, the most extensively studied data mining technique.

Classification is a form of data analysis in that it is the process of extracting models (or patterns) to describe data classes or concepts. The extracted model is used to predict the class of unseen data whose class is unknown. For example, each data item in the dataset gathered from patients who were checked-up for a specific type of cancer was labeled as either negative (no cancer) or positive (having cancer). The extracted model might be the common characteristics and symptoms of most patients who had

*This research has been supported by the grant from Suaranaree University of Technology, year 2545 (2002).

cancer. This model is useful for the future prediction to determine who is at high risk of having cancer.

Data classification is a two-step process[8]: learning and classification-testing. In the learning phase, data whose classes are known (called the training data) are analyzed by the classification algorithm to build the model. This model can be represented in various forms, for instance, a decision tree, a set of rules, a mathematical formulae. Since the class of each training data is provided, the classification is categorized as *supervised learning*. In the classification-testing phase, the model is tested on another set of data whose classes are also known (called the test data). The purpose of testing is to estimate the accuracy of the classification model. The process of classification is illustrated in Figure 1.1.

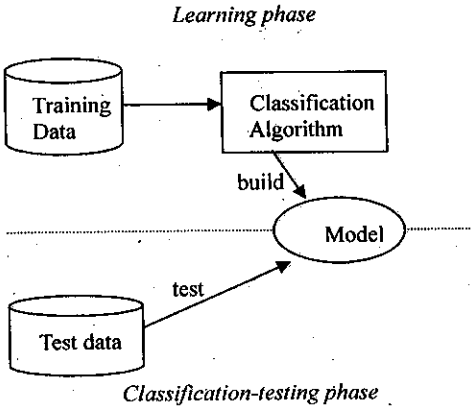


Figure 1.1 The data classification process

Due to the numerous applications of classification (e.g., credit approval, medical diagnosis, stock forecasting), many researchers from diverse disciplines attempt to use various techniques on data classifica-tion. These techniques range from decision tree induction, Bayesian classification, nearest neighbor classification, to case-based reasoning. As the characteristics of data vary from application to application, there is no single technique that performs the best classification on all data types [4, 5, 17, 18].

There is much research in comparing different classification techniques [3, 4, 5, 20]. The team in STATLOG project [18] compares tree-based algorithms against some other classification algorithms on several types of datasets. Another extensive study [17] compares thirty-three classification algorithms. Most comparison studies investigate the algorithms that perform generally well on any kinds of datasets. Our project, on the contrary, emphasizes on the cancer datasets to identify the most appropriate algorithms for this specific domain.

This research compares fourteen different algorithms on two cancer datasets. These datasets are obtained from the UCI Machine Learning Repository [2]. For the purpose of a consistency comparison, we do all experiments in the same environment using the MLC++ system [12]. Each algorithm is compared on the basis of predicting accuracy. The next section explains the datasets used in our experiments. Section 3 briefly describes the classification algorithms. Section 4 outlines the experimental setup. Section 5 reports the results. The last section concludes the paper with some general comments and recommendations.

2. DATASETS

The cancer datasets briefly described in this section are from the UCI Repository [2].

Breast Cancer Dataset

This dataset was obtained from the University Medical Centre, Institute of Oncology, Ljubljana, Yugoslavia. The dataset was reported by M. Zwitter and M. Soklic. The problem is to predict whether a patient who has been treated for breast tumor has recurred-breast-tumor or is safe from the recurrence. The dataset contains 201 instances of no-recurrence class and 85 instances of recurrence class. The

instances are described by 9 attributes, four of which are numerical and five are nominal. Nine instances are removed due to the missing values. Our results are thus based on 277 instances.

Lung Cancer Dataset

The data describes three types of pathological lung cancers. The donor is Stefan Aeberhard. He gives no information on the individual variables. The data contains 32 instances, 56 predictive attributes (all are nominal). The class distribution is 9 instances of class 1, 13 instances of class 2, and 10 instances of class 3.

3. CLASSIFICATION ALGORITHMS

This section describes each classification algorithm briefly. The 14 algorithms are grouped into four categories: basic algorithms (use simple techniques), statistical algorithms, tree-based algorithms, and miscellaneous (use different techniques, e.g., instance-based, decision graph).

3.1 Basic Algorithms

OneR: It is a simple algorithm proposed by Holte [9]. OneR induces classification rules based on the value of a single attribute. OneR is usually used as a base algorithm to compare the predictive accuracy with other sophisticated algorithms. It is shown [9] that we can get reasonable accuracy on many tasks by simply looking at one attribute. The average accuracy of OneR for the datasets tested by Holt is 5.7% lower than that of C4.5.

Const: The algorithm [12] predicts a constant class by simply predicting the majority class in the training data. Although it makes little sense to use this classification scheme for prediction, it can be used as the baseline accuracy to evaluate various classifiers.

Table-majority: A simple table-lookup algorithm [12]. All instances are stored in a table for the purpose of predicting. If an instance is not found, table-majority predicts the majority class of the table.

3.2 Statistical Algorithms

Naïve Bayes: The naïve-Bayes classification algorithm [16] is based on Bayes theorem of posterior probability. Given the instance, the algorithm computes conditional probabilities of the classes and picks the class with the highest posterior. Naïve-Bayes classification assumes that attributes are independence. The probabilities for nominal attributes are estimated by counts, while continuous attributes are estimated by assuming a normal distribution for each attribute and class. Unknown attributes are simply skipped.

Disc-Naïve-Bayes: This is a variant [6] of Naïve Bayes to achieve a better classification by discretizing the continuous attributes. Discretization is performed as a preprocessing step prior to the Naïve-Bayes classification process.

3.3 Tree-Based Algorithms

ID3 and MC4: These are greedy algorithms to induce decision trees for classification. A decision-tree model is built by analyzing training data and the model is used to classify unseen data. ID3 [19] is a very basic decision-tree algorithm with no tree-pruning. The algorithm uses an information-theoretic measure to select the attribute tested for each nonleaf node of the tree. MC4 [12] is a decision-tree algorithm with pruning. Pruning is the technique to improve accuracy by removing the branches reflecting noise in the data.

Option decision tree: The tree has option that allow several optional splits, which are then voted as experts during classification [14].

Lazy DT: Lazy decision tree is an algorithm for building the best decision

tree regarding each test instance [7].

Nbtree: A decision tree algorithm hybrid with Naïve-Bayes at the level of the leaf nodes [11].

3.4 Miscellaneous

IB: IB is an instance-based (or nearest-neighbor) algorithm [1]. The algorithm stores all training instances and builds the classifier when an unseen instance needs to be classified. The non-trivial computation is performed in the prediction time to search the pattern closest to the unknown sample.

HOODG: This Hill-climbing Oblivious, read-Once Decision Graph algorithm uses a bottom-up approach to build a decision graph [10] with a hill-climbing technique implemented.

EODG: This is a classification algorithm to build oblivious decision graph top-down [13]. It cannot handle unknown values.

FSS: The Feature Subset Selection is an algorithm that selects a good subset of features (or attributes) for the improved accuracy performance [15].

4. EXPERIMENTS

For each dataset, the experimentation on fourteen classification algorithms has been performed under the same environment, that is, using the MLC++ system [12]. MLC++ is a library of C++ classes and tools supporting supervised learning of concepts. The system provides a variety of tools that help comparing different learning algorithms.

In supervised machine learning, we try to find a set of rules (a classifier) that can be used to accurately predict the class of unseen instance. Thus, the key factor to compare the performance of different classification algorithms is the accuracy. Accuracy estimation is the process of approximating the future performance of a classifier. We use the holdout method to estimate the accuracy. About two thirds of the data are allocated to the training set (for building a classifier), and the remaining (one third) is allocated to the test set. The accuracy on the test set is the estimated accuracy.

5. RESULTS

The classification accuracy of each algorithm on each dataset is reported as the error rate on the test dataset. The results of performance comparison are summarized in Table 5.1 and are also shown graphically in Figure 5.1.

The following conclusions may be drawn from the results:

1. The basic algorithms (i.e., OneR, Const, Table-majority) employ a simple scheme in building a classifier, mostly predicting a majority class. Thus, their performance can be used as a baseline to compare against sophisticated algorithms.
2. The algorithms that perform better (or as good as) the basic algorithms are LazyDT, MC4, OptionDT. These three algorithms are tree-based.
3. The error rates of most algorithms on the lung-cancer dataset are high due to the small size of the dataset.

6. CONCLUSIONS AND DISCUSSION

By the criterion of error-rate comparison, the most accurate algorithms are those in the group of decision-tree induction. The low error rates of IB, EODG, FSS algorithms require further experimentation on a larger dataset. Even though it is natural to measure a classifier's performance in term of the error rate, for the specific domain of medical diagnosis, the cost of missclassification error should be taken into account. Healthy person incorrectly predicted to be ill (false positive) is much less harmful than sick person incorrectly predicted as healthy (false negative).

Therefore, our future plan is to investigate further the decision-tree

induction algorithms on a larger dataset with different evaluation methods and comparison criteria.

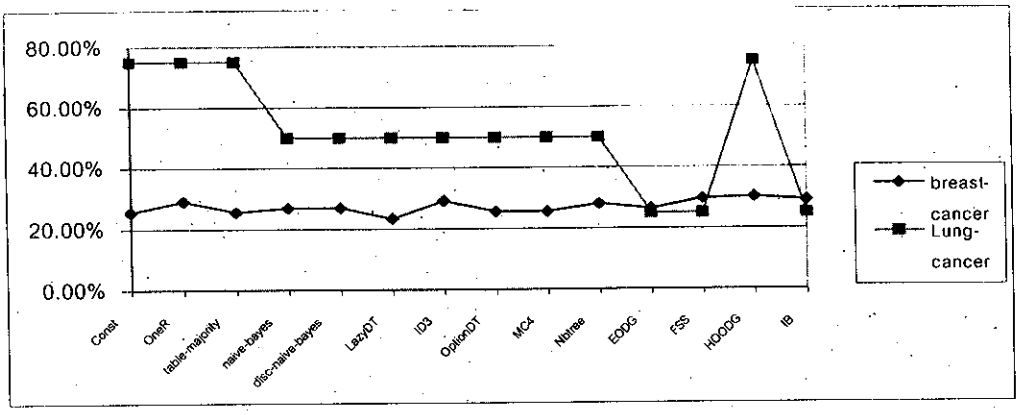


Figure 5.1 Predicting error rate of fourteen algorithms on two datasets

Table 5.1 Error rates of fourteen classification algorithms on two datasets

Dataset	Algorithms						
	Const	OneR	Table-majority	Naïve Bayes	Disc-Naïve-Bayes	LazyDT	ID3
Breast	25.58%	29.07%	25.58%	26.74%	26.74%	23.25%	29.07%
Lung	75%	75%	75%	50%	50%	50%	50%

Dataset	Algorithms						
	OptionDT	MC4	NBtree	EODG	FSS	HOODG	IB
Breast	25.58%	25.58%	27.91%	26.31%	29.47%	30.23%	29.07%
Lung	50%	50%	50%	25%	25%	75%	25%

REFERENCES

[1] D.W.Aha: "Tolerating noisy, irrelevant and novel attributes in instances-based learning algorithms," International Journal of Man-Machine Studies, Vol. 36, No. 1, pp. 267-287, 1992.

[2] C.L. Blake and C.J. Merz: "UCI Repository of Machine Learning Databases," University of California, Irvine, Department of Information and Computer Science, 1998. [http:// www.ics.uci.edu/~mlearn/MLRepository.html]

[3] C.E. Brodley and P.E. Utgoff: "Multi-variate versus univariate decision tree," Technical Report 92-8, Department of Computer Science, University of Massachusetts, Amherst, MA, 1992.

[4] D.E. Brown, V. Corruble, and C.L. Pittard: "A comparison of decision tree classifiers with backpropagation neural networks for multimodal classification problem," Pattern Recognition, Vol. 26, pp.953-961, 1993.

[5] S.P. Curram and J. Mingers: "Neural networks, decision tree induction and discriminant analysis: An empirical comparison," Journal of Operational Research Society, Vol.45, pp.440-450, 1994.

[6] J. Dougherty, R. Kohavi, and M. Sahami: "Supervised and unsupervised discretization of continuous features," Machine Learning: Proceedings of the 12th International Conference, pp.194-202, 1995.

[7] J. Friedman, R. Kohavi, and Y. Yun: "Lazy decision trees," Proceedings of the 13th National Conference on Artificial Intelligence, pp.717-724, 1996.

[8] J.Han and M.Kamber: "Data Mining: Concepts and Techniques,"Morgan Kaufmann, 2001.

[9] R.C.Holte: "Very simple classification rules perform well on most commonly used datasets," Machine Learning, Vol. 11, pp.63-90, 1993.

[10] R. Kohavi: "Bottom-up induction of oblivious, read-once decision graphs," Proceedings of the European Conference on Machine Learning, 1994.

[11] R. Kohavi: "Scaling up the accuracy of Naïve-Bayes classifiers: A decision-tree hybrid," Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining, pp.202-207, 1996.

[12] R. Kohavi, G. John, R. Long, D. Manley, and K. Pfleger: "MLC++: A Machine Learning Library in C++," Tools with Artificial Intelligence, pp.740-743, 1994.

[13] R. Kohavi and C.-H. Li: "Oblivious decision trees, graphs, and top-down pruning," Proceedings of the 14th International Joint Conference on Artificial Intelligence, pp.1071-1077, 1995.

[14] R. Kohavi and C. Kunz: "Option decision trees with majority votes," Machine Learning: Proceedings of the 14th International Conference, pp.161-169, 1997.

[15] R. Kohavi and D. Sommerfield: "Feature subset selection using wrapper model: Overfitting and dynamic search topology," Proceedings of the 1st International Conference on Knowledge Discovery and Data Mining, pp.192-197, 1995.

[16] P. Langley, W. Iba, and K. Thompson: "An analysis of bayesian classifiers," Proceedings of the 10th National Conference on Artificial Intelligence, pp.223-228, 1992.

[17] T.-S. Lim, W.-Y. Loh, and Y.-S. Shih: "A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms," Machine Learning, Vol. 40, pp.203-229, 2000.

[18] D. Michie, D.J. Spiegelhalter, and C.C.Taylor: "Machine Learning, Neural and Statistical Clasification," Ellis Horwood, 1994.

[19] J.R. Quinlan: "Induction of decision trees," Machine Learning, Vol. 1, pp.81-106, 1986.

[20] J.W. Shavlik, R.J. Mooney, and G.G. Towell: "Symbolic and neural learning algorithms: An empirical comparison," Machine Learning, Vol. 6, pp.111-144, 1991.

ภาคผนวก ข

คู่มือการใช้งานโปรแกรม WEKA

WEKA 3.2.1

WEKA Data Mining System

Weka Experiment Environment

Introduction

WEKA พัฒนาโดยมหาวิทยาลัย Waikato ประเทศนิวซีแลนด์ เป็นโปรแกรมที่ใช้ในการวิเคราะห์ข้อมูลที่เราต้องการทราบแนวโน้ม หรือความเป็นไปได้ต่างๆ โดยผู้ใช้สามารถเลือก Algorithms ได้ว่าจะใช้ Algorithms ใดในการวิเคราะห์ข้อมูล

WEKA ได้รวบรวม Algorithms ของ Machine learning เพื่อใช้ในการทำ Data Mining ซึ่งถูกเขียนด้วยภาษา Java จึงสามารถใช้งานได้ในทุกๆ Platform โดย Algorithms ของโปรแกรมสามารถใช้กับ Dataset ได้โดยตรง หรือ เรียกใช้จาก Code ที่ผู้ใช้เขียนขึ้นเองก็ได้

WEKA จึงเป็นโปรแกรมที่เหมาะสำหรับ ผู้ที่ค้นคว้าและพัฒนาเทคโนโลยีทางด้าน Data Mining และ Machine Learning

Download โปรแกรม WEKA ได้ที่

http://prdownloads.sourceforge.net/weka/weka3-2-1jre.exe

หรือ

http://www.cs.waikato.ac.nz/~ml

Scheme ต่างๆใน WEKA

WEKA ได้จัดเตรียม Scheme ต่างๆ สำหรับงาน Data Mining ดังนี้

Scheme ในส่วนของ Classification

- Decision tree inducers
- Rule learners
- Naive Bayes
- Decision tables
- Locally weighted regression
- Support vector machines
- Instance-based learners
- Logistic regression
- Voted perceptrons
- Multi-layer perceptron

Scheme ในส่วนของ Numeric Prediction

- Linear regression
- Model tree generators
- Locally weighted regression
- Instance-based learners
- Decision tables
- Multi-layer perceptron

Scheme ในส่วนของ " Meta-Schemes "

- Bagging
- Stacking
- Boosting
- Regression via classification
- Classification via regression
- Cost sensitive classification

ทั้งนี้ ยังรวมถึง Method ในการทำ Clustering เช่น Cobweb and an EM algorithm และ Association เช่น Apriori และเครื่องมืออีกมากมายที่จะใช้ในกระบวนการ pre-processing datasets

Package ของ WEKA

WEKA ซึ่งเขียนด้วย Java ดังนั้น Functions การทำงานหลักๆ จึงเขียนเป็น Package ของ Java ซึ่งมีทั้งหมด 21 Package ดังนี้

- package weka.associations
- package weka.attributeSelection
- package weka.classifiers
- package weka.classifiers.adtree
- package weka.classifiers.evaluation
- package weka.classifiers.j48
- package weka.classifiers.kstar
- package weka.classifiers.m5
- package weka.classifiers.neural
- package weka.clusterers
- package weka.core
- package weka.core.converters
- package weka.estimators
- package weka.experiment
- package weka.filters
- package weka.gui
- package weka.gui.experiment
- package weka.gui.explorer
- package weka.gui.streams
- package weka.gui.treevisualizer
- package weka.gui.visualize

เนื่องจาก Package ของโปรแกรม WEKA นั้น มีค่อนข้างมาก จึงนำเสนอ เพียง package ที่สำคัญต่อการใช้งาน คือ Associations , Classifier และ Clustering ดังนี้

1. package weka.associations

เป็น Package ที่ประกอบด้วย Class ที่ใช้ในการทำ Associations Dataset โดยแต่ละ Class จะเป็น Algorithms ที่ใช้ในการ Associate ต่างๆ กันทั้งหมด 3 Class ดังนี้

- Apriori
- Associator
- ItemSet

2. package weka.classifiers

เป็น Package ที่ประกอบด้วย Class ที่ใช้ในการทำ Classify Dataset โดยแต่ละ Class จะเป็นAlgorithms ที่ใช้ในการ Classify ต่างๆ กันทั้งหมด 41 Class ดังนี้

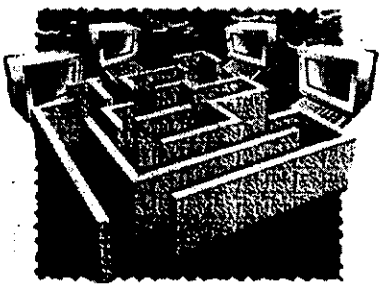
- AdaBoostM1
- AdditiveRegression
- AttributeSelectedClassifier
- BVDecompose
- Bagging
- CVParameterSelection
- CheckClassifier
- ClassificationViaRegression
- Classifier
- CostMatrix
- CostSensitiveClassifier
- DecisionStump
- DecisionTable
- DistributionClassifier
- DistributionMetaClassifier
- Evaluation
- FilteredClassifier
- HyperPipes
- IB1
- IBk
- Id3
- KernelDensity
- LWR
- LinearRegression
- Logistic
- LogitBoost

- MetaCost
- MultiClassClassifier
- MultiScheme
- NaiveBayes
- NaiveBayesSimple
- OneR
- Prism
- RegressionByDiscretization
- SMO
- Stacking
- ThresholdSelector
- UserClassifier
- VFI
- VotedPerceptron
- ZeroR

3. package weka.clusterers

เป็น Package ที่ประกอบด้วย Class ที่ใช้ในการทำ Clustering แต่ละ Class จะเป็น Algorithms ที่ใช้ในการ Clustering ต่างๆกันทั้งหมด 7 Class ดังนี้

- ClusterEvaluation
- Clusterer
- Cobweb
- DistributionClusterer
- DistributionMetaClusterer
- EM
- SimpleKMeans



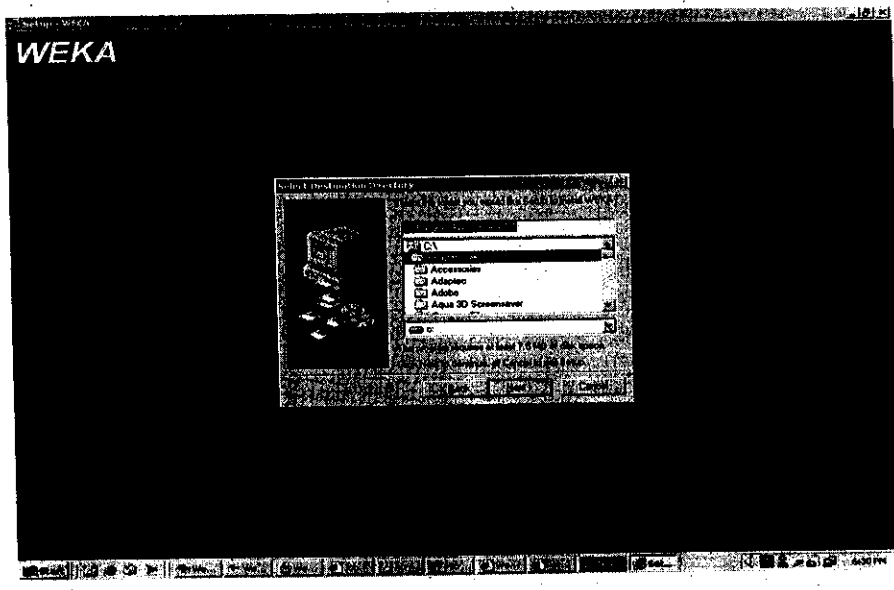
การติดตั้งโปรแกรม WEKA-3-2

การ Download สำหรับผู้ที่ใช้ Microsoft Windows

จะมี 2 รูปแบบ คือ

- Download เฉพาะ Software WEKA (ใช้หน่วยความจำประมาณ 3,898,241 bytes)
- Download WEKA พร้อมกับ Java runtime สำหรับเครื่องที่ไม่ได้ลง Java Runtime หรือ Java Compiler (ใช้หน่วยความจำประมาณ 11,520,440 bytes)

การติดตั้งโปรแกรม WEKA >>> ตัว Installer จะทำการ Install ให้โดยอัตโนมัติ



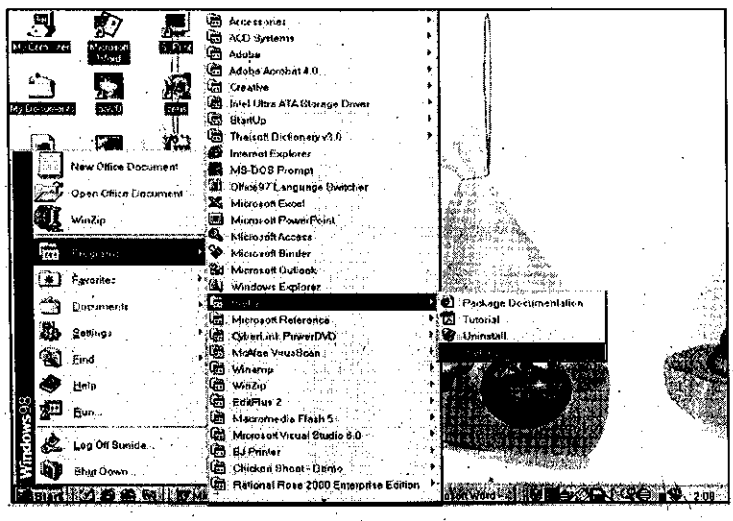
รูปที่ ข1 แสดงการ Install โปรแกรม WEKA

SYSTEM REQUIREMENT

- Java 1.2 (หรือใหม่กว่านั้น) แต่ผู้เขียนแนะนะให้ใช้ Java 2 ขึ้นไป
- Java Swing เพื่อการใช้งาน Graphic User Interface ที่ดีกว่า
- เนื้อที่ว่าง 7.5 Megabyte

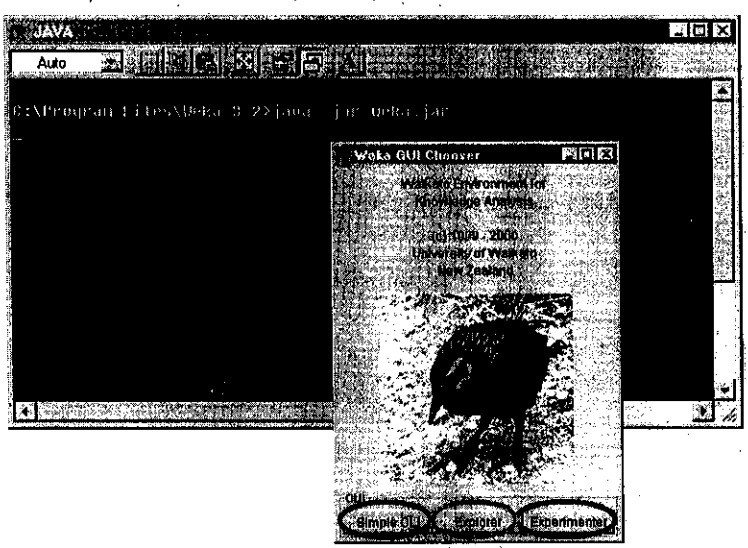
การใช้งาน WEKA

⇒ เริ่มต้นการใช้งาน WEKA ให้ Click ที่ **Menu Start > Programs > WEKA > Weka-3-2** ดังนี้



รูปที่ ข2 แสดงการเริ่มต้นใช้งานโปรแกรม WEKA

เมื่อ Click เข้าไปใน WEKA แล้ว จะขึ้นหน้าต่างของ Program ดังนี้



รูปที่ ข3 แสดงหน้าต่างของโปรแกรม WEKA

จากรูปข้างต้นจะเห็นได้ว่าใน Weka GUI Chooser นั้นจะมี Menu อยู่ 3 Menu ดังนี้

- Simple CLI
- Explorer
- Experimenter

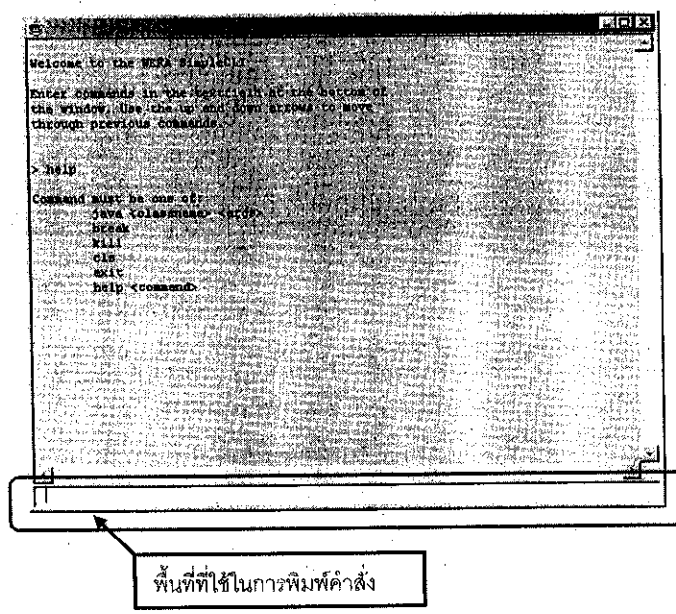
. WEKA จะแบ่ง Interface การทำงานกับผู้ใช้ได้ 2 แบบ คือ GUI (Graphic User Interface) และ Command Line โดย

- Simple CLI คือ Mode การทำงานแบบ Command Line
- Explorer และ Experiment คือ Mode การทำงานแบบ GUI

Simple CLI

Simple CLI เป็น Mode การทำงานแบบ Command Line ในการใช้งานนั้นเราสามารถพิมพ์คำสั่งลงไปในช่องว่างด้านล่างของส่วน CLI ได้เลย โดยพิมพ์คำสั่งต่อเนื่องลงไปเหมือนการใช้งานในโหมด Command Line

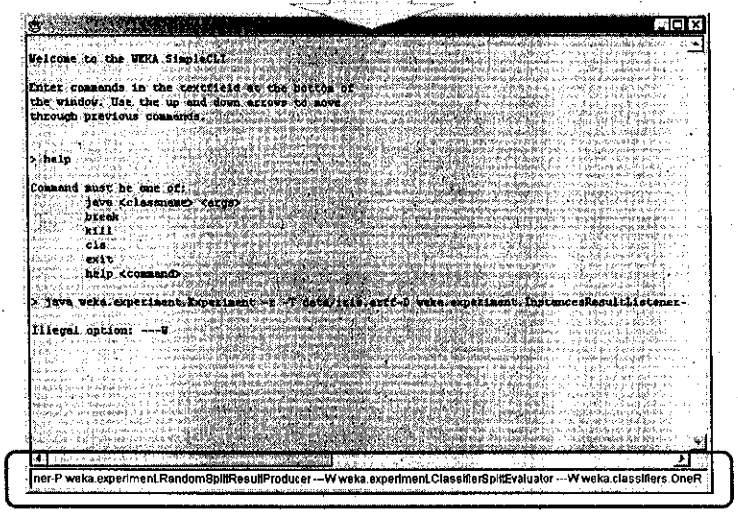
เมื่อ Click ที่ Simple CLI แล้ว จะขึ้นหน้าต่างดังนี้



รูปที่ ข4 แสดงหน้าต่างของ Simple CLI

ตัวอย่างการทำงานโดยใช้ Simple CLI : คำสั่งดังต่อไปนี้สามารถถูกพิมพ์เข้าไปใน CLI เพื่อ Run Scheme บน Iris Dataset ซึ่งคำสั่งจะถูกพิมพ์ต่อเนื่องลงไป บนหนึ่งบรรทัดใน CLI

```
java weka.experiment.Experiment -r -T data/iris.arff
-D weka.experiment.InstancesResultListener
-P weka.experiment.RandomSplitResultProducer --
-W weka.experiment.ClassifierSplitEvaluator --
-W weka.classifiers.OneR
```

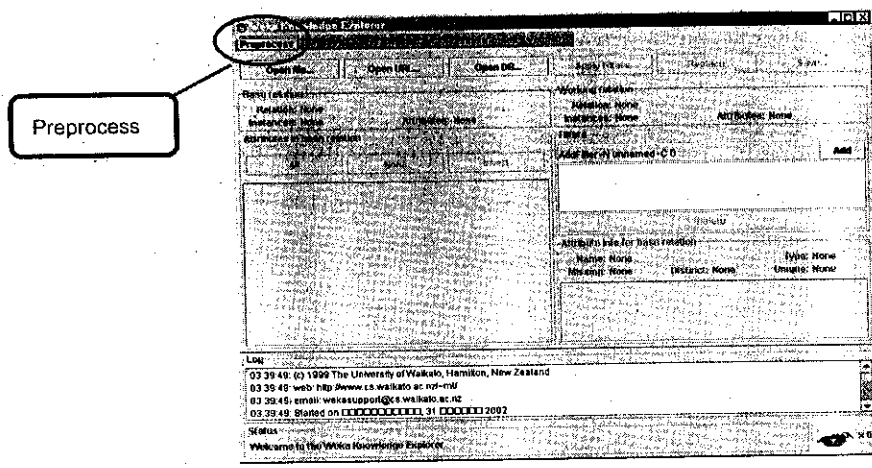


รูปที่ ข5 แสดงการใช้งานโดยพิมพ์คำสั่งแบบ Command Line

จะเห็นได้ว่าการพิมพ์คำสั่งแบบนี้ไม่สะดวกเท่าที่ควร คือ ต้องพิมพ์คำสั่งโดยตรง เข้าไปใน CLI ซึ่งคำสั่งบางคำสั่งนั้นยาวมาก และในการพิมพ์คำสั่งแต่ละครั้งอาจผิดพลาดได้ เนื่องจากต้องพิมพ์ต่อเนื่องกันในบรรทัดเดียว ทำให้ผลที่ได้อาจไม่ตรงตามต้องการ หรือเกิด error ขึ้น นอกจากนี้ในส่วน Experiments นั้น ทำการเปลี่ยนแปลงได้ยาก เพื่อทำให้ง่ายและยืดหยุ่นมากขึ้น จึงได้พัฒนาให้สามารถใช้งานแบบกราฟฟิก (GUI) ได้ คือ Explorer และ Experiment

Explorer

Explorer เป็น mode การทำงานแบบ GUI (Graphic User Interface) ซึ่งเมื่อ Click ที่ Explorer แล้ว ก็จะขึ้นหน้าต่างดังภาพ



รูปที่ ข6 แสดงหน้าต่างของ Explorer

ภายในหน้าต่างดังกล่าวจะมี Tool ต่างๆ อยู่ 6 Tool ด้วยกัน ดังนี้

1. **Preprocess :** ภายใน Tool นี้ ก็จะมี Menu ย่อย 3 Menu ให้เลือกใช้ ดังนี้
   - Open file...
   - Open URL...
   - Open DB ...
2. **Classify :** ภายใน Tool นี้ ก็จะมี Tool ย่อย ให้ Set ค่าต่างๆ ดังนี้
   - Classifier
   - Test options
   - Classifier output
   - Result list
   - Log
   - Status

3. **Cluster :** ภายใน Tool นี้ ก็จะมี Tool ย่อย ให้ Set ค่าต่างๆ ดังนี้

- Clusterer
- Cluster mode
- Cluster output
- Result list
- Log
- Status

4. **Associate :** ภายใน Tool นี้ ก็จะมี Tool ย่อย ให้ Set ค่าต่างๆ ดังนี้

- Associator
- Associator output
- Result list
- Log
- Status

5. **Select attributes :** ภายใน Tool นี้ ก็จะมี Tool ย่อย ให้ Set ค่าต่างๆ ดังนี้

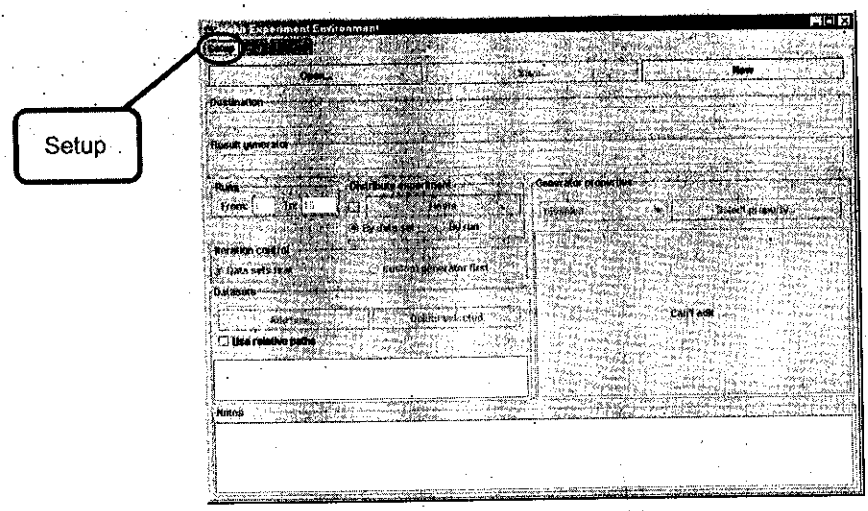
- Attribute Evaluator
- Search Method
- Attribute Selection Mode
- Attribute Selection Output
- Result list
- Log
- Status

6. **Visualize :** ภายใน Tool นี้ ก็จะมี Tool ย่อย ให้ Set ค่าต่างๆ ดังนี้

- Plot CPU
- Class color
- Jitter
- Log
- Status

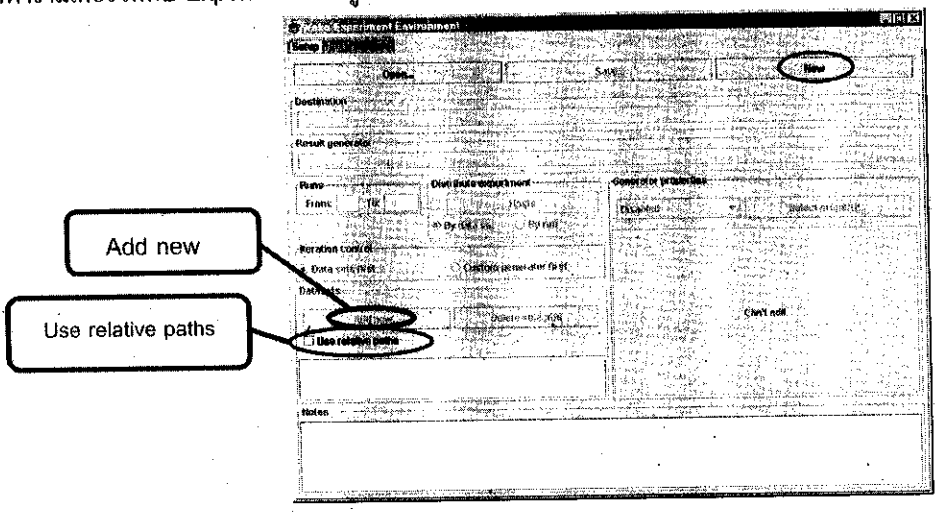
Experimenter

Experimenter เป็น Mode การทำงานแบบ GUI (Graphic User Interface) ซึ่งเมื่อ Click ที่ Experimenter แล้ว ก็จะขึ้นหน้าต่างดังนี้



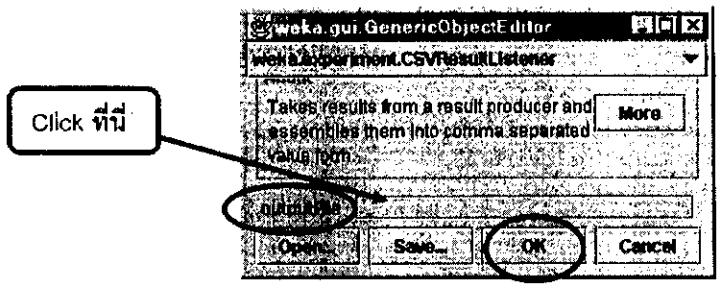
รูปที่ ข7 แสดงหน้าต่างของ Experiment

เมื่อเรียกใช้ Experimenter จะแสดงหน้าต่าง Setup ให้ Click ที่ **New** เพื่อกำหนดค่าพารามิเตอร์ให้กับ Experiment ดังรูป



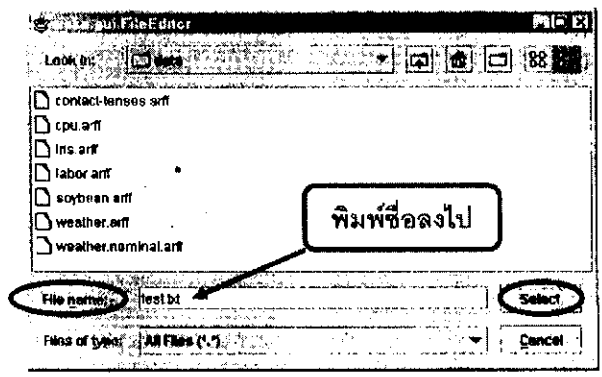
รูปที่ ข8 แสดงการกำหนดค่าพารามิเตอร์ให้กับ Experiment

การกำหนดค่า Dataset ที่ต้องการให้ถูกประมวลผล ในขั้นแรกเลือก "Use relative paths" แล้วเลือก Dataset ที่ต้องการทำ Experiment เมื่อเลือกแล้วจะปรากฏ ชื่อของ Dataset ใน Dataset panel ของหน้าต่าง Setup และ Click ที่ "Add new ..." ในการเปิด Dialog Window จากนั้นจึงทำการเก็บผลลัพธ์ของการ Experiment โดยไป Click ที่ ช่อง Destination จะปรากฏหน้าต่าง ดังรูป



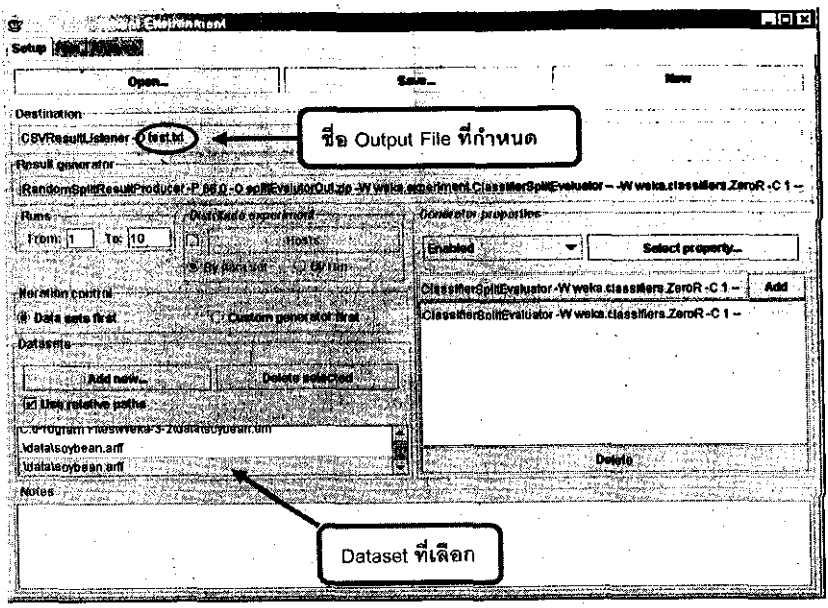
รูปที่ ข9 แสดงหน้าต่าง Output File

จากนั้นให้เลือก Output File ที่จะทำการเก็บผลลัพธ์โดยการ Click ที่ ในช่องของ Output File จะปรากฏหน้าต่าง จากนั้นให้ ตั้งชื่อ Output file ดังรูปที่ ข10



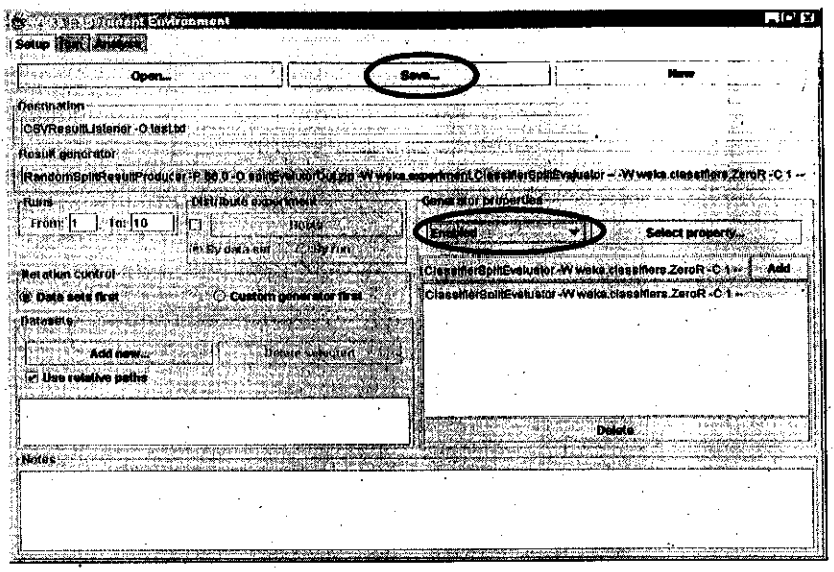
รูปที่ ข10 แสดงการกำหนด Output File

กดปุ่ม Select แล้ว ทำการปิดหน้าต่างนั้น โดยคลิกที่กากบาท มุมขวาบน และ Click ที่ปุ่ม OK ที่หน้าต่างแสดงการกำหนด output File ผลลัพธ์หรือ output file ที่ได้จะเก็บอยู่ในรูป text file เมื่อทำตามขั้นตอนดังกล่าวมาแล้วจะได้หน้าจอ ดังรูปที่ ข11



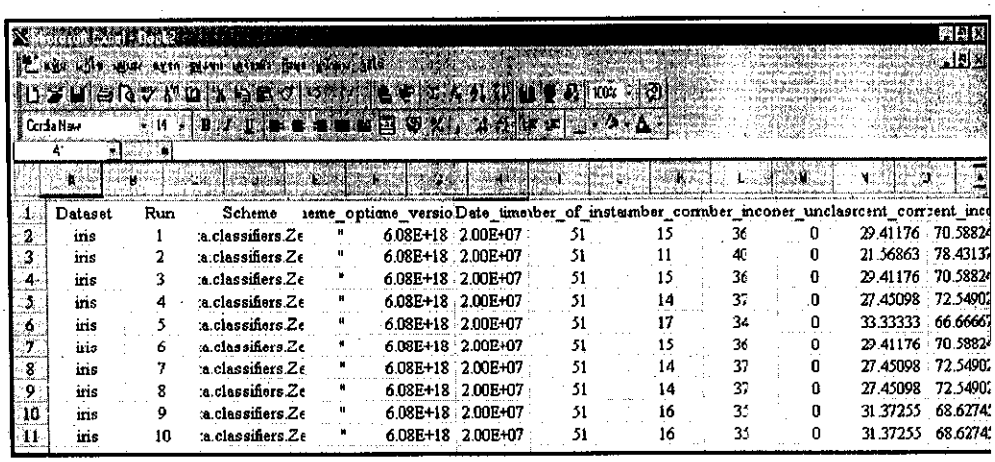
รูปที่ ข11 แสดงผลการเลือก Dataset และการกำหนด Output file

สามารถบันทึกการทำ Experiment ได้โดยการ Click ที่ปุ่ม Save จะปรากฏหน้าจอให้ตั้งชื่อ โดยให้ระบุนามสกุลเป็น .exp เมื่อทำการบันทึกแล้วเราสามารถเปิดมาใช้งานได้ใหม่โดยคลิกที่ปุ่ม openจากนั้นเมื่อเราจะทำการ Analyze เราต้องเลือกในส่วนของ Generator Properties จากที่ Disabled อยู่ให้เป็น Enabled โดยเลือก Property ที่ต้องการ ดังรูปที่ ข12



รูปที่ ข12 แสดงการเลือก Properties ที่ต้องการใช้

เมื่อเลือก Generator properties เสร็จแล้วทำการ Run ผลลัพธ์ที่ได้จะเก็บไว้ใน Output File ที่กำหนดไว้เบื้องต้น ซึ่งในการดูผลลัพธ์นั้น เราสามารถนำ Text file ที่ได้ไปเปิดด้วย Microsoft Excel ได้ผลลัพธ์ดังนี้



	Dataset	Run	Scheme	ieme_optiome_versio	Date_timember_of_instamber_commber_inconer_unclasrcent_corrcent_inc
2	iris	1	:a.classifiers.Ze	" 6.08E+18	2.00E+07 51 15 36 0 29.41176 70.5882
3	iris	2	:a.classifiers.Ze	" 6.08E+18	2.00E+07 51 11 40 0 21.56863 78.4313
4	iris	3	:a.classifiers.Ze	" 6.08E+18	2.00E+07 51 15 36 0 29.41176 70.5882
5	iris	4	:a.classifiers.Ze	" 6.08E+18	2.00E+07 51 14 37 0 27.45098 72.5490
6	iris	5	:a.classifiers.Ze	" 6.08E+18	2.00E+07 51 17 34 0 33.33333 66.6666
7	iris	6	:a.classifiers.Ze	" 6.08E+18	2.00E+07 51 15 36 0 29.41176 70.5882
8	iris	7	:a.classifiers.Ze	" 6.08E+18	2.00E+07 51 14 37 0 27.45098 72.5490
9	iris	8	:a.classifiers.Ze	" 6.08E+18	2.00E+07 51 14 37 0 27.45098 72.5490
10	iris	9	:a.classifiers.Ze	" 6.08E+18	2.00E+07 51 16 35 0 31.37255 68.6274
11	iris	10	:a.classifiers.Ze	" 6.08E+18	2.00E+07 51 16 35 0 31.37255 68.6274

รูปที่ ข13 แสดง MS Excel ที่แสดง Output ของ Dataset ที่ run แล้ว

ผู้ใช้สามารถสร้างหนึ่ง Experiment แล้ว Run ได้หลาย Scheme เพื่อนำมาเปรียบเทียบกันบนชุดของ Dataset เดียวกันแล้ววิเคราะห์ผลลัพธ์เพื่อตัดสินใจเลือก Scheme ที่เหมาะสมที่สุด

ประวัตินักวิจัย

นิตยา เกิดประสพ สำเร็จการศึกษาในระดับปริญญาเอกสาขา Computer Science จาก Nova Southeastern University เมือง Fort Lauderdale รัฐฟลอริดา สหรัฐอเมริกา เมื่อปีพุทธศักราช 2542 (ค.ศ. 1999) ด้วยทุนการศึกษาของกระทรวงวิทยาศาสตร์ฯ โดยทำวิทยานิพนธ์ระดับปริญญาเอกในหัวข้อเรื่อง "The application of inductive logic programming to support semantic query optimization" หลังสำเร็จการศึกษาได้ปฏิบัติราชการในตำแหน่งอาจารย์ ประจำสาขาคอมพิวเตอร์ ภาควิชาคณิตศาสตร์ คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีพุทธศักราช 2543 ได้มาปฏิบัติงานในตำแหน่งอาจารย์ประจำสาขาวิชาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยเทคโนโลยีสุรนารี จนถึงปัจจุบัน

นับตั้งแต่สำเร็จการศึกษามีผลงานวิจัยตีพิมพ์ในวารสารวิชาการและวารสารการประชุมวิชาการ จำนวนรวม 14 เรื่อง ในสาขาฐานข้อมูลขั้นสูง และสาขาการทำเหมืองข้อมูล ในปัจจุบันดำเนินการวิจัยเกี่ยวกับการพัฒนาระบบเหมืองข้อมูลประสิทธิภาพสูงที่สามารถทนต่อข้อมูลรบกวน และการเพิ่มความสามารถในการจัดการความรู้ของระบบเหมืองข้อมูล