วิธีที่เหมาะสมสำหรับการตัดกิ่งต้นไม้ตัดสินใจของการทำเหมืองข้อมูลทางด้านวิทยาศาสตร์

นายนฤพนต์ ว่องประชานุกูล

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์
มหาวิทยาลัยเทคโนโลยีสุรนารี
ปีการศึกษา 2548

ISBN 974-533-558-4

A PROPER METHOD FOR DECISION TREE PRUNING IN SCIENTIFIC DATA MINING

Narupon Wongprachanukul

A Thesis Submitted in Partial Fulfillment of the Requirements for the

Degree of Master of Engineering in Computer Engineering

Suranaree University of Technology

Academic Year 2005

ISBN 974-533-558-4

วิธีที่เหมาะสมสำหรับการตัดกิ่งต้นไม้ตัดสินใจของการทำเหมืองข้อมูล
ทางด้านวิทยาศาสตร์

มหาวิทยาลัยเทคโนโลยีสุรนารี อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

คณะกรรมการสอบวิทยานิพนธ์

(ผศ. ดร.พิชโยทัย มหัทธนาภิวัฒน์)
ประธานกรรมการ

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

(ผศ. ดร.กิตติศักดิ์ เกิดประสพ)
กรรมการ

(อ. ดร.ปรเมศวร์ ห่อแก้ว)
กรรมการ

(รศ. ดร.เสาวณีย์ รัตนพานี)
รองอธิการบดีฝ่ายวิชาการ

(รศ. น.อ. ดร.วรพจน์ ขำพิศ)
คณบดีสำนักวิชาวิศวกรรมศาสตร์

นฤพนต์ ว่องประชานุกูล : วิธีที่เหมาะสมสำหรับการตัดกิ่งต้นไม้ตัดสินใจของการทำเหมืองข้อมูลทางด้านวิทยาศาสตร์ (A PROPER METHOD FOR DECISION TREE PRUNING IN SCIENTIFIC DATA MINING) อาจารย์ที่ปรึกษา : รศ. ดร.นิตยา เกิดประสพ, 118 หน้า. ISBN 974-533-558-4

ต้นไม้ตัดสินใจเป็นเครื่องมือหนึ่งที่นิยมนำมาใช้ในการทำเหมืองข้อมูล ที่เกี่ยวข้องกับงานการจำแนกข้อมูล โดยโมเดลสังเคราะห์ขึ้นจากกลุ่มตัวอย่างที่เรียกว่าชุดข้อมูลฝึก ในแต่ละเรคคอร์ดประกอบด้วยแอททริบิวต์จำนวนหลายแอททริบิวต์ โดยที่มีแอททริบิวต์หนึ่งแสดงกลุ่มของตัวอย่างนั้น แต่การนำเทคนิคการสร้างต้นไม้ตัดสินใจไปใช้กับข้อมูลจริง โมเดลที่สังเคราะห์ขึ้นอาจมีความซับซ้อนมากเกินไป เนื่องจากพยายามที่จะขยายโครงสร้างให้สามารถอธิบายข้อมูลรบกวนที่อาจมีอยู่ในชุดข้อมูลฝึกให้ได้ ปัญหานี้คือการเจาะจงโมเดลกับข้อมูลมากเกินไป เทคนิคที่สำคัญสำหรับแก้ไขการเจาะจงโมเดลกับข้อมูลมากเกินไปคือใช้เทคนิคการตัดกิ่งต้นไม้ตัดสินใจ เพื่อตัดกิ่งที่มีความน่าเชื่อถือน้อยออกไปจากต้นไม้ตัดสินใจ ซึ่งจะส่งผลให้ได้โครงสร้างต้นไม้ที่ใช้เวลาในการจำแนกข้อมูลได้เร็วขึ้น และปรับปรุงความสามารถของต้นไม้ให้ใช้จำแนกข้อมูลใหม่ได้อย่างถูกต้อง

งานวิจัยนี้ได้เสนอวิธีการตัดกิ่งต้นไม้ตัดสินใจ REP+ ที่พัฒนาขึ้นเพื่อเพิ่มประสิทธิภาพของต้นไม้ตัดสินใจ โดยใช้การทดสอบทางสถิติเพื่อตรวจสอบความสัมพันธ์กันอย่างมีนัยสำคัญระหว่างกลุ่มของข้อมูลที่ได้จากการทำนายของต้นไม้ตัดสินใจ และกลุ่มของข้อมูลจากชุดข้อมูลฝึก ดำเนินการทดสอบกับข้อมูลทางด้านวิทยาศาสตร์จำนวน 21 ชุดข้อมูล จากผลการวิจัยสามารถสรุปได้ว่า ต้นไม้ตัดสินใจที่สังเคราะห์ขึ้นจากงานวิจัยนี้จะมีความซับซ้อนลดลง สามารถใช้จำแนกข้อมูลได้รวดเร็วขึ้นโดยไม่ทำให้ความแม่นตรงในการจำแนกข้อมูลลดลง

สาขาวิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อนักศึกษา _____

ปีการศึกษา 2548 ลายมือชื่ออาจารย์ที่ปรึกษา _____

ลายมือชื่ออาจารย์ที่ปรึกษาร่วม _____

NARUPON WONGPRACHANUKUL : A PROPER METHOD FOR DECISION TREE PRUNING IN SCIENTIFIC DATA MINING. THESIS ADVISOR : ASSOC. PROF. NITTAYA KERDPRASOP, Ph.D., 118 PP.
ISBN 974-533-558-4

DECISION TREE/PRUNING/STATISTICAL TEST

Decision tree is one of the tools used for data mining. The main application area is classification task. The model is built from a set of records, called training set. Each record consists of a number of attribute-value pairs. One of these attributes represents class of the record. When a decision tree is built, many of the branches may be overly expanded due to noise or outliers in the training set. The built model is too complex, since it tries to classify all records in the training set including noise and outliers. This problem is called "overfitting". We use tree pruning method to remove the least reliable branches, generally resulting in faster classification and improvement in the ability of the tree to correctly classify unknown data.

This research proposed a new method for decision tree pruning, called REP+. We used the statistical test to check the significant dependency between predicted classes and actual classes in the training set. We conduct the experiments on 21 scientific data sets. The pruned trees result in reduced model complexity and faster classification while maintaining their predictive accuracy.

School of Computer Engineering

Academic Year 2005

Student's Signature Narupon

Advisor's Signature

Co-advisor's Signature

กิตติกรรมประกาศ

วิทยานิพนธ์นี้สำเร็จลุล่วงด้วยดี ผู้วิจัยขอกราบขอบพระคุณบุคคลและคณะบุคคลต่าง ๆ ที่ได้กรุณาให้คำปรึกษา แนะนำและช่วยเหลืออย่างดียิ่ง ทั้งในด้านวิชาการและด้านการดำเนินงานวิจัยดังต่อไปนี้

รองศาสตราจารย์ ดร.นิตยา เกิดประสพ อาจารย์ที่ปรึกษาวิทยานิพนธ์

ผู้ช่วยศาสตราจารย์ ดร.กิตติศักดิ์ เกิดประสพ อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม และคณาจารย์ทุกท่านในสาขาวิชาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยเทคโนโลยีสุรนารี

ขอขอบคุณ คณะผู้บริหารของศูนย์ปฏิบัติการวิจัยเครื่องกำเนิดแสงซินโครตรอนแห่งชาติ ที่ได้ให้โอกาสผู้วิจัยลาศึกษาต่อในระดับสูงขึ้น พี่ ๆ และเพื่อน ๆ พนักงานทุกท่านที่ให้กำลังใจและคอยติดตามความก้าวหน้าของผู้วิจัยตลอดมา

ขอขอบคุณ เพื่อน ๆ บัณฑิตศึกษาทุกท่าน ที่ให้คำปรึกษา กำลังใจ และให้ความช่วยเหลืออย่างดีจนงานวิจัยสำเร็จลุล่วงด้วยดี

ขอขอบคุณ คุณวันวิสาข์ สีทานันท์ ที่คอยให้กำลังใจ เอาใจใส่และช่วยเหลือผู้วิจัยในทุกเรื่องอย่างดียิ่ง

สุดท้ายนี้ ขอกราบขอบพระคุณบิดา มารดา ที่ให้การเลี้ยงดูอบรมและส่งเสริมการศึกษาเป็นอย่างดีมาโดยตลอด จนทำให้ผู้วิจัยประสบความสำเร็จในชีวิตตลอดมา

นฤพนต์ ว่องประชานุกูล

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

ตารางที่ **หน้า**

สารบัญรูป

รูปที่ **หน้า**

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มาของปัญหาการวิจัย

ในปัจจุบันการวิเคราะห์และการค้นหาความรู้จากข้อมูลโดยอัตโนมัติ เป็นงานที่จำเป็นสำหรับองค์กร เนื่องจากข้อมูลที่ถูกบันทึกอยู่ในรูปแบบอิเล็กทรอนิกส์มีปริมาณสูงขึ้น ไม่สามารถใช้แรงงานคนหรือโปรแกรมทางสถิติวิเคราะห์ข้อมูลเพื่อนำมาใช้ประโยชน์ได้ทันเวลา จำเป็นต้องแก้ไขปัญหาความล่าช้านี้ โดยทำให้กระบวนการวิเคราะห์ข้อมูลเป็นอัตโนมัติมากขึ้น ลดขั้นตอนการควบคุมและสั่งงานจากผู้เชี่ยวชาญให้น้อยลง โดยให้ระบบคอมพิวเตอร์ทำหน้าที่ค้นหาแนวโน้มต่าง ๆ ที่น่าสนใจ จากข้อมูลหรือวิเคราะห์ความสัมพันธ์ระหว่างข้อมูลได้ด้วยความสามารถของระบบเอง

กระบวนการที่กระทำกับข้อมูลจำนวนมาก เพื่อค้นหารูปแบบและความสัมพันธ์ที่ซ่อนอยู่ในชุดข้อมูลนั้นเรียกว่า การทำเหมืองข้อมูล (data mining) ในปัจจุบันการทำเหมืองข้อมูลได้ถูกนำไปประยุกต์ใช้ในงานหลายประเภท ทั้งในด้านธุรกิจที่ช่วยในการตัดสินใจของผู้บริหาร ในด้านวิทยาศาสตร์และการแพทย์ เช่น โครงการวิเคราะห์สารพันธุกรรมของมนุษย์ ระบบสารสนเทศภูมิศาสตร์ (Geographical Information Systems) ใช้ค้นหาผลข้างเคียงของการใช้ยาโดยอาศัยข้อมูลจากแฟ้มประวัติของผู้ป่วย เป็นต้น รวมทั้งในด้านเศรษฐกิจและสังคม (นิตยา เกิดประสพ, 2547)

เทคนิคหนึ่งที่สำคัญของการทำเหมืองข้อมูลคือ การค้นหารูปแบบหรือโมเดลเพื่อการจำแนกข้อมูล (classification) โดยใช้วิธีการสร้างต้นไม้ตัดสินใจ (decision tree) ซึ่งนับว่าเป็นวิธีการเรียนรู้ที่นิยมใช้มากที่สุดรูปแบบหนึ่งในการเรียนรู้ของเครื่อง (machine learning) การเรียนรู้แบบนี้เป็นการเรียนรู้โดยการแยกแยะข้อมูลออกเป็นกลุ่ม โดยใช้คุณสมบัติหรือแอททริบิวต์ (attribute) ของข้อมูลในการแยกแยะ โดยคุณสมบัติแต่ละประเภทของข้อมูลจะมีความสำคัญมากน้อยต่างกัน ซึ่งเป็นประโยชน์ช่วยให้ผู้ใช้สามารถวิเคราะห์ข้อมูลและตัดสินใจได้ถูกต้องยิ่งขึ้น การเรียนรู้ด้วยต้นไม้ตัดสินใจเป็นเทคนิคที่ได้รับความสนใจ เนื่องจากการแสดงผลของการวิเคราะห์ข้อมูลสามารถทำความเข้าใจได้ง่าย ใช้เวลาน้อยในการวิเคราะห์ข้อมูล และให้ค่าความถูกต้องที่ดีในการจำแนกกลุ่มข้อมูลและทำนายประเภทข้อมูลที่จะเกิดขึ้นในอนาคต (Murthy, 1998)

แต่การนำเทคนิคการวิเคราะห์ข้อมูลอัตโนมัติแบบนี้ไปใช้กับข้อมูลจริงมักจะพบกับปัญหาข้อมูลผิดเพี้ยน ซึ่งอาจเกิดจากการบันทึกข้อมูลผิดพลาดหรือการสูญหายไปบางส่วนของข้อมูล ข้อมูลผิดเพี้ยนแบบนี้เรียกว่าข้อมูลรบกวน (noise) ซึ่งทำให้ต้นไม้ตัดสินใจที่สังเคราะห์ขึ้นจะวิเคราะห์ข้อมูลผิดพลาดและอาจจะสร้างต้นไม้ตัดสินใจที่มีขนาดใหญ่และซับซ้อนเกินไป เนื่องจากพยายามที่จะขยายโครงสร้างให้สามารถอธิบายข้อมูลรบกวนเหล่านั้นให้ได้

งานวิจัยนี้เน้นการพัฒนาและการทดสอบกับข้อมูลทางด้านวิทยาศาสตร์ ที่เกี่ยวข้องกับข้อมูลทางด้านการแพทย์และการศึกษาโครงสร้างทางพันธุกรรมของมนุษย์ ซึ่งเกิดจากการสังเกตการวิเคราะห์และการทดลอง ข้อมูลประเภทนี้มีลักษณะแตกต่างจากข้อมูลประเภทอื่น ๆ เนื่องจากปริมาณข้อมูลที่นำมาวิเคราะห์อาจจะมีจำนวนตัวอย่างค่อนข้างมาก และเป็นข้อมูลที่มีรายละเอียดของคุณสมบัติจำนวนมากในแต่ละตัวอย่าง ซึ่งอาจเกิดความผิดพลาดในการบันทึกหรือมีข้อมูลรบกวนปะปนอยู่ในชุดข้อมูล

ดังนั้นงานวิจัยนี้ จึงถูกเสนอขึ้นเพื่อศึกษาวิเคราะห์เทคนิคในการจัดการกับข้อมูลรบกวนและคิดค้นเทคนิคในการจัดการกับข้อมูลรบกวนให้ได้อย่างมีประสิทธิภาพ โดยพัฒนาเทคนิคการตัดกิ่งต้นไม้ตัดสินใจที่เหมาะสม เพื่อตัดกิ่งที่มีความน่าเชื่อถือน้อยออกไปจากต้นไม้ตัดสินใจที่เติบโตเต็มที่ ซึ่งจะส่งผลให้ได้โครงสร้างต้นไม้ที่มีความซับซ้อนลดลง และยังคงสามารถใช้จำแนกข้อมูลใหม่ได้อย่างถูกต้อง นอกจากนี้ยังจะใช้เป็นแนวทางสำคัญในการพัฒนาการทำเหมืองข้อมูลเฉพาะทางด้านวิทยาศาสตร์ได้ต่อไปในอนาคต และสามารถนำความรู้ที่ได้จากงานวิจัยนี้พัฒนาเทคนิคการตัดกิ่งต้นไม้ตัดสินใจที่เหมาะสมกับข้อมูลทุกกลุ่ม โดยไม่จำกัดอยู่เฉพาะข้อมูลทางด้านวิทยาศาสตร์เท่านั้น

1.2 วัตถุประสงค์ของการวิจัย

1.2.1 เพื่อศึกษาค้นคว้าวิธีการตัดกิ่งต้นไม้ตัดสินใจที่น่าสนใจ เปรียบเทียบข้อดีและข้อเสียของแต่ละวิธี และสามารถนำเอาวิธีการต่าง ๆ มาประยุกต์ใช้เพื่อพัฒนาวิธีการตัดกิ่งต้นไม้ตัดสินใจที่มีประสิทธิภาพ

1.2.2 เพื่อศึกษาค้นคว้า พัฒนาวิธีการตัดกิ่งต้นไม้ตัดสินใจที่สามารถใช้งานได้ดีกับข้อมูลทางด้านวิทยาศาสตร์

1.2.3 เพื่อพัฒนาวิธีการตัดกิ่งต้นไม้ตัดสินใจที่เหมาะสม ทำให้ได้โครงสร้างต้นไม้ที่มีความซับซ้อนลดลงและยังคงสามารถใช้จำแนกข้อมูลใหม่ได้อย่างถูกต้อง

1.3 ขอบเขตของการวิจัย

1.3.1 ในขั้นตอนการสร้างต้นไม้ตัดสินใจ ใช้อัลกอริทึม C4.5 (หรือ J48) ที่พัฒนาบนระบบเปิดเผยซอร์สโค้ดที่ชื่อ WEKA

1.3.2 ศึกษาค้นคว้าเฉพาะวิธีการตัดกิ่งต้นไม้ตัดสินใจชนิด post-pruning ที่ขั้นตอนการตัดกิ่งเริ่มทำงานหลังจากต้นไม้ตัดสินใจได้สร้างขึ้นสมบูรณ์แล้ว

1.3.3 งานวิจัยนี้เน้นการพัฒนาและทดสอบกับข้อมูลเฉพาะทางด้านวิทยาศาสตร์

1.3.4 วิธีการตัดกิ่งต้นไม้ตัดสินใจที่นำมาเปรียบเทียบกับวิธีที่พัฒนาขึ้นประกอบด้วย วิธีการตัดกิ่งแบบความผิดพลาดลดลง (Reduced-error pruning) และวิธีการตัดกิ่งโดยใช้ค่าความผิดพลาด (Error-based pruning)

1.3.5 เกณฑ์ที่ใช้ศึกษาเพื่อเปรียบเทียบวิธีการตัดกิ่งต้นไม้ตัดสินใจที่เหมาะสมกับข้อมูลทางด้านวิทยาศาสตร์ประกอบด้วย

1) เวลาที่ใช้ในการสร้างต้นไม้ตัดสินใจ
2) ขนาดของต้นไม้ที่ได้หลังจากผ่านการตัดกิ่งแล้ว
3) ความแม่นตรง (accuracy) ในการจำแนกกลุ่มข้อมูล

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1.4.1 ทำให้เกิดการพัฒนาเทคนิคการตัดกิ่งต้นไม้ตัดสินใจที่มีประสิทธิภาพสูงขึ้น

1.4.2 ทำให้ได้โครงสร้างต้นไม้ตัดสินใจที่มีความซับซ้อนลดลง จากการที่สามารถลดจำนวนโหนดที่ไม่มีความสำคัญกับการจำแนกกลุ่มของข้อมูลออกไปได้

1.4.3 ทำให้ได้โครงสร้างต้นไม้ตัดสินใจที่ยังคงมีความแม่นยำในการจำแนกกลุ่มของข้อมูลใหม่ได้อย่างถูกต้อง

1.4.4 ความรู้ที่ได้จากงานวิจัยนี้ สามารถใช้เป็นแนวทางในการพัฒนาการทำเหมืองข้อมูลเฉพาะทางด้านวิทยาศาสตร์ได้ต่อไปในอนาคต

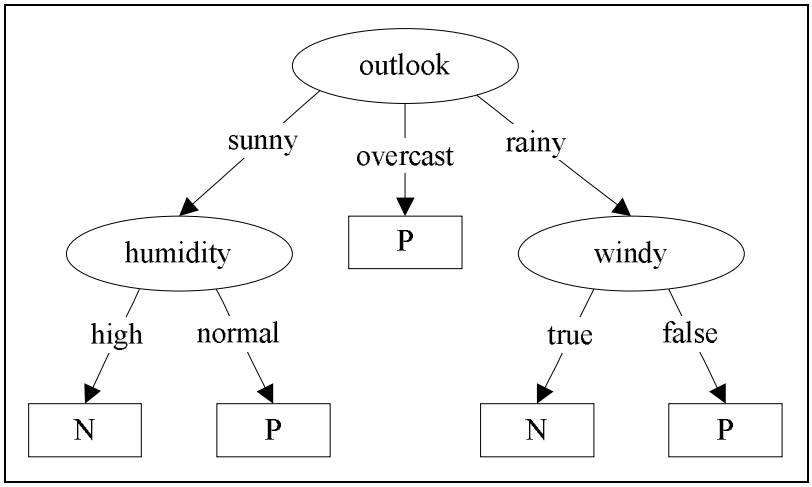
บทที่ 2
ปริทัศน์วรรณกรรมและงานวิจัยที่เกี่ยวข้อง

การวิเคราะห์และค้นหาความรู้จากข้อมูลโดยอัตโนมัติโดยการใช้ต้นไม้ตัดสินใจ โดยมีจุดมุ่งหมายเพื่อจำแนกกลุ่มของข้อมูลและอธิบายลักษณะของข้อมูลในแต่ละกลุ่ม มีรายละเอียดของทฤษฎีและงานวิจัยที่เกี่ยวข้อง ได้แก่ การสร้างต้นไม้ตัดสินใจ การเลือกแอททริบิวต์เพื่อการจำแนกกลุ่มของข้อมูล วิธีการตัดกิ่งต้นไม้ตัดสินใจและการศึกษาเปรียบเทียบวิธีการตัดกิ่งต้นไม้ตัดสินใจ รวมทั้งทฤษฎีการตัดสินใจเชิงสถิติที่เกี่ยวข้องกับสมมติฐานเชิงสถิติ

2.1 การสร้างต้นไม้ตัดสินใจ

ต้นไม้ตัดสินใจ (decision tree) คือต้นไม้ที่ใช้สนับสนุนการตัดสินใจ มีลักษณะเป็นโครงสร้างคล้ายกับต้นไม้หัวกลับที่มีรากอยู่ด้านบนและใบอยู่ด้านล่างสุด โดยที่ภายในต้นไม้จะประกอบไปด้วยโหนด (node) ซึ่งแต่ละโหนดนั้นจะแสดงถึงการทดสอบหรือการตัดสินใจบนข้อมูลของคุณสมบัติหรือแอททริบิวต์ต่าง ๆ กิ่งของต้นไม้แสดงถึงค่าหรือผลลัพธ์ที่ได้จากการทดสอบ และใบซึ่งเป็นสิ่งที่อยู่ล่างสุดของต้นไม้ตัดสินใจแสดงถึงกลุ่มของข้อมูล (class) หรือผลลัพธ์ที่ได้จากการทำนาย โหนดที่อยู่บนสุดของต้นไม้เรียกว่าโหนดราก (root node) ดังแสดงโครงสร้างของต้นไม้ตัดสินใจได้ดังรูปที่ 2.1 ซึ่งแสดงถึงต้นไม้ที่ใช้ในการตัดสินใจว่าจะออกไปเล่นกอล์ฟหรือไม่ (Quinlan, 1986) โดยพิจารณาจากสภาพอากาศต่าง ๆ ประกอบการตัดสินใจ โดยโหนดที่แสดงในรูปวงรีแสดงถึงการทดสอบค่าที่เป็นไปได้ของแอททริบิวต์นั้น ๆ และใบที่แสดงในรูปสี่เหลี่ยมจะแสดงการจำแนกกลุ่มของข้อมูล ซึ่งเป็นผลลัพธ์ของการทำนายว่าจะออกไปเล่นกอล์ฟ (P) หรือไม่ออกไปเล่น (N) จากการทดสอบตามเส้นทางของต้นไม้ตัดสินใจ

ในการจำแนกข้อมูลที่ได้รับเข้ามาใหม่นั้น ค่าของแอททริบิวต์ต่าง ๆ ของข้อมูลเหล่านั้นจะถูกทดสอบด้วยต้นไม้ตัดสินใจ โดยจะเริ่มต้นการทดสอบตั้งแต่โหนดรากไปจนถึงใบ โดยที่ใบจะแสดงถึงกลุ่มของการทำนายข้อมูลนั้น



รูปที่ 2.1 ต้นไม้ตัดสินใจที่ใช้ในการตัดสินใจการออกไปเล่นกอล์ฟ

โดยหลักการพื้นฐานของการสร้างต้นไม้ตัดสินใจ จะเป็นการสร้างในลักษณะจากบนลงล่าง (top-down) คือจะเริ่มจากการสร้างรากของต้นไม้ก่อนแล้วจึงแตกกิ่งไปจนถึงใบ โดยจะแสดงขั้นตอนการสร้างต้นไม้ตัดสินใจได้ดังนี้ (Han and Kamber, 2001)

1) ต้นไม้จะเริ่มต้นโดยมีโหนดเพียงโหนดเดียวแสดงถึงชุดข้อมูลฝึก (training set)
2) ถ้าข้อมูลทั้งหมดอยู่ในกลุ่มเดียวกันแล้ว ให้โหนดนั้นเป็นใบและตั้งชื่อตามกลุ่มของข้อมูลนั้น
3) ถ้าในโหนดมีข้อมูลหลายกลุ่มปะปนอยู่ จะต้องวัดค่าเกน (gain) ของแต่ละแอททริบิวต์เพื่อที่จะใช้เป็นเกณฑ์ในการคัดเลือกแอททริบิวต์ ที่มีความสามารถในการแบ่งแยกข้อมูลออกเป็นกลุ่มต่าง ๆ ได้ดีที่สุด โดยแอททริบิวต์ที่มีค่าเกนมากที่สุดจะถูกเลือกให้เป็นตัวทดสอบหรือแอททริบิวต์ที่ใช้ในการตัดสินใจ โดยแสดงในรูปของโหนดบนต้นไม้
4) กิ่งของต้นไม้จะถูกสร้างขึ้นจากค่าต่าง ๆ ที่เป็นไปได้ของโหนดทดสอบ และข้อมูลจะถูกแบ่งออกตามกิ่งต่าง ๆ ที่สร้างขึ้น
5) ทำการวนซ้ำเพื่อหาแอททริบิวต์ที่มีค่าเกนมากที่สุด สำหรับข้อมูลที่ถูกแบ่งแยกออกมาในแต่ละกิ่งเพื่อนำแอททริบิวต์นี้มาสร้างเป็นโหนดตัดสินใจต่อไป โดยที่แอททริบิวต์ที่ถูกเลือกมาเป็นโหนดแล้วจะไม่ถูกเลือกมาอีก สำหรับโหนดในระดับต่อ ๆ ไป
6) ทำการวนซ้ำเพื่อแบ่งข้อมูลและแตกกิ่งของต้นไม้ไปเรื่อย ๆ โดยการวนซ้ำจะสิ้นสุดก็ต่อเมื่อเงื่อนไขข้อใดข้อหนึ่งต่อไปนี้เป็นจริง

(1) ถ้าข้อมูลทุกตัวในโหนดนั้นอยู่ในกลุ่มเดียวกัน ให้สร้างใบตามกลุ่มของข้อมูลนั้น

(2) ถ้าไม่เหลือแอททริบิวต์ใดสำหรับใช้ในการแบ่งข้อมูลแล้ว ซึ่งในกรณีนี้จะใช้กลุ่มที่มีข้อมูลสนับสนุนมากที่สุดเป็นใบ

(3) ถ้าไม่มีข้อมูลสนับสนุนสำหรับกิ่งนั้น ๆ แล้ว ให้สร้างใบตามกลุ่มที่มีข้อมูลสนับสนุนมากที่สุด

ในขั้นตอนการสร้างต้นไม้ตัดสินใจ อัลกอริทึม C4.5 เป็นอัลกอริทึมที่มีชื่อเสียงและเป็นที่รู้จักกันอย่างแพร่หลาย พัฒนาโดย Quinlan (1993) ที่ได้พัฒนาต่อมาจากอัลกอริทึม ID3 ที่เขาได้พัฒนาขึ้น (Quinlan, 1986) เป็นวิธีการเรียนรู้จากกลุ่มตัวอย่างที่เรียกว่า ชุดข้อมูลฝึก (training set) ที่อาศัยวิธีการจัดหมวดหมู่เพื่อสร้างต้นไม้ตัดสินใจ

ชุดข้อมูลฝึกจะมีลักษณะคล้ายกับข้อมูลในฐานข้อมูลเชิงสัมพันธ์ (relational database) แสดงในรูปของตารางที่ประกอบด้วย แถวแสดงข้อมูลหรือตัวอย่าง และคอลัมน์แสดงแอททริบิวต์ของข้อมูล ซึ่งแบ่งออกเป็น 2 ชนิดคือ

1) แอททริบิวต์ที่เป็นจุดมุ่งหมาย (goal attribute) ของการจำแนกกลุ่มข้อมูล เป็นแอททริบิวต์ที่กำหนดว่าตัวอย่างนั้น ๆ ถูกจัดอยู่ในกลุ่มไหน โดยจะมีเพียงแอททริบิวต์เดียวในแต่ละชุดข้อมูล และข้อมูลจะเป็นชนิดข้อความเท่านั้น

2) แอททริบิวต์ประกอบการทำนาย (predicting attribute) เป็นแอททริบิวต์ที่บ่งบอกถึงคุณสมบัติต่าง ๆ ของตัวอย่างแต่ละตัวอย่าง โดยแต่ละแอททริบิวต์อาจมีข้อมูลเป็นชนิดข้อความหรือตัวเลขก็ได้

จากตารางที่ 2.1 เป็นตัวอย่างชุดข้อมูลฝึกที่ใช้ประกอบการตัดสินใจออกไปเล่นกอล์ฟ โดยพิจารณาจากสภาพอากาศต่าง ๆ ประกอบการตัดสินใจ (Quinlan, 1986) เมื่อนำมาสร้างเป็นต้นไม้ตัดสินใจสามารถแสดงโครงสร้างของต้นไม้ได้ดังรูปที่ 2.1 โดยชุดข้อมูลฝึกนี้ประกอบด้วยแอททริบิวต์ Class เป็นแอททริบิวต์ที่เป็นจุดมุ่งหมาย มีค่าที่เป็นไปได้คือ P หรือ N แอททริบิวต์ outlook, temperature, humidity และ windy เป็นแอททริบิวต์ประกอบการทำนายของชุดข้อมูล

ประสิทธิภาพของต้นไม้ตัดสินใจไม่ได้อยู่ที่การสร้างต้นไม้ตัดสินใจ เพื่อให้สามารถจัดกลุ่มชุดข้อมูลฝึกได้อย่างถูกต้องเท่านั้น แต่ต้องสามารถจัดกลุ่มข้อมูลจากตัวอย่างใหม่ ๆ ที่นอกเหนือจากชุดข้อมูลฝึกได้อย่างถูกต้องด้วย ดังนั้นการสร้างต้นไม้ตัดสินใจจึงควรจะมีชุดข้อมูลทดสอบ (test set) ที่จะใช้ตรวจสอบความถูกต้องของต้นไม้ตัดสินใจด้วย

ตารางที่ 2.1 ชุดข้อมูลฝึกที่ใช้ประกอบการตัดสินใจเล่นกอล์ฟ

ID	Attributes				Class
	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

2.2 การคัดเลือกแอททริบิวต์เพื่อจำแนกกลุ่มของข้อมูล

ในการสร้างต้นไม้ตัดสินใจ ปัญหาสำคัญที่ต้องพิจารณาคือ ควรจะตัดสินใจเลือกแอททริบิวต์ใดมาทำหน้าที่เป็นโหนดราก ในแต่ละขั้นตอนของการสร้างต้นไม้และต้นไม้ย่อย (subtree) ของต้นไม้ตัดสินใจ เกณฑ์ที่ใช้ช่วยประกอบการเลือกแอททริบิวต์คือการคำนวณค่าเกน (gain) ซึ่งเป็นค่าที่บ่งบอกว่าแอททริบิวต์นั้นจะสามารถจำแนกกลุ่มของข้อมูลได้ดีเพียงใด โดยทดลองเลือกแต่ละแอททริบิวต์ที่เป็นไปได้จากชุดข้อมูลมาทำหน้าที่เป็นโหนดราก ถ้าแอททริบิวต์ใดให้ค่าเกนที่สูงที่สุด แสดงว่าแอททริบิวต์นั้นสามารถจำแนกกลุ่มของข้อมูลได้ดีที่สุด หรือเป็นแอททริบิวต์ที่จัดกลุ่มของข้อมูลแล้ว ได้ข้อมูลในแต่ละใบของต้นไม้เป็นกลุ่มเดียวกันทั้งหมด หรือมีข้อมูลต่างกลุ่มปะปนมาบ้างเพียงเล็กน้อยเท่านั้น โดยค่าเกนสำหรับการเลือกแอททริบิวต์ที่สำคัญแสดงได้ดังนี้

2.2.1 ค่ามาตรฐานเกน (Gain criterion)

วิธีการสร้างต้นไม้ตัดสินใจโดยใช้อัลกอริทึม ID3 จะใช้ค่ามาตรฐานเกนในการตัดสินใจเลือกแอททริบิวต์ที่จะใช้เป็นโหนดรากของต้นไม้หรือของต้นไม้ย่อย โดยการคำนวณค่า

เกนของแต่ละแอททริบิวต์เมื่อใช้แบ่งกลุ่มตัวอย่าง และเลือกแอททริบิวต์ที่มีค่าเกนสูงที่สุดมาเป็นโหนดรากซึ่งแอททริบิวต์นี้จะมีความสามารถในการจำแนกกลุ่มข้อมูลสูง โดยที่ต้องการข้อมูลจำนวนน้อยที่สุดในการที่จะระบุว่าข้อมูลนั้นอยู่ในกลุ่มใด และการคัดเลือกแอททริบิวต์นี้ทำให้สามารถแบ่งข้อมูลออกมาโดยที่มีการปะปนกันของกลุ่มที่ต่างกันเกิดขึ้นน้อยอีกด้วย ค่าเกนนี้คำนวณได้โดยใช้ความรู้จากทฤษฎีสารสนเทศ (information theory) ซึ่งมีสาระสำคัญคือ ค่าสารสนเทศของข้อมูลจะขึ้นอยู่กับความน่าจะเป็นของข้อมูล ซึ่งสามารถวัดอยู่ในรูปของบิต (bits) เขียนเป็นสมการได้ดังนี้

ค่าสารสนเทศของข้อมูล = $-\log_2$ (ความน่าจะเป็นของข้อมูล) (2-1)

การใช้ค่า information gain จะช่วยลดจำนวนครั้งของการทดสอบในการแยกแยะข้อมูล อีกทั้งยังรับประกันว่าต้นไม้ตัดสินใจที่ได้จะไม่มีความซับซ้อนมากจนเกินไป ซึ่งค่า information gain นั้นสามารถคำนวณได้จากสมการดังต่อไปนี้ (Han and Kamber, 2001) กำหนดให้

S เป็นเซตของข้อมูลซึ่งประกอบด้วยข้อมูล s เรคคอร์ด

m เป็นจำนวนกลุ่มทั้งหมดที่ต่างกันของข้อมูลชุดนั้น

C_i แทนกลุ่มในลำดับที่ i โดยที่ i มีค่าระหว่าง 1 ถึง m

s_i แทนจำนวนข้อมูลที่เป็นสมาชิกของ S และอยู่ในกลุ่ม C_i

s_{ij} แทนจำนวนข้อมูลที่เป็นสมาชิกของ S ในกลุ่ม C_i จากการแบ่งข้อมูลด้วยค่าที่เป็นไปได้ j ของแอททริบิวต์ A โดยที่ j มีค่าระหว่าง 1 ถึง v

s_i/s แทนค่าความน่าจะเป็นที่ข้อมูลจะอยู่ในกลุ่ม C_i

ค่า information gain ที่ต้องการสำหรับจำแนกข้อมูลออกเป็นแต่ละกลุ่มหาได้โดย

$$I(s_1, s_2, ..., s_m) = -\sum_{i=1}^{m} \frac{s_i}{s} \log_2 \frac{s_i}{s} \quad (2\text{-}2)$$

ค่า entropy ของแอททริบิวต์ A ซึ่งมีค่าของแอททริบิวต์เป็น (a_1, a_2, a_3,..., a_v) หาได้โดย

$$E(A) = \sum_{j=1}^{v} \frac{s_{1j} + ... + s_{mj}}{s} I(s_{1j}, ..., s_{mj}) \quad (2\text{-}3)$$

ค่ามาตรฐานเกนที่จะใช้พิจารณาเลือกแอททริบิวต์ A มาเป็นโหนดของต้นไม้มีค่าเท่ากับ ปริมาณ

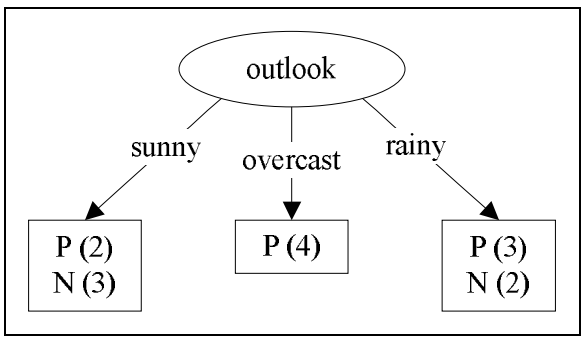
ข้อมูลที่ต้องการเพื่อให้สามารถจำแนกกลุ่มของข้อมูลได้ ลบด้วยปริมาณข้อมูลที่ต้องการเพื่อการจำแนกกลุ่มของข้อมูลโดยใช้แอททริบิวต์ A เป็นตัวตรวจสอบเพื่อจำแนกกลุ่มของข้อมูล เขียนเป็นสมการได้ดังนี้

$$Gain(A) = I(s_1, s_2, ..., s_m) - E(A) \quad (2\text{-}4)$$

จากตัวอย่างข้อมูลสภาพอากาศประกอบการตัดสินใจเล่นกอล์ฟในตารางที่ 2.1 เซตของข้อมูลฝึก T ประกอบด้วยข้อมูลจำนวน 14 เรคคอร์ด แบ่งข้อมูลออกเป็น 2 กลุ่มคือ ข้อมูลที่ตัดสินใจออกไปเล่นกอล์ฟ (Class = P) จำนวน 9 เรคคอร์ด และตัดสินใจไม่ออกไปเล่นกอล์ฟ (Class = N) จำนวน 5 เรคคอร์ด การจะระบุว่าข้อมูลหนึ่งเรคคอร์ดอยู่ในกลุ่ม P หรือ N ต้องการปริมาณข้อมูลประกอบการตัดสินใจเพื่อจำแนกกลุ่มโดยใช้สมการที่ 2-2 ดังนี้

$$I(T) = -(9/14) \times \log_2(9/14) - (5/14) \times \log_2(5/14)$$
$$= 0.940 \text{ บิต}$$

การจะจำแนกกลุ่มของข้อมูลเพื่อตัดสินใจออกไปเล่นกอล์ฟหรือไม่นั้น ต้องใช้ข้อมูลจากแอททริบิวต์อื่นประกอบการตัดสินใจ ถ้าแบ่งข้อมูลชุดนี้โดยใช้แอททริบิวต์ outlook จะสามารถจำแนกกลุ่มของข้อมูลได้ดังรูปที่ 2.2 โดยได้แสดงจำนวนเรคคอร์ดของแต่ละกลุ่มข้อมูลไว้ในวงเล็บด้วย เมื่อแบ่งตามค่าที่เป็นไปได้จะต้องการปริมาณข้อมูลเพิ่มเพื่อประกอบการเลือกกลุ่ม และสามารถคำนวณค่า entropy ของแอททริบิวต์ได้โดยใช้สมการที่ 2-3



รูปที่ 2.2 แสดงการจำแนกกลุ่มของข้อมูลโดยใช้แอททริบิวต์ outlook

$$E(outlook) = (5/14) \times (-(2/5) \times \log_2(2/5) - (3/5) \times \log_2(3/5)) + (4/14) \times (-(4/4) \times \log_2(4/4) - (0/4) \times \log_2(0/4)) + (5/14) \times (-(3/5) \times \log_2(3/5) - (2/5) \times \log_2(2/5))$$

$$= 0.693 \text{ บิต}$$

นั่นคือถ้าต้องการจำแนกกลุ่มของข้อมูลใหม่ โดยใช้แอททริบิวต์ outlook เป็นตัวตรวจสอบเพื่อจำแนกกลุ่มของข้อมูล การพิจารณาจากค่า outlook ของข้อมูลใหม่นี้ จะต้องใช้ข้อมูลเพิ่มอีก 0.693 บิต จึงจะบอกกลุ่มที่ถูกต้องของข้อมูลใหม่นี้ได้

ดังนั้นสามารถคำนวณค่าเกนจากการเลือกแอททริบิวต์ outlook เป็นแอททริบิวต์เพื่อใช้แบ่งข้อมูลได้จากสมการที่ 2-4 ดังนี้

$$Gain(outlook) = I(T) - E(outlook)$$

$$= 0.940 - 0.693$$

$$= 0.247 \text{ บิต}$$

แอททริบิวต์ที่เหลือที่สามารถถูกเลือกมาเป็นตัวทดสอบเพื่อจัดกลุ่มของข้อมูลฝึกคือแอททริบิวต์ temperature, humidity และ windy สามารถคำนวณค่าเกนจากการเลือกแต่ละแอททริบิวต์ได้ดังนี้

$$Gain(temperature) = I(T) - E(temperature)$$

$$= 0.940 - 0.911$$

$$= 0.029 \text{ บิต}$$

$$Gain(humidity) = I(T) - E(humidity)$$

$$= 0.940 - 0.788$$

$$= 0.152 \text{ บิต}$$

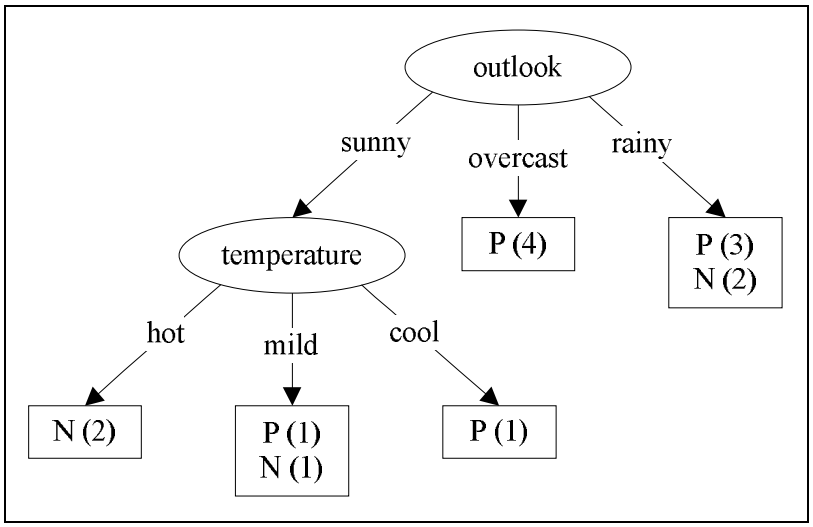
$$Gain(windy) = I(T) - E(windy)$$

$$= 0.940 - 0.892$$

$$= 0.048 \text{ บิต}$$

จะเห็นว่าแอททริบิวต์ที่ให้ค่าเกนสูงที่สุดคือ outlook ดังนั้นแอททริบิวต์ outlook จึงถูกเลือกเป็นโหนดรากของต้นไม้ตัดสินใจ แต่เนื่องจากยังไม่สามารถจัดกลุ่มของข้อมูลให้เป็นกลุ่มเดียวกันได้ทั้งหมด จึงต้องสร้างต้นไม้ตัดสินใจต่อไป โดยพิจารณาเลือกแอททริบิวต์ที่จะมาเป็นโหนดในระดับที่ 2 ต่อจากโหนดรากเพื่อจัดกลุ่มของข้อมูล ในกรณี outlook = overcast ไม่ต้องสร้างโหนดเพิ่มเติมอีกเนื่องจากสามารถจัดกลุ่มของข้อมูลที่เป็นกลุ่ม P ได้ทั้งหมดแล้ว

แอททริบิวต์ที่สามารถถูกเลือกเป็นโหนดในระดับที่ 2 ได้ประกอบด้วย temperature, humidity และ windy โดยที่แอททริบิวต์ outlook จะไม่ถูกเลือกมาอีกสำหรับโหนดในระดับต่อไป เมื่อพิจารณาการสร้างโหนดลูกทางด้านซ้ายมือ (outlook = sunny) ถ้าเลือกแอททริบิวต์ temperature จะสามารถจำแนกกลุ่มของข้อมูลได้ดังรูปที่ 2.3 และสามารถคำนวณค่าเกนได้ดังต่อไปนี้



รูปที่ 2.3 แสดงการจำแนกกลุ่มของข้อมูลโดยใช้แอททริบิวต์ temperature เป็นโหนดระดับที่ 2

$$I(\text{outlook} = \text{sunny}) = -(2/5) \times \log_2(2/5) - (3/5) \times \log_2(3/5)$$
$$= 0.971 \text{ บิต}$$

$$E_{\text{temperature}}(\text{outlook} = \text{sunny}) = (2/5) \times (-(0/2) \times \log_2(0/2) - (2/2) \times \log_2(2/2))$$
$$+ (2/5) \times (-(1/2) \times \log_2(1/2) - (1/2) \times \log_2(1/2))$$
$$+ (1/5) \times (-(1/1) \times \log_2(1/1) - (0/1) \times \log_2(0/1))$$
$$= 0.4 \text{ บิต}$$

$$\text{Gain(temperature)} = I(\text{outlook} = \text{sunny}) - E_{\text{temperature}}(\text{outlook} = \text{sunny})$$
$$= 0.971 - 0.4$$
$$= 0.571 \text{ บิต}$$

แอททริบิวต์ที่เหลือที่สามารถถูกเลือกมาเป็นตัวทดสอบเพื่อจัดกลุ่มของข้อมูลฝึกคือแอททริบิวต์ humidity และ windy สามารถคำนวณค่าเกนจากการเลือกแต่ละแอททริบิวต์ได้ดังนี้

$$\text{Gain(humidity)} = I(\text{outlook} = \text{sunny}) - E_{\text{humidity}}(\text{outlook} = \text{sunny})$$
$$= 0.971 - 0$$
$$= 0.971 \text{ บิต}$$

$$\text{Gain(windy)} = I(\text{outlook} = \text{sunny}) - E_{\text{windy}}(\text{outlook} = \text{sunny})$$
$$= 0.971 - 0.951$$
$$= 0.020 \text{ บิต}$$

จะเห็นว่าแอททริบิวต์ที่ให้ค่าเกนสูงที่สุดคือ humidity ดังนั้นแอททริบิวต์นี้จึงถูกเลือกเป็นโหนดระดับที่ 2 ต่อจาก outlook = sunny และยังคงเหลือโหนดลูกทางขวาของโหนด outlook (outlook = rainy) ที่ต้องพิจารณาเลือกแอททริบิวต์และจากวิธีการคำนวณค่าเกนที่แสดงด้วยตัวอย่างก่อนหน้านี้ สามารถเลือกได้ว่าแอททริบิวต์ windy จะให้ค่าเกนสูงที่สุด จึงถูกเลือกเป็นโหนดระดับที่ 2 ต่อจาก outlook = rainy กระบวนการสร้างต้นไม้ตัดสินใจจะสิ้นสุดเมื่อโหนดใบเป็นกลุ่มของข้อมูลเดียวกันทั้งหมด และจะได้โครงสร้างของต้นไม้ตัดสินใจเป็นดังรูปที่ 2.1

2.2.2 ค่ามาตรฐานอัตราส่วนเกน (Gain ratio criterion)

ในอัลกอริทึม ID3 จะใช้ค่ามาตรฐานเกนเป็นหลักในการเลือกแอททริบิวต์ที่จะใช้เป็นโหนดรากของต้นไม้ตัดสินใจหรือของต้นไม้ย่อย แต่ในอัลกอริทึม C4.5 ได้เพิ่มการใช้ค่ามาตรฐานอัตราส่วนเกนในการตัดสินใจเลือกแอททริบิวต์ที่จะใช้เป็นโหนดรากเข้ามาด้วย เนื่องจากค่ามาตรฐานเกนจะมีความลำเอียงอย่างมาก กับข้อมูลที่ประกอบด้วยแอททริบิวต์ที่มีค่าที่เป็นไปได้จำนวนมาก ๆ เช่น ชุดข้อมูลที่ประกอบด้วยแอททริบิวต์หมายเลขประจำตัวซึ่งมีค่าไม่ซ้ำกันในแต่ละตัวอย่าง ถ้าแบ่งข้อมูลตามแอททริบิวต์นี้จะทำให้ได้จำนวนตัวอย่างเพียง 1 ตัวอย่างต่อ 1 กิ่งของต้นไม้ และเมื่อคำนวณค่า entropy จากการแบ่งตัวอย่างบนแอททริบิวต์นี้จะได้เท่ากับ 0 ทำให้ค่าเกนที่ได้ของแอททริบิวต์นี้มีค่าสูงที่สุด (ก้องศักดิ์ จงเกษมวงศ์, 2543)

จากตัวอย่างข้อมูลการตัดสินใจเล่นกอล์ฟในตารางที่ 2.1 ถ้าใช้แอททริบิวต์ ID ในการจัดกลุ่มข้อมูลจะต้องการปริมาณข้อมูลประกอบการตัดสินใจเพื่อจำแนกกลุ่มดังนี้

$$E(ID) = (1/14) \times (-(0/1) \times \log_2(0/1) - (1/1) \times \log_2(1/1)) + \ldots$$
$$+ (1/14) \times (-(0/1) \times \log_2(0/1) - (1/1) \times \log_2(1/1))$$
$$= 0 \text{ บิต}$$

เมื่อแบ่งตัวอย่างบนแอททริบิวต์นี้จะได้ค่า entropy เท่ากับ 0 ดังนั้นค่ามาตรฐานเกนของแอททริบิวต์นี้จะเท่ากับปริมาณข้อมูลที่ต้องการจะระบุว่าข้อมูลหนึ่งเรคคอร์ดอยู่ในกลุ่ม P หรือ N ที่โหนดรากซึ่งมีค่าเท่ากับ 0.940 บิต ทำให้ค่ามาตรฐานเกนนี้มีค่าสูงกว่าแอททริบิวต์อื่น ๆ ดังนั้นแอททริบิวต์ ID นี้จะถูกเลือกมาเป็นตัวทดสอบเพื่อจัดกลุ่มของข้อมูลฝึก

ดังนั้นจะเห็นว่า การวัดค่ามาตรฐานเกนจะได้ค่าสูงเมื่อแอททริบิวต์นั้นมีค่าที่เป็นไปได้จำนวนมาก ๆ ซึ่งไม่สามารถนำมาใช้เป็นโหนดของต้นไม้ เพื่อทำนายกลุ่มของข้อมูลใหม่ที่ไม่เคยเห็นได้อย่างถูกต้อง จึงต้องแก้ไขความลำเอียงนี้โดยการปรับค่าเกนให้ถูกต้อง โดยใช้ค่าสารสนเทศการแบ่งแยก (split information) ของแต่ละแอททริบิวต์เพื่อใช้คำนวณค่ามาตรฐานอัตราส่วนเกน (Witten and Frank, 2005)

ถ้ากำหนดให้ T แทนชุดของข้อมูลฝึก เมื่อแบ่งตัวอย่างโดยใช้แอททริบิวต์ A จะได้ชุดของตัวอย่างย่อยในแต่ละกิ่งเป็น $\{t_1, t_2, \ldots, t_v\}$ จำนวน v ชุด ตามค่าที่เป็นไปได้ของแอททริบิวต์ A และสามารถคำนวณค่าสารสนเทศการแบ่งแยกได้ดังนี้

$$\text{ค่าสารสนเทศการแบ่งแยก} = -\sum_{i=1}^{v} \frac{|t_i|}{|T|} \log_2\left(\frac{|t_i|}{|T|}\right) \qquad (2\text{-}5)$$

ค่าสารสนเทศการแบ่งแยกนี้จะแสดงถึงระดับการกระจายของข้อมูล เมื่อแบ่งข้อมูลตัวอย่าง T เป็น v ชุดย่อยตามค่าที่เป็นไปได้ของแอททริบิวต์ A โดยค่านี้จะมีค่าสูงสุดเมื่อ $|t_i|$ เป็น 1 เท่ากันในทุกกิ่ง และจะลดลงเมื่อค่า $|t_i|$ เพิ่มขึ้น เมื่อนำค่านี้ไปหารค่ามาตรฐานเกนจะได้ค่ามาตรฐานอัตราส่วนเกน ซึ่งช่วยแก้ไขความลำเอียงที่เกิดขึ้นของค่ามาตรฐานเกนได้ โดยทำให้ค่ามาตรฐานอัตราส่วนเกนของแอททริบิวต์ที่มีค่าที่เป็นไปได้จำนวนมากถูกปรับลดลง (ก้องศักดิ์ จงเกษมวงศ์, 2543)

ค่ามาตรฐานอัตราส่วนเกน = ค่ามาตรฐานเกน / ค่าสารสนเทศการแบ่งแยก

จากตัวอย่างข้อมูลการตัดสินใจเล่นกอล์ฟในตารางที่ 2.1 สามารถคำนวณค่า gain ratio ของแอททริบิวต์ outlook ได้ดังนี้

$$\text{ค่าสารสนเทศการแบ่งแยก (outlook)} = -(5/14) \times \log_2(5/14) - (4/14) \times \log_2(4/14) - (5/14) \times \log_2(5/14)$$
$$= 1.577 \text{ บิต}$$

$$\text{Gain ratio (outlook)} = 0.247 / 1.577 = 0.156$$

และเมื่อแบ่งข้อมูลตัวอย่างด้วยแอททริบิวต์ temperature, humidity และ windy สามารถคำนวณค่า gain ratio ได้ดังนี้

$$\text{Gain ratio (temperature)} = 0.029 / 1.362 = 0.021$$

$$\text{Gain ratio (humidity)} = 0.152 / 1.000 = 0.152$$

$$\text{Gain ratio (windy)} = 0.048 / 0.985 = 0.049$$

จะเห็นว่าแอททริบิวต์ที่ให้ค่า gain ratio สูงที่สุดคือ outlook เช่นเดียวกับการคำนวณค่า information gain ดังนั้นแอททริบิวต์ outlook จึงถูกเลือกเป็นโหนดรากของต้นไม้ตัดสินใจ และจะสร้างต้นไม้ตัดสินใจต่อไปจนกระทั่งสามารถจัดกลุ่มของข้อมูลให้เป็นกลุ่มเดียวกันได้ทั้งหมด

2.3 การตัดกิ่งต้นไม้ตัดสินใจ

ในขณะที่กำลังสร้างต้นไม้ตัดสินใจ ในแต่ละกิ่งอาจเกิดการสร้างอย่างผิดปกติ อันเนื่องมาจากข้อมูลฝึกที่มีข้อมูลรบกวน (noise) ซึ่งเกิดจากการบันทึกข้อมูลผิดพลาดหรือความผิดพลาดที่เกิดจากระบบเอง หรือในชุดข้อมูลอาจจะมีข้อมูลที่ผิดปกติจากข้อมูลส่วนใหญ่ (outlier) ปะปนมาด้วย การตัดกิ่งต้นไม้ตัดสินใจเป็นเทคนิคที่ใช้สำหรับแก้ปัญหานี้ และจะช่วยลดการเกิดปัญหาการเจาะจงโมเดลกับข้อมูลมากเกินไป (overfitting) ได้ (Cohen and Jensen, 1997) โดยปัญหานี้ทำให้ได้โครงสร้างต้นไม้ที่สามารถจำแนกข้อมูลได้ดีกับชุดข้อมูลที่ใช้สร้างต้นไม้ตัดสินใจเท่านั้น

แต่เมื่อนำไปใช้กับข้อมูลใหม่ประสิทธิภาพในการจำแนกกลุ่มข้อมูลจะลดลง การตัดกิ่งต้นไม้ตัดสินใจจะใช้ค่าทางสถิติในการตัดกิ่งที่มีความน่าเชื่อถือน้อยที่สุดออกไป เพื่อทำให้ต้นไม้ใหม่ที่ได้สามารถทำงานได้รวดเร็วขึ้น และยังเป็นการปรับปรุงขีดความสามารถของต้นไม้ในการทำนายข้อมูลใหม่ ๆ ได้แม่นยำมากยิ่งขึ้นอีกด้วย โดยการตัดกิ่งต้นไม้ตัดสินใจที่เป็นที่นิยมมีอยู่ 2 ประเภท ดังต่อไปนี้

1) การตัดกิ่งขณะที่เรียนรู้ (pre-pruning)

เป็นการตัดกิ่งต้นไม้หรือหยุดการแตกกิ่งในขั้นตอนการสร้างต้นไม้ตัดสินใจ โดยการทำให้โหนดที่ถูกตัดนั้นเปลี่ยนเป็นใบ และให้ใบนั้นแสดงกลุ่มที่มีจำนวนข้อมูลสนับสนุนหรือมีความน่าจะเป็นที่ข้อมูลจะอยู่ในกลุ่มนั้นมากที่สุด (Breslow and Aha, 1997)

ในขณะที่ทำการสร้างต้นไม้ตัดสินใจนั้น จะต้องมีการคำนวณหรือวัดค่าทางสถิติที่สำคัญต่าง ๆ เช่น χ^2, information gain เพื่อใช้ประเมินว่าควรที่จะสร้างหรือแตกกิ่งของต้นไม้อย่างไร ถ้าค่าที่วัดได้ไม่ถึงจุดที่กำหนดไว้ก็จะถือว่าโหนดนั้นไม่สมควรที่จะทำการแตกกิ่งต่อไป ซึ่งเป็นการยากที่จะกำหนดว่าค่าที่ใช้เป็นเกณฑ์เหล่านั้นควรจะมีค่าเป็นเท่าไร ถ้ากำหนดค่านั้นสูงเกินไปก็จะทำให้ได้ต้นไม้ที่มีความซับซ้อน แต่ถ้ากำหนดค่าต่ำเกินไปก็จะทำให้ได้ต้นไม้ที่มีขนาดเล็กจนไม่สามารถนำไปใช้งานได้

2) การตัดกิ่งหลังการเรียนรู้ (post-pruning)

เป็นการตัดกิ่งของต้นไม้ตัดสินใจที่ถูกสร้างขึ้นสมบูรณ์แล้ว โดยใช้การวัดค่าความซับซ้อนของแต่ละโหนด หลังจากที่ทำการตัดกิ่งของต้นไม้แล้วโหนดที่อยู่ล่างสุดที่ไม่ได้ถูกตัดจะถูกเปลี่ยนไปเป็นใบและจะแสดงกลุ่มที่มีจำนวนข้อมูลสนับสนุนมากที่สุด (Breslow and Aha, 1997)

สำหรับทุก ๆ โหนดที่ไม่ใช่ใบของต้นไม้ จะมีการคำนวณค่าอัตราความผิดพลาดที่คาดหวังไว้ ซึ่งเป็นค่าที่แสดงถึงความผิดพลาดที่จะเกิดขึ้นถ้าโหนดของต้นไม้ย่อยนั้นถูกตัดออกไป โดยที่ค่าความผิดพลาดของโหนดที่ไม่ถูกตัดจะถูกคำนวณโดยใช้ค่าผลรวมความผิดพลาดของแต่ละกิ่ง และให้ค่าน้ำหนักตามสัดส่วนของกิ่งนั้น ๆ ถ้าการตัดโหนดนั้นนำไปสู่การเกิดความผิดพลาดที่สูงขึ้น โหนดของต้นไม้ย่อยนั้นก็จะต้องยังคงไว้ แต่ถ้าการตัดนั้นทำให้ได้ค่าความผิดพลาดเป็นที่ยอมรับได้โหนดนั้นก็จะถูกตัดออกไป หลังจากทำการตัดกิ่งต้นไม้ตัดสินใจแล้วจะต้องทำการวัดค่าความแม่นตรง (accuracy) ของต้นไม้ที่ทำการตัดกิ่งแล้วด้วย โดยที่ต้นไม้ที่ให้ค่าความผิดพลาดน้อยที่สุดจะถูกเลือก

นอกจากการตัดกิ่งต้นไม้ตัดสินใจโดยอาศัยการวัดค่าความผิดพลาดที่จะเกิดขึ้นแล้ว ยังมีเทคนิคอื่น ๆ เช่น การใช้ค่าการเข้ารหัส (encode) ในการพิจารณาตัดกิ่งของต้นไม้โดยใช้หลักการ

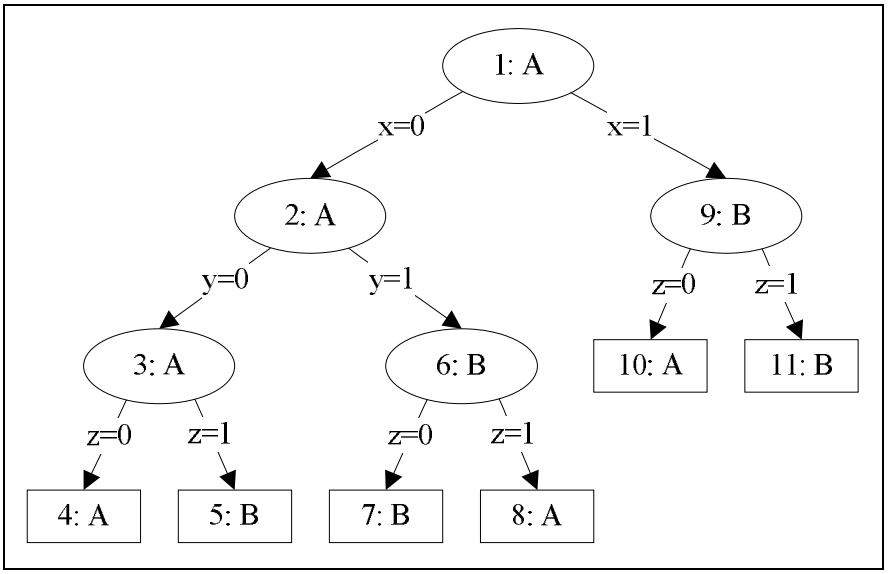
ของ Minimum Description Length (MDL) ด้วย (Quinlan and Rivest, 1989) เป็นต้น

วิธีการตัดกิ่งต้นไม้ตัดสินใจที่ศึกษาค้นคว้าในงานวิจัยนี้เป็นชนิด post-pruning ที่ขั้นตอนการตัดกิ่งเริ่มทำงานหลังจากต้นไม้ตัดสินใจได้สร้างขึ้นสมบูรณ์แล้ว โดยที่ Breiman, Friedman, Olshen, and Stone (1984) ได้ศึกษาพบว่าการตัดกิ่งประเภทนี้มีความเสถียรมากกว่าและให้ประสิทธิภาพสูงกว่าการตัดกิ่งขณะที่เรียนรู้ (pre-pruning) เพราะสามารถเลือกตัดโหนดที่ไม่เกิดประโยชน์จากต้นไม้ตัดสินใจที่สร้างขึ้นอย่างดีจากชุดข้อมูลแล้ว และใช้วิธีการต่าง ๆ ในการวัดค่าความผิดพลาดของโหนดเพื่อพิจารณาว่าจะตัดกิ่งของต้นไม้หรือไม่ วิธีการตัดกิ่งต้นไม้ตัดสินใจชนิด post-pruning ที่น่าสนใจสามารถแสดงขั้นตอนการทำงานได้ดังนี้

2.3.1 การตัดกิ่งแบบความผิดพลาดลดลง (Reduced-error pruning)

Reduced-errror pruning (REP) (Quinlan, 1987) วิธีนี้เป็นเทคนิคการตัดกิ่งต้นไม้ตัดสินใจที่มีแนวคิดง่ายที่สุด โดยใช้ชุดข้อมูลการตัดกิ่ง (pruning set) ที่แยกออกจากชุดข้อมูลฝึก (training set) เพื่อประเมินความถูกต้องของโหนดและใบของต้นไม้ที่ได้จากขั้นตอนการสร้างต้นไม้ตัดสินใจ ทำให้การประเมินค่าอัตราความผิดพลาดมีความลำเอียงน้อยลงเมื่อนำไปใช้กับตัวอย่างใหม่ที่ไม่เคยเห็น

ขั้นตอนการทำงานของวิธีการนี้ จะตรวจสอบโหนดจากล่างสุดขึ้นไปยังรากของต้นไม้ (bottom-up strategy) โดยใช้วิธีการเข้าถึงโหนดแบบ post-order traversal ทำการเปลี่ยนโหนดเป็นใบที่มีกลุ่มของข้อมูลเป็นกลุ่มหลักของกลุ่มตัวอย่าง จากการแบ่งกลุ่มของชุดข้อมูลฝึกที่โหนดนั้นของต้นไม้ นับจำนวนตัวอย่างที่ไม่ถูกต้องหรือไม่ใช่พวกเดียวกันกับตัวอย่างที่ใบนี้ เมื่อทดสอบด้วยชุดข้อมูลการตัดกิ่งเปรียบเทียบกับจำนวนตัวอย่างที่ไม่ถูกต้องของโหนดลูกของมัน ถ้าค่าความผิดพลาดในการจำแนกข้อมูลของโหนดที่เปลี่ยนเป็นใบ มีค่าน้อยกว่าหรือเท่ากับค่าความผิดพลาดในการจำแนกข้อมูลของโหนดลูกแล้วจะเลือกตัดโหนดนั้นออกไปแล้วเปลี่ยนเป็นใบ การตรวจสอบนี้จะทำซ้ำในแต่ละโหนด ถ้าไม่ทำให้ผลรวมทั้งหมดของความผิดพลาดในการจำแนกมีค่าเพิ่มขึ้น ผลที่ได้จากวิธีการนี้จะได้ต้นไม้ตัดสินใจที่มีขนาดเล็ก และให้ค่าความผิดพลาดต่ำที่สุดเมื่อทดสอบกับชุดข้อมูลการตัดกิ่ง

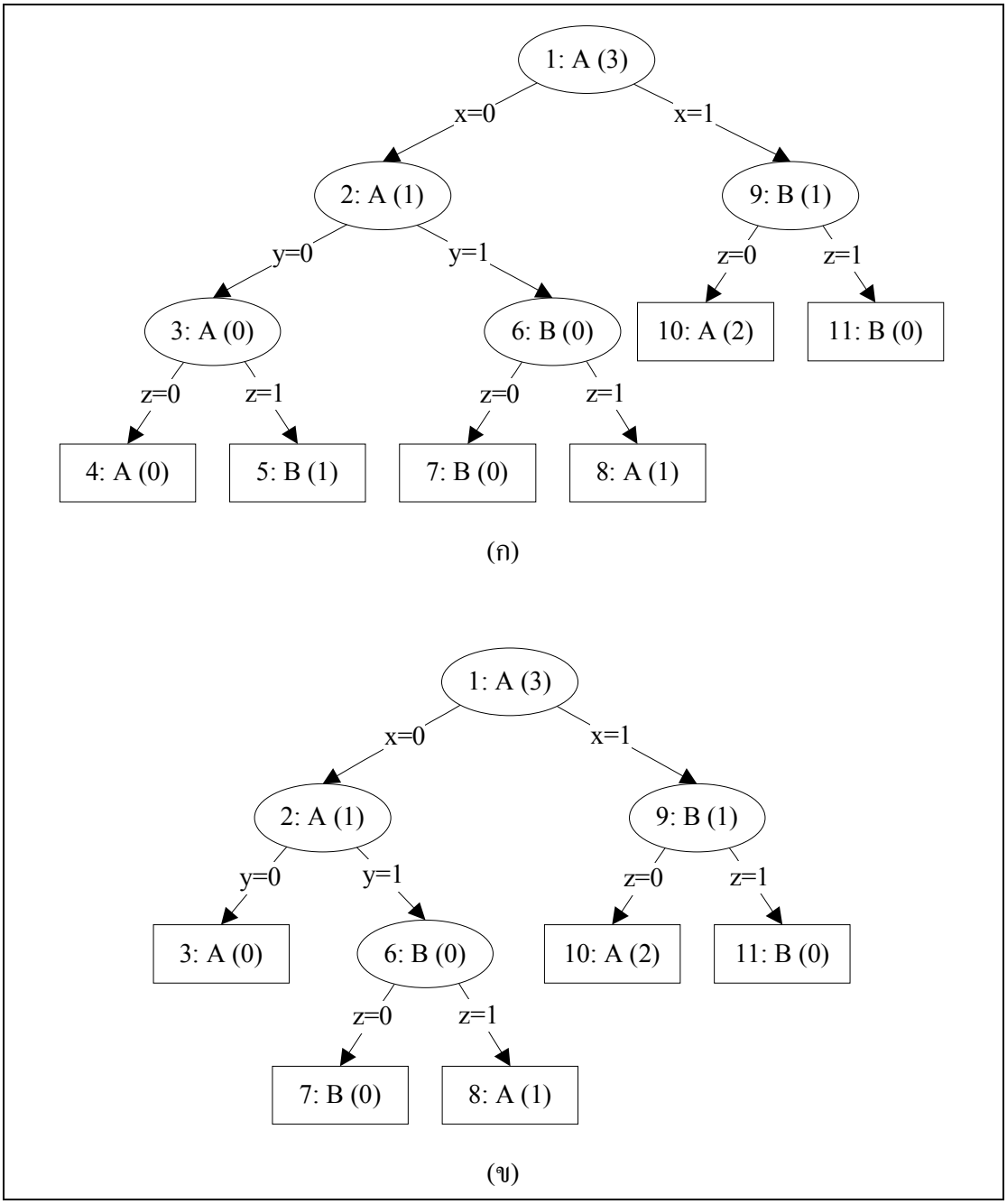


รูปที่ 2.4 ต้นไม้ตัดสินใจที่สร้างขึ้นโดยมีข้อมูล 2 กลุ่ม (A และ B)

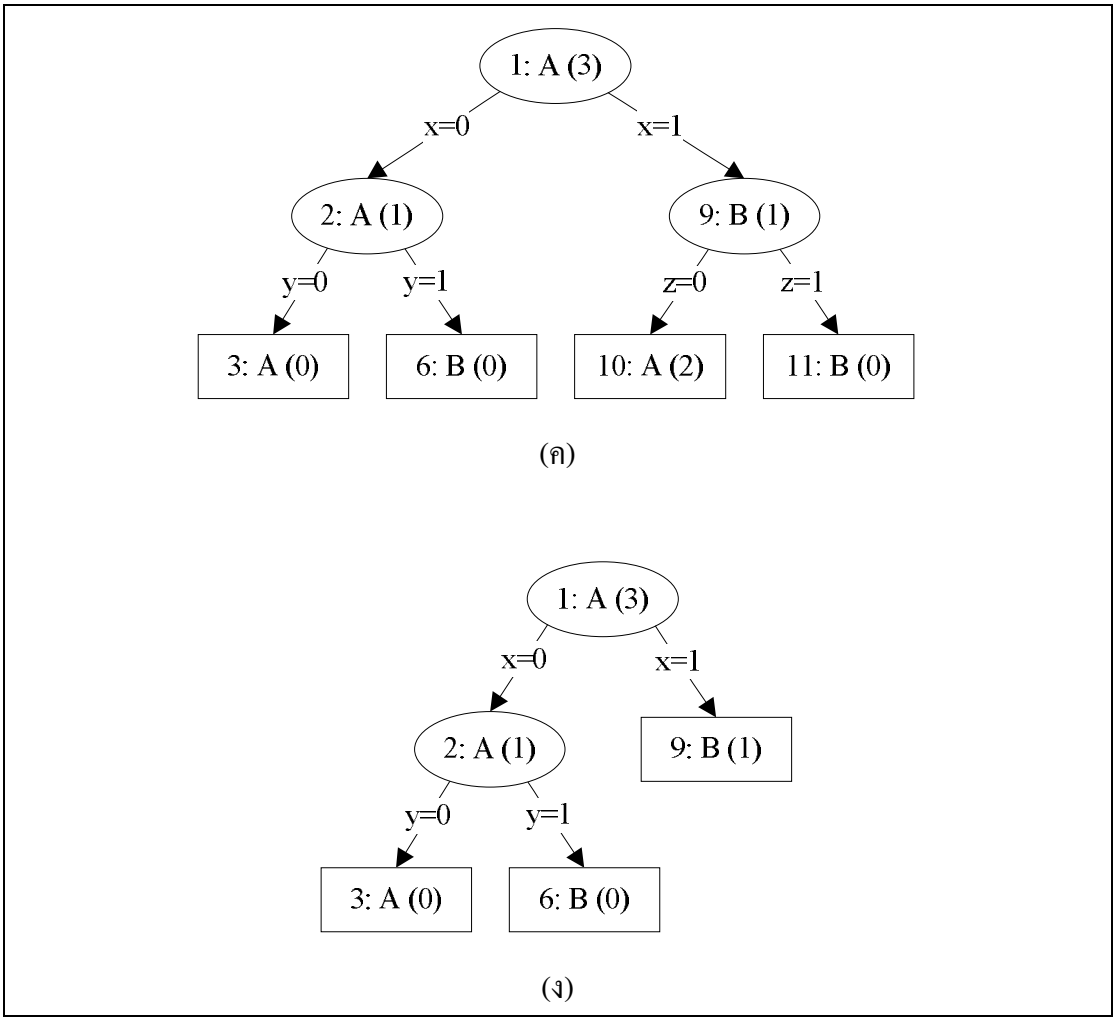
ตารางที่ 2.2 ตัวอย่างของชุดข้อมูลการตัดกิ่ง

x	y	z	class
0	0	1	A
0	1	1	B
1	1	0	B
1	0	0	B
1	1	1	A

ต้นไม้ตัดสินใจที่แสดงในรูปที่ 2.4 (Frank, 2000) ได้จากขั้นตอนการสร้างต้นไม้ซึ่งยังไม่ได้ตัดกิ่งใด ๆ ออกไป โดยแสดงกลุ่มหลักของข้อมูลจากการแบ่งกลุ่มของชุดข้อมูลฝึกที่แต่ละโหนดของต้นไม้และแสดงหมายเลขประจำแต่ละโหนดด้วย การตัดกิ่งต้นไม้ตัดสินใจด้วยวิธี REP ต้องใช้ชุดข้อมูลการตัดกิ่งประเมินความถูกต้องของต้นไม้ ซึ่งแสดงตัวอย่างชุดข้อมูลการตัดกิ่งสำหรับต้นไม้ตัดสินใจรูปที่ 2.4 ได้ดังตารางที่ 2.2



รูปที่ 2.5 แสดงตัวอย่างการตัดกิ่งต้นไม้ตัดสินใจด้วยวิธี Reduced-error pruning



รูปที่ 2.5 แสดงตัวอย่างการตัดกิ่งต้นไม้ตัดสินใจด้วยวิธี Reduced-error pruning (ต่อ)

ขั้นตอนการทำงานของการตัดกิ่งด้วยวิธี REP แสดงได้ดังรูปที่ 2.5 (Frank, 2000) โดยต้นไม้แต่ละต้นจะแสดงจำนวนตัวอย่างที่ไม่ถูกต้อง เมื่อทดสอบแต่ละโหนดด้วยชุดข้อมูลการตัดกิ่งจากตารางที่ 2.2 ไว้ในวงเล็บด้วย สามารถแสดงขั้นตอนการทำงานได้ดังนี้

1) พิจารณาที่โหนด 3 ของต้นไม้จากรูปที่ 2.5(ก) จะได้จำนวนตัวอย่างที่ไม่ถูกต้องเท่ากับ 0 ซึ่งมีค่าน้อยกว่าผลรวมของจำนวนตัวอย่างที่ไม่ถูกต้องที่ใบของมันซึ่งเท่ากับ 1 ดังนั้นจะตัดโหนดลูกของโหนด 3 ทิ้งแล้วเปลี่ยนโหนด 3 เป็นโหนดใบ แสดงได้ดังรูปที่ 2.5(ข)

2) พิจารณาที่โหนด 6 ของต้นไม้จากรูปที่ 2.5(ข) จะได้จำนวนตัวอย่างที่ไม่ถูกต้องเท่ากับ 0 ซึ่งมีค่าน้อยกว่าผลรวมของจำนวนตัวอย่างที่ไม่ถูกต้องที่ใบของมันซึ่งเท่ากับ 1 ดังนั้นจะตัดโหนดลูกของโหนด 6 ทิ้งแล้วเปลี่ยนโหนด 6 เป็นโหนดใบ แสดงได้ดังรูปที่ 2.5(ค)

3) พิจารณาที่โหนด 2 ของต้นไม้จากรูปที่ 2.5(ค) จะได้จำนวนตัวอย่างที่ไม่ถูกต้องเท่ากับ 1 ซึ่งมีค่ามากกว่าผลรวมของจำนวนตัวอย่างที่ไม่ถูกต้องที่ใบของมันซึ่งเท่ากับ 0 จะเห็นว่าความผิดพลาดที่โหนดเมื่อตัดกิ่งแล้วมีค่ามากกว่า ดังนั้นจึงไม่ตัดกิ่งของโหนดนี้ แสดงได้ดังรูปที่ 2.5(ง)

4) พิจารณาที่โหนด 9 ของต้นไม้จากรูปที่ 2.5(ค) จะได้จำนวนตัวอย่างที่ไม่ถูกต้องเท่ากับ 1 ซึ่งมีค่าน้อยกว่าผลรวมของจำนวนตัวอย่างที่ไม่ถูกต้องที่ใบของมันซึ่งเท่ากับ 2 ดังนั้นจะตัดโหนดลูกของโหนด 9 ทิ้งแล้วเปลี่ยนโหนด 9 เป็นโหนดใบ แสดงได้ดังรูปที่ 2.5(ง)

5) พิจารณาที่โหนด 1 ของต้นไม้จากรูปที่ 2.5(ง) จะได้จำนวนตัวอย่างที่ไม่ถูกต้องเท่ากับ 3 ซึ่งมีค่ามากกว่าผลรวมของจำนวนตัวอย่างที่ไม่ถูกต้องจากโหนดลูกของโหนด 1 ซึ่งเท่ากับ 2 จะเห็นว่าความผิดพลาดที่โหนดเมื่อตัดกิ่งแล้วมีค่ามากกว่า ดังนั้นจึงไม่ตัดกิ่งของโหนดนี้ ดังนั้นต้นไม้ตัดสินใจที่ตัดกิ่งด้วยวิธี REP อย่างสมบูรณ์แล้วแสดงได้ดังรูปที่ 2.5(ง)

การตัดกิ่งด้วยวิธี REP จะเข้าถึงแต่ละโหนดเพียงครั้งเดียวเพื่อประเมินโอกาสที่จะตัดโหนดนั้นออกไป ทำให้ค่าความซับซ้อนเชิงคำนวณ (computational complexity) เป็นแบบเชิงเส้นตามจำนวนโหนดของต้นไม้ตัดสินใจ O(n) เมื่อ n เป็นจำนวนโหนดของต้นไม้ตัดสินใจ แต่วิธีนี้ก็มีข้อเสียคือจำเป็นต้องใช้ชุดข้อมูลการตัดกิ่งแยกจากชุดข้อมูลฝึกซึ่งอาจมีปัญหากับบางชุดข้อมูลที่มีกลุ่มตัวอย่างขนาดเล็ก และอาจจะไม่สามารถจำแนกข้อมูลชนิดพิเศษที่อยู่นอกเหนือจากกลุ่มตัวอย่างส่วนใหญ่ได้อย่างถูกต้อง ถ้าข้อมูลนั้นไม่มีอยู่ในชุดข้อมูลการตัดกิ่ง เนื่องจากส่วนนั้นของต้นไม้ตัดสินใจจะถูกตัดออกไปด้วย (Esposito, Malerba, Semeraro, and Tamma, 1999)

2.3.2 การตัดกิ่งแบบความผิดพลาดในแง่ร้าย (Pessimistic error pruning)

Pessimistic error pruning (PEP) (Quinlan, 1987) การตัดกิ่งด้วยวิธีนี้ไม่ต้องการชุดข้อมูลที่แยกต่างหากเพื่อตัดกิ่งต้นไม้ตัดสินใจ แต่ชุดข้อมูลฝึกจะถูกใช้ทั้งในขั้นตอนการสร้างและตัดกิ่งที่ไม่มีความจำเป็นออกไปด้วย

กำหนดให้ต้นไม้ตัดสินใจ T สร้างขึ้นจากชุดข้อมูลฝึกที่มีจำนวนตัวอย่างเป็น N ถ้ามีตัวอย่าง K ตัวอย่างที่ใบและมีตัวอย่างที่ไม่ถูกต้องหรือไม่ใช่พวกเดียวกันกับตัวอย่างที่ใบนี้เป็น J ดังนั้นค่าอัตราความผิดพลาดที่ใบนี้จะเท่ากับ J/K จากการทดสอบด้วยชุดข้อมูลฝึกค่าอัตราความผิดพลาดนี้จะไม่ใช่ค่าที่แท้จริงเมื่อนำไปทดสอบกับข้อมูลใหม่ที่ไม่เคยเห็น จึงต้องกำหนดค่าคงที่เพิ่มเข้าไปด้วย โดยให้ค่าอัตราความผิดพลาดเกิดจากการใช้ continuity correction ของการแจกแจงทวินาม (binomial distribution) โดยให้ค่า J สามารถแทนด้วย J + 1/2 (Quinlan, 1987)

ถ้าเราพิจารณาต้นไม้ย่อย T' ของต้นไม้ T ซึ่งมีจำนวนใบของต้นไม้ย่อยเป็น L(T') จะได้จำนวนตัวอย่างที่ไม่ถูกต้องทั้งหมดเท่ากับ ΣJ เมื่อแทนค่า J ด้วย J + 1/2 จะได้จำนวนตัวอย่างที่

ไม่ถูกต้องเมื่อทดสอบกับข้อมูลใหม่ที่ไม่เคยเห็นเท่ากับ ΣJ + L(T')/2 วิธีนี้จะตัดกิ่งต้นไม้ย่อยออกไป แล้วเปลี่ยนเป็นโหนดใบที่มีกลุ่มของข้อมูลเป็นกลุ่มหลักของกลุ่มตัวอย่างจากการแบ่งกลุ่มของชุดข้อมูลฝึกที่โหนดนั้นของต้นไม้ ถ้าจำนวนตัวอย่างที่ไม่ถูกต้องของโหนดใบนี้ มีค่าน้อยกว่าหรือเท่ากับจำนวนตัวอย่างที่ไม่ถูกต้องของต้นไม้ย่อยบวกด้วยค่าความผิดพลาดมาตรฐาน (standard error) ของมัน (Quinlan, 1987)

```
TSH < 6.05:
| T4U measured = t: negative (1918)
| T4U measured = f:
| | age > 43.5: negative (58)
| | age < 43.5:
| | | query hypothyroid = f: negative (41)
| | | query hypothyroid = t: secondary hypothyroid (1)
TSH > 6.05:
| FTI < 64.5:
| | thyroid surgery = f:
| | | T3 < 2.3: primary hypothyroid (51)
| | | T3 > 2.3:
| | | | sex = M: negative (1)
| | | | sex = F: primary hypothyroid (4)
| | thyroid surgery = t:
| | | referral source = SVI: primary hypothyroid (1)
| | | referral source = <other>: negative (2)
| FTI > 64.5:
| | on thyroxine = t: negative (32)
| | on thyroxine = f:
| | | thyroid surgery = t: negative (3)
| | | thyroid surgery = f:
| | | | TT4 < 150.5: compensated hypothyroid (120)
| | | | TT4 > 150.5: negative (6)
```

รูปที่ 2.6 ต้นไม้ตัดสินใจก่อนการตัดกิ่งเพื่อใช้วินิจฉัยสภาวะ hypothyroid

จากรูปที่ 2.6 เป็นต้นไม้ตัดสินใจที่สร้างขึ้นเพื่อใช้วินิจฉัยสภาวะ hypothyroid ของร่างกายโดยสามารถจำแนกกลุ่มของข้อมูลได้เป็น 4 กลุ่ม {primary hypothyroid, secondary hypothyroid, compensated hypothyroid, negative} ต้นไม้ตัดสินใจนี้สร้างขึ้นจากชุดข้อมูลฝึกจำนวน 2,514 ตัวอย่าง โดยตัวเลขที่อยู่ด้านหลังโหนดใบ จะแสดงจำนวนตัวอย่างที่สามารถแบ่งกลุ่มตัวอย่างได้จากใบนั้น ถ้าพิจารณาที่ต้นไม้ย่อยดังนี้

T4U measured = t: negative (1918)
T4U measured = f:
| age > 43.5: negative (58)
| age < 43.5:
| | query hypothyroid = f: negative (41)
| | query hypothyroid = t: secondary hypothyroid (1)

ต้นไม้ย่อย T' นี้มีจำนวนตัวอย่างทั้งหมด ΣK เท่ากับ 2,018 ตัวอย่าง มีจำนวนใบ L(T') เป็น 4 และมีจำนวนตัวอย่างที่จำแนกไม่ถูกต้อง ΣJ เท่ากับ 0 เมื่อทดสอบต้นไม้ย่อยนี้ด้วยข้อมูลใหม่ที่ไม่เคยเห็น จะได้จำนวนตัวอย่างที่จำแนกข้อมูลไม่ถูกต้อง e(T') เป็นดังนี้

$$e(T') = \Sigma J + L(T')/2$$
$$= 0 + 4/2 = 2$$

คำนวณค่าความผิดพลาดมาตรฐาน (standard error, SE) ได้ดังนี้

$$SE(e(T')) = \sqrt{\frac{e(T') \times (\Sigma(- e(T'))}{\Sigma K}}$$

$$= \sqrt{\frac{2 \times (2018 - 2)}{2018}} = 1.41$$

ถ้าเปลี่ยนต้นไม้ย่อยนี้เป็นใบ ที่มีกลุ่มของข้อมูลเป็นกลุ่มหลักของกลุ่มตัวอย่างจากการแบ่งกลุ่มของชุดข้อมูลฝึกที่โหนดนั้นของต้นไม้ จะทำให้มีจำนวนตัวอย่างที่จำแนกกลุ่มของข้อมูลผิดพลาดเท่ากับ 1 แต่เนื่องจาก 1 + 1/2 < 2 + 1.41 ดังนั้นต้นไม้ย่อยนี้จะถูกเปลี่ยนเป็นใบ การ

ตรวจสอบนี้จะทำซ้ำในทุก ๆ ต้นไม้ย่อย โดยต้นไม้ตัดสินใจที่ผ่านการตัดกิ่งด้วยวิธี PEP เป็นที่เรียบร้อยแล้วแสดงได้ดังรูปที่ 2.7

```
TSH < 6.05: negative (2018)
TSH > 6.05:
| FTI < 64.5: primary hypothyroid (62)
| FTI > 64.5:
| | on thyroxine = t: negative (32)
| | on thyroxine = f:
| | | thyroid surgery = t: negative (3)
| | | thyroid surgery = f:
| | | | TT4 < 150.5: compensated hypothyroid (120)
| | | | TT4 > 150.5: negative (6)
```

รูปที่ 2.7 ต้นไม้ตัดสินใจหลังการตัดกิ่งเพื่อใช้วินิจฉัยสภาวะ hypothyroid

การใช้ชุดข้อมูลฝึกเพียงชุดข้อมูลเดียวสำหรับสร้างและตัดกิ่งต้นไม้ตัดสินใจ เป็นข้อดีสำหรับวิธีนี้ และสามารถทำงานได้อย่างรวดเร็วเนื่องจากการเข้าถึงโหนดจะทำงานจากบนลงล่าง ทำการตรวจสอบเพียงครั้งเดียวเริ่มจากรากไปยังใบของต้นไม้ โดยเมื่อต้นไม้ย่อยถูกตัดออกไปแล้วก็ไม่จำเป็นต้องตรวจสอบต้นไม้ย่อยที่อยู่ด้านล่างอีก แต่การนำค่าคงที่ 1/2 มาใช้และประมาณค่าความผิดพลาดที่มีการแจกแจงทวินามให้ใกล้เคียงกับการแจกแจงปกติ ยังไม่สามารถใช้ได้อย่างถูกต้องกับกลุ่มตัวอย่างทุกชุดข้อมูล (Esposito, Malerba, and Semeraro, 1997)

2.3.3 การตัดกิ่งโดยใช้ค่าความผิดพลาด (Error-based pruning)

Error-based pruning (EBP) วิธีการตัดกิ่งต้นไม้ตัดสินใจนี้ ได้นำมาใช้ในอัลกอริทึมสำหรับสร้างต้นไม้ตัดสินใจที่ชื่อ C4.5 (Quinlan, 1993) โดยพัฒนารูปแบบมาจากวิธี Pessimistic error pruning ปรับปรุงการคำนวณค่าอัตราความผิดพลาดที่คาดว่าจะเกิดขึ้น วิธีนี้ใช้ชุดข้อมูลฝึกสำหรับสร้างและตัดกิ่งต้นไม้ตัดสินใจ โดยไม่ต้องใช้ชุดข้อมูลที่แยกออกต่างหากสำหรับการตัดกิ่งโดยเฉพาะ

ขั้นตอนการทำงานจะตรวจสอบโหนดจากล่างสุดขึ้นไปยังรากของต้นไม้ โดยใช้วิธีการเข้าถึงโหนดแบบ post-order traversal สำหรับแต่ละโหนด t มีทางเลือก 2 วิธีในการปรับปรุงต้นไม้ตัดสินใจคือ ตัดกิ่งต้นไม้ T ตรงตำแหน่งของโหนด t แล้วเปลี่ยนเป็นโหนดใบ ที่มีกลุ่มของ

ข้อมูลเป็นกลุ่มหลักของกลุ่มตัวอย่างจากการแบ่งกลุ่มของชุดข้อมูลฝึกที่โหนดนั้น หรือตัดกิ่ง $T_{t'}$ ที่เป็นต้นไม้ย่อยของ T_t ที่มีผลรวมของจำนวนตัวอย่างมากที่สุดขึ้นมาแทนที่ตรงตำแหน่งของโหนด t โดยที่ไม่ทำให้ค่าความผิดพลาดเมื่อปรับปรุงต้นไม้ตัดสินใจแล้วมีค่าเพิ่มขึ้น (Esposito, Malerba, Semeraro, and Tamma, 1999)

การประเมินความถูกต้องของต้นไม้ตัดสินใจ จะใช้เพียงชุดข้อมูลฝึกเท่านั้นไม่ได้ จะต้องเป็นค่าประมาณจากประชากรทั้งหมด เพื่อใช้แทนค่าความผิดพลาดที่คาดว่าจะเกิดขึ้นเมื่อใช้ทดสอบกับข้อมูลใหม่ที่ไม่เคยเห็น จึงใช้ค่าจำกัดบนของการแจกแจงทวินาม (binomial distribution) ที่ระดับความเชื่อมั่นเท่ากับ CF (confidence factor) เป็นตัวแทนความผิดพลาดของประชากรแทนด้วย $U_{CF}(E,N)$ โดย E แทนจำนวนตัวอย่างที่แบ่งกลุ่มไม่ถูกต้องจาก N ตัวอย่าง (ก้องศักดิ์ จงเกษมวงศ์, 2543)

จากตัวอย่างของต้นไม้ตัดสินใจก่อนการตัดกิ่งในรูปที่ 2.8 เป็นต้นไม้ตัดสินใจที่สร้างขึ้นเพื่อใช้จำแนกลักษณะของกลุ่มคนที่เลือกพรรคการเมืองในการเลือกตั้ง โดยจำแนกกลุ่มของข้อมูลออกเป็น 2 กลุ่ม {democrat, republican} ต้นไม้ตัดสินใจนี้สร้างขึ้นจากชุดข้อมูลฝึกจำนวน 300 ตัวอย่าง โดยตัวเลขที่อยู่ด้านหลังโหนดใบ จะแสดงจำนวนตัวอย่างที่สามารถแบ่งกลุ่มตัวอย่างได้จากใบนั้น ต้นไม้นี้สร้างจากอัลกอริทึม C4.5 โดยแสดงในรูปแบบข้อความ ในโหนดหนึ่งของต้นไม้ที่ประกอบด้วยกิ่งและใบดังนี้

education spending = n: democrat (6.0)

education spending = y: democrat (9.0)

education spending = u: republican (1.0)

```
physician fee freeze = n:
| adoption of the budget resolution = y: democrat (151.0)
| adoption of the budget resolution = u: democrat (1.0)
| adoption of the budget resolution = n:
| | education spending = n: democrat (6.0)
| | education spending = y: democrat (9.0)
| | education spending = u: republican (1.0)
physician fee freeze = y:
| synfuels corporation cutback = n: republican (97.0/3.0)
| synfuels corporation cutback = u: republican (4.0)
| synfuels corporation cutback = y:
| | duty free exports = y: democrat (2.0)
| | duty free exports = u: republican (1.0)
| | duty free exports = n:
| | | education spending = n: democrat (5.0/2.0)
| | | education spending = y: republican (13.0/2.0)
| | | education spending = u: democrat (1.0)
physician fee freeze = u:
| water project cost sharing = n: democrat (0.0)
| water project cost sharing = y: democrat (4.0)
| water project cost sharing = u:
| | mx missile = n: republican (0.0)
| | mx missile = y: democrat (3.0/1.0)
| | mx missile = u: republican (2.0)
```

รูปที่ 2.8 ต้นไม้ตัดสินใจก่อนการตัดกิ่ง

จะเห็นได้ว่าไม่มีตัวอย่างที่แบ่งกลุ่มผิดพลาดเมื่อสร้างจากชุดข้อมูลฝึก สำหรับใบแรกที่มีจำนวนตัวอย่างเท่ากับ 6 ตัวอย่าง (N=6) ถูกจัดอยู่ในกลุ่ม democrat โดยที่ไม่มีตัวอย่างผิดพลาด (E=0) เมื่อคำนวณค่าจำกัดบนของการแจกแจง $U_{25\%}(0,6)$ ได้เท่ากับ 0.206 (อัลกอริทึม C4.5 ใช้ค่าความเป็นอิสระ CF=25% เป็นค่าโดยปริยาย) ดังนั้นจำนวนตัวอย่างผิดพลาดที่คาดว่าจะเกิดขึ้นเมื่อใช้แบ่งกลุ่มข้อมูลใหม่ที่ไม่เคยเห็น 6 ตัวอย่าง จะเท่ากับ 6×0.206 สำหรับใบที่เหลือ

คำนวณค่า $U_{25\%}(0,9)$ ได้เท่ากับ 0.143 และ $U_{25\%}(0,1)$ ได้เท่ากับ 0.750 ตามลำดับ เมื่อคำนวณตัวอย่างที่คาดว่าจะทำนายผิดที่โหนดนี้จะได้

$$\text{จำนวนตัวอย่างที่คาดว่าจะทำนายผิดพลาด} = 6 \times 0.206 + 9 \times 0.143 + 1 \times 0.750$$
$$= 3.273 \text{ ตัวอย่าง}$$

ถ้าโหนดนี้ถูกเปลี่ยนเป็นใบที่มีกลุ่มเป็น democrat จะทำให้ใบนี้มีจำนวนตัวอย่างทั้งหมด 16 ตัวอย่าง และมีจำนวนตัวอย่างที่ไม่ใช่กลุ่มนี้อยู่ 1 ตัวอย่าง ดังนั้นสามารถคำนวณจำนวนตัวอย่างที่คาดว่าจะทำนายผิดพลาดได้ดังนี้

$$\text{จำนวนตัวอย่างที่คาดว่าจะทำนายผิดพลาด} = 16 \times U_{25\%}(1,16)$$
$$= 16 \times 0.157$$
$$= 2.512 \text{ ตัวอย่าง}$$

```
physician fee freeze = n: democrat (168.0/2.6)
physician fee freeze = y: republican (123.0/13.9)
physician fee freeze = u:
|  mx missile = n: democrat (3.0/1.1)
|  mx missile = y: democrat (4.0/2.2)
|  mx missile = u: republican (2.0/1.0)
```

รูปที่ 2.9 ต้นไม้ตัดสินใจหลังการตัดกิ่ง

จะเห็นได้ว่าเราสามารถจะเปลี่ยนโหนดนี้เป็นใบที่มีค่าเป็น democrat ได้ เนื่องจากจำนวนตัวอย่างที่ทำนายผิดพลาดหลังจากเปลี่ยนโหนดเป็นใบแล้ว มีค่าน้อยกว่าเมื่อโหนดนี้ยังคงอยู่ เมื่อตัดกิ่งด้วยวิธี EBP เป็นที่เรียบร้อยแล้วจะได้ต้นไม้ตัดสินใจดังรูปที่ 2.9

ในต้นไม้ตัดสินใจหลังการตัดกิ่งแล้ว ค่า N/E ที่อยู่ด้านหลังใบของต้นไม้ แทน N ด้วยจำนวนตัวอย่างฝึกทั้งหมดที่ตกอยู่ที่ใบนี้ และ E แทนจำนวนตัวอย่างที่คาดว่าจะทำนายผิดเมื่อทำนายข้อมูล N ตัวอย่างที่ไม่เคยเห็นบนต้นไม้นี้ (ก้องศักดิ์ จงเกษมวงศ์, 2543)

การตัดกิ่งด้วยวิธี EBP ใช้การคำนวณช่วงความเชื่อมั่น (confidence level) เพื่อลดความลำเอียงที่เกิดจากการใช้ชุดข้อมูลฝึกคำนวณค่าความผิดพลาดเพียงชุดข้อมูลเดียว โดยกำหนดให้รูปแบบการแจกแจงทวินามประมาณได้เป็นการแจกแจงปกติ (normal distribution) สำหรับชุดข้อมูลที่มีกลุ่มตัวอย่างขนาดใหญ่ ทำให้ผลของการตัดกิ่งต้นไม้ตัดสินใจวิธีนี้จะให้ค่าความถูกต้องที่น้อยลงเมื่อใช้กับชุดข้อมูลที่มีกลุ่มตัวอย่างจำนวนน้อยกว่า 100 ตัวอย่าง (Frank, 2000) การตัดกิ่งด้วยวิธี EBP มีขั้นตอนการทำงาน 2 วิธีในการปรับปรุงต้นไม้ตัดสินใจคือ ตัดกิ่งต้นไม้ T ตรงตำแหน่งของโหนด t แล้วเปลี่ยนเป็นโหนดใบ ที่มีกลุ่มของข้อมูลเป็นกลุ่มหลักของกลุ่มตัวอย่างจากการแบ่งกลุ่มของชุดข้อมูลฝึกที่โหนดนั้น หรือตัดกิ่ง $T_{t'}$ ที่เป็นต้นไม้ย่อยของ T_t ที่มีผลรวมของจำนวนตัวอย่างมากที่สุดขึ้นมาแทนที่ตรงตำแหน่งของโหนด t โดยที่ไม่ทำให้ค่าความผิดพลาดเมื่อปรับปรุงต้นไม้ตัดสินใจแล้วมีค่าเพิ่มขึ้น ดังนั้นค่าความซับซ้อนเชิงคำนวณของการตัดกิ่งด้วยวิธี EBP จึงมีค่าเท่ากับ $O(n(\log n)^2)$ เมื่อ n เป็นจำนวนโหนดของต้นไม้ตัดสินใจ (Witten and Frank, 2005)

2.3.4 การตัดกิ่งแบบค่าความซับซ้อน (Cost-complexity pruning)

Cost-complexity pruning (CCP) วิธีนี้เป็นเทคนิคการตัดกิ่งต้นไม้ตัดสินใจที่ใช้ในอัลกอริทึมสำหรับสร้างต้นไม้ตัดสินใจที่ชื่อ CART (Breiman, Friedman, Olshen, and Stone, 1984) โดยสามารถแบ่งการทำงานออกเป็น 2 ขั้นตอนคือ (Esposito, Malerba, and Semeraro, 1997)

1) การคัดเลือกเซตของต้นไม้ย่อย จากต้นไม้ที่ได้จากขั้นตอนการสร้างต้นไม้ตัดสินใจ T_{max} ได้เป็น $\{ T_0, T_1, T_2, \dots, T_L \}$
   โดยที่ $T_0 = T_{max}$ และ T_L คือรากของต้นไม้ตัดสินใจ
2) การเลือกต้นไม้ที่ดีที่สุด T_i จากเซตที่ได้ โดยใช้การประเมินความถูกต้องของต้นไม้ตัดสินใจ

ในขั้นตอนแรกต้นไม้ T_{i+1} ได้รับมาจาก T_i โดยการตัดกิ่งที่ทำให้ค่าอัตราความผิดพลาดในการจำแนก (resubstitution errors) เพิ่มขึ้นน้อยที่สุด โดยเมื่อต้นไม้ T ถูกตัดกิ่งที่โหนด t จะได้ค่าอัตราความผิดพลาดเพิ่มขึ้นเท่ากับ $R(t) - R(T_t)$ และทำให้จำนวนของใบลดลงเท่ากับ $L(T_t)$ ลบด้วย 1

$$\frac{R(t) - R(T_t)}{L(T_t) - 1} = \alpha_T$$

ค่าอัตราส่วนการเพิ่มขึ้นของอัตราความผิดพลาด ต่อจำนวนใบของต้นไม้ที่ถูกตัดกิ่งออกไปเป็นค่าความซับซ้อน (cost-complexity) ของต้นไม้ T' ดังนั้น T_{i+1} จะได้รับมาจาก T_i โดย

การตัดกิ่งต้นไม้ตัดสินใจที่ทำให้ค่าความซับซ้อนมีค่าต่ำที่สุด โดยถ้าต้นไม้มีค่าความซับซ้อนเท่ากัน จะเลือกต้นไม้ที่มีจำนวนโหนดน้อยกว่า (Quinlan, 1987)

จากต้นไม้ตัดสินใจในรูปที่ 2.6 ซึ่งสร้างขึ้นจากชุดข้อมูลฝึกจำนวน 2,514 ตัวอย่าง ถ้าพิจารณาที่ต้นไม้ย่อยดังนี้

T4U measured = t: negative (1918)
T4U measured = f:
| age > 43.5: negative (58)
| age < 43.5:
| | query hypothyroid = f: negative (41)
| | query hypothyroid = t: secondary hypothyroid (1)

จะเห็นได้ว่ากลุ่มหลักของตัวอย่างแสดงที่ใบของต้นไม้ย่อยนี้เป็น negative ถ้าเปลี่ยนต้นไม้ย่อยเป็นใบที่มีกลุ่มเป็น negative จะทำให้มีจำนวนตัวอย่างที่จำแนกกลุ่มของข้อมูลผิดพลาดเท่ากับ 1 ดังนั้นค่าอัตราความผิดพลาดในการจำแนกที่โหนดนี้ R(t) จะเท่ากับ 1/2514 ถ้าพิจารณาจากต้นไม้ย่อยนี้ที่มีจำนวนใบเท่ากับ 4 จะเห็นว่าไม่มีตัวอย่างที่จำแนกไม่ถูกต้องอยู่เลย ดังนั้นสามารถคำนวณค่า cost-complexity ได้ดังนี้

$$\alpha_T = \frac{(1/2514) - 0}{4 - 1} = 0.00013$$

ค่า cost-complexity ที่ได้นี้มีค่าต่ำที่สุด ดังนั้นต้นไม้ T_1 ซึ่งได้รับมาจากต้นไม้ที่ได้จากขั้นตอนการสร้าง T_0 โดยแทนที่ต้นไม้ย่อยนี้ด้วยใบจะถูกคัดเลือกไว้ในเซ็ต เพื่อนำไปประเมินความถูกต้อง เพื่อให้ได้ต้นไม้ตัดสินใจที่สามารถจำแนกข้อมูลใหม่ได้อย่างดีที่สุดต่อไป และต้นไม้ตัดสินใจที่ผ่านการตัดกิ่งด้วยวิธี CCP เป็นที่เรียบร้อยแล้วแสดงได้ดังรูปที่ 2.7 เช่นเดียวกัน

ในขั้นที่สองเป็นการเลือกต้นไม้ที่ดีที่สุด โดยใช้วิธีเปรียบเทียบความถูกต้องในการจำแนกเพื่อตรวจสอบค่าอัตราความผิดพลาดของต้นไม้แต่ละต้น โดยอาจใช้วิธี cross-validation หรือใช้ชุดข้อมูลการตัดกิ่งในการตรวจสอบ

การตัดกิ่งด้วยวิธี CCP มีข้อเสียเมื่อเปรียบเทียบกับวิธี REP สำหรับการใช้ชุดข้อมูลการตัดกิ่ง เนื่องจากเราสามารถเลือกตรวจสอบต้นไม้ได้จากภายในเซตเท่านั้น แทนที่จะสามารถตรวจสอบได้จากต้นไม้ย่อยที่เป็นไปได้ทั้งหมดของต้นไม้ที่สร้างขึ้นอย่างสมบูรณ์ ดังนั้นถ้าต้นไม้

ย่อยที่มีความถูกต้องมากที่สุดเมื่อทดสอบกับชุดข้อมูลการตัดกิ่งไม่อยู่ในเซตแล้ว วิธีนี้จะไม่สามารถเลือกต้นไม้นั้นได้ (Esposito et al., 1997) และการใช้วิธี cross-validation เพื่อประเมินความถูกต้องของต้นไม้ตัดสินใจจำเป็นต้องใช้เวลาในการทำงานเพิ่มมากขึ้นด้วย (Quinlan, 1987)

2.4 ทฤษฎีการตัดสินใจเชิงสถิติ

ในการศึกษาข้อมูลมีอยู่บ่อยครั้งที่เราจำเป็นต้องตัดสินใจเกี่ยวกับประชากรบนพื้นฐานของข้อเท็จจริงที่ได้จากกลุ่มตัวอย่าง การตัดสินใจลักษณะนี้เรียกว่า การตัดสินใจเชิงสถิติ เช่น การทดสอบข้อมูลโดยวิธี ก. ให้ประสิทธิภาพที่ดีกว่าวิธี ข. และ ค. หรือไม่? เราควรคงสมมติฐานหรือปฏิเสธสมมติฐานทางสถิติที่ระดับนัยสำคัญที่กำหนดไว้ เป็นต้น

2.4.1 สมมติฐานทางสถิติ

สมมติฐาน หมายถึง ข้อความที่ใช้คาดคะเนความสัมพันธ์ระหว่างตัวแปรตั้งแต่ 2 ตัวขึ้นไป เป็นเครื่องชี้แนวทางในการค้นหาหรือพิสูจน์ข้อเท็จจริงได้อย่างมีระบบ ถูกต้องตรงเป้าหมาย และทำให้สามารถกำหนดปัญหาการวิจัยได้แคบลง (ยุทธ ไกยวรรณ์, 2546) โดยทั่วไปสามารถจำแนกสมมติฐานออกได้เป็น 2 ประเภทคือ

1) สมมติฐานทางการวิจัย (research hypothesis) เป็นข้อความที่เขียนขึ้นเพื่อพยากรณ์คำตอบจากการวิจัย โดยอาศัยเหตุผล ประสบการณ์ ความรู้ ผลการวิจัยที่ผ่านมา โดยเขียนสมมติฐานการวิจัยในลักษณะความสัมพันธ์ของตัวแปรตั้งแต่สองตัวขึ้นไป เช่น นักศึกษาที่เรียนจากวิธีสอนต่างกันมีผลสัมฤทธิ์ทางการเรียนต่างกัน เป็นต้น
2) สมมติฐานทางสถิติ (statistical hypothesis) เป็นสมมติฐานที่เขียนอยู่ในรูปโครงสร้างทางคณิตศาสตร์ และใช้สัญลักษณ์ทางสถิติแทน เพื่อให้สามารถทดสอบได้ด้วยวิธีการทางสถิติ สมมติฐานทางสถิติแบ่งออกเป็น 2 ชนิดคือ
   (1) สมมติฐานว่าง (null hypothesis) เป็นสมมติฐานที่ไม่แสดงความแตกต่างหรือไม่แสดงความสัมพันธ์ระหว่างกลุ่มหรือผลต่างเท่ากับศูนย์ สัญลักษณ์ที่ใช้คือ H_0
   (2) สมมติฐานทางเลือก (alternative hypothesis) เป็นสมมติฐานที่แสดงความแตกต่างหรือความสัมพันธ์กันระหว่างกลุ่มตัวอย่าง และจะต้องอยู่คู่กับสมมติฐานว่างเสมอ สัญลักษณ์ที่ใช้คือ H_1

2.4.2 ความคลาดเคลื่อนในการตัดสินใจ

ผลของการทดสอบสมมติฐานไม่ว่าจะตัดสินใจปฏิเสธหรือยอมรับ H_0 ก็ตาม จะมีโอกาสที่จะเกิดความคลาดเคลื่อนได้ ความคลาดเคลื่อนดังกล่าวแบ่งออกได้เป็น 2 ประเภทคือ

1) ความคลาดเคลื่อนประเภทที่ 1 (Type I error) เป็นความคลาดเคลื่อนที่เกิดจากการปฏิเสธสมมติฐานว่าง ทั้ง ๆ ที่สมมติฐานว่างนั้นเป็นจริง สัญลักษณ์ที่ใช้คือ α
2) ความคลาดเคลื่อนประเภทที่ 2 (Type II error) เป็นความคลาดเคลื่อนที่เกิดจากการยอมรับสมมติฐานว่าง ทั้ง ๆ ที่สมมติฐานว่างนั้นไม่เป็นจริง สัญลักษณ์ที่ใช้คือ β

สามารถสรุปผลการตัดสินใจได้ดังนี้

ตารางที่ 2.3 ประเภทของความคลาดเคลื่อนของการทดสอบสมมติฐานทางสถิติ

สภาพความเป็นจริง	การตัดสินใจ	
	ยอมรับ H_0	ปฏิเสธ H_0
H_0 เป็นจริง	ไม่มีความคลาดเคลื่อน	ความคลาดเคลื่อนประเภทที่ 1
H_0 ไม่เป็นจริง	ความคลาดเคลื่อนประเภทที่ 2	ไม่มีความคลาดเคลื่อน

2.4.3 ระดับความมีนัยสำคัญ

ระดับความมีนัยสำคัญ (level of significance) เป็นการกำหนดขอบเขตของความคลาดเคลื่อนที่จะยอมให้เกิดขึ้นได้ในการทดสอบสมมติฐานครั้งนั้น ความคลาดเคลื่อนดังกล่าวเป็นความคลาดเคลื่อนประเภทที่ 1 การกำหนดความคลาดเคลื่อนนี้นิยมกำหนดเป็นค่าความน่าจะเป็น เช่น เมื่อทดสอบสมมติฐานโดยใช้ระดับความมีนัยสำคัญเป็น 0.05 เขียนแทนด้วยสัญลักษณ์ดังนี้ $\alpha = 0.05$

2.4.4 บริเวณวิกฤต

บริเวณวิกฤต (critical region) เป็นขอบเขตซึ่งกำหนดตามระดับความมีนัยสำคัญ ถ้าค่าสถิติที่คำนวณได้ตกอยู่ในขอบเขตนี้ ก็จะตัดสินใจปฏิเสธสมมติฐานว่าง (H_0) และจะถือว่าการทดสอบนั้นมีนัยสำคัญ นั่นคือ ความแตกต่างระหว่างคุณลักษณะของกลุ่มตัวอย่างประชากรกับประชากรมีมากเกินขอบเขตที่กำหนดไว้

2.4.5 ขั้นตอนการทดสอบสมมติฐานทางสถิติ

ในการทดสอบสมมติฐานโดยใช้สถิติ จะต้องกำหนดประชากรที่ต้องการศึกษา เลือกกลุ่มตัวอย่างเพื่อให้เป็นตัวแทนจากประชากรนั้น ๆ กำหนดสมมติฐานทางการวิจัย รวบรวมข้อมูล

จากกลุ่มตัวอย่าง นำข้อมูลมาวิเคราะห์และทดสอบสมมติฐาน โดยขั้นตอนในการทดสอบสมมติฐานเป็นดังนี้

1) กำหนดสมมติฐานว่าง (H_0) ซึ่งเป็นสมมติฐานที่ตรงกันข้ามกับสมมติฐานทางเลือก (H_1) สมมติฐานว่างนี้จะใช้สำหรับทดสอบด้วยวิธีการทางสถิติ
2) กำหนดระดับความมีนัยสำคัญ (α) ที่จะใช้เป็นเกณฑ์ในการตัดสินใจ
3) คำนวณค่าทางสถิติที่ใช้ในการทดสอบสมมติฐาน จากข้อมูลที่รวบรวมมาจากกลุ่มตัวอย่าง
4) พิจารณาว่าถ้าค่าความน่าจะเป็นของสถิติที่คำนวณได้ในขั้นที่ 3 มีค่าน้อยกว่าหรือเท่ากับระดับความมีนัยสำคัญ ก็จะตัดสินใจปฏิเสธ H_0 และยอมรับ H_1 แต่ถ้าค่าความน่าจะเป็นของสถิติที่คำนวณได้มากกว่าระดับความมีนัยสำคัญ ก็จะยอมรับ H_0 ณ ระดับความมีนัยสำคัญนั้น

2.4.6 การทดสอบความเป็นอิสระต่อกัน

การทดสอบความเป็นอิสระต่อกันของข้อมูล สามารถใช้ตารางของตัวเลขที่เรียกว่าตารางการณ์จร (contingency table) ถ้าตารางนี้ประกอบด้วย r แถว และ c คอลัมน์ จะเรียกว่า r × c table ผลรวมของความถี่ในแต่ละแถวหรือแต่ละคอลัมน์เรียกว่า marginal frequencies และผลรวมของความถี่ทั้งหมดในตารางการณ์จรเรียกว่า grand total โดยสามารถแสดงรูปแบบของตารางการณ์จรได้ดังตารางที่ 2.4

ตารางที่ 2.4 ตัวอย่างตารางการณ์จรของความถี่ที่ได้จากการทดลอง

	A_1	A_2	…	A_j	
B_1	o_{11}	o_{12}	…	o_{1j}	R_1
B_2	o_{21}	o_{22}	…	o_{2j}	R_2
:	:	:		:	:
B_i	o_{i1}	o_{i2}	…	o_{ij}	R_i
	C_1	C_2	…	C_j	N

กำหนดให้ H_0 : เหตุการณ์ A และเหตุการณ์ B เป็นอิสระต่อกัน โดยที่

o_{ij} เป็นจำนวนของความถี่ที่ได้จากการทดลอง

e_{ij} เป็นจำนวนของความถี่ที่คาดว่าจะได้ของเหตุการณ์ B_i และเหตุการณ์ A_j

เราสามารถคำนวณค่าของ e_{ij} ได้โดยการคูณความน่าจะเป็นแต่ละค่าด้วยผลรวมของความถี่ทั้งหมด N ซึ่งจะได้ว่า

$$e_{ij} = N \times P(A_j \cap B_i) = N \times \frac{C_j}{N} \times \frac{R_i}{N} = \frac{R_i C_j}{N}$$

ดังนั้นจะเห็นได้ว่า ถ้า H_0 เป็นจริง จะสามารถหาความถี่ที่คาดว่าจะได้คือ $e \approx RC/N$ เมื่อ R และ C เป็นผลรวมของความถี่ในแถวและคอลัมน์ที่ต้องการ และ N คือผลรวมของความถี่ทั้งหมด เราสามารถคำนวณค่าของ Chi-squared (χ^2) ได้ดังนี้

$$\chi^2 = \sum_{i=1}^{r} \sum_{j=1}^{c} \frac{(o_{ij} - e_{ij})^2}{e_{ij}}$$

2.5 งานวิจัยที่เกี่ยวข้อง

งานวิจัยที่เกี่ยวข้องกับการตัดกิ่งต้นไม้ตัดสินใจ เป็นงานหนึ่งที่ได้รับการศึกษาค้นคว้าและมีผู้ให้ความสนใจเป็นจำนวนมากในสาขาการเรียนรู้ของเครื่อง (machine learning) ทั้งจากการวิเคราะห์วิธีการตัดกิ่งที่มีผู้พัฒนาขึ้น (Esposito, Malerba, and Semeraro, 1997; Oates and Jensen, 1997, 1998, 1999) และการพัฒนาเทคนิคใหม่เพื่อปรับปรุงประสิทธิภาพของวิธีการตัดกิ่งที่มีอยู่ อีกทั้งยังมีการศึกษาเปรียบเทียบวิธีการตัดกิ่งต้นไม้ตัดสินใจอีกหลายวิธีด้วย (Esposito, et al., 1997; Esposito, Malerba, Semeraro, and Tamma, 1999; Frank, 2000; Mingers, 1989; Quinlan, 1987)

Quinlan (1987) เป็นนักวิจัยที่ทำการศึกษาเปรียบเทียบวิธีการตัดกิ่งต้นไม้ตัดสินใจ โดยทำการทดสอบข้อมูลกับวิธีการตัดกิ่ง 3 แบบคือ cost-complexity pruning, reduced-error pruning และ pessimistic error pruning เกณฑ์ที่ใช้เปรียบเทียบประสิทธิภาพของต้นไม้ตัดสินใจที่ได้คือ ค่าความแม่นตรงและขนาดของต้นไม้ตัดสินใจ โดยทดสอบข้อมูลกับชุดข้อมูลจำนวน 6 ชุดที่ได้จากการวิเคราะห์ต่าง ๆ และจากการสังเคราะห์ขึ้น จากผลการทดสอบข้อมูลสามารถสรุปได้ว่า เมื่อทดสอบทุกชุดข้อมูลกับการตัดกิ่งต้นไม้ตัดสินใจ ในทุกวิธีจะสร้างต้นไม้ตัดสินใจที่มีขนาดเล็กลงอย่างมีนัยสำคัญ โดยวิธี cost-complexity pruning มีแนวโน้มสร้างต้นไม้ตัดสินใจที่มีขนาดเล็กกว่าวิธี reduced-error pruning และ pessimistic error pruning เมื่อเปรียบเทียบค่าความแม่นตรงในการจำแนกกลุ่มของข้อมูล ถ้าต้นไม้ที่สร้างขึ้นมีความซับซ้อนน้อยกว่าและให้ค่าความแม่นตรงสูงกว่าหรือใกล้เคียงกับต้นไม้ที่ยังไม่ได้ตัดกิ่ง แสดงว่าการตัดกิ่งให้ผลที่ดีกว่า การตัดกิ่งโดยวิธี reduced-error pruning ให้ค่าความแม่นตรงสูงกว่าวิธี cost-complexity pruning เล็กน้อย เป็นผลมาจากขนาด

ของต้นไม้ตัดสินใจที่เล็กกว่า การตัดกิ่งวิธี cost-complexity pruning และ reduced-error pruning จำเป็นต้องใช้ชุดข้อมูลที่แยกต่างหากเพื่อใช้ตัดกิ่ง เป็นจุดที่เสียเปรียบวิธีที่ใช้เพียงชุดข้อมูลฝึกเพียงชุดข้อมูลเดียว

การศึกษาเปรียบเทียบวิธีการตัดกิ่งอีกงานหนึ่งได้รับการทดสอบโดย Mingers (1989) ซึ่งใช้วิธีการตัดกิ่ง 3 แบบเช่นเดียวกับที่ทดสอบโดย Quinlan (1987) โดยที่ Mingers ได้ทดสอบเพิ่มเติมกับวิธีการตัดกิ่ง 2 แบบรวมเข้ากับการเปรียบเทียบของเขาด้วยคือ critical value pruning และ minimum-error pruning แต่อย่างไรก็ตามจากผลงานการวิจัยของเขายังคงถูกวิพากษ์วิจารณ์เนื่องจากไม่ได้ใช้ข้อมูลชุดเดียวกันในการทดสอบกับวิธีการตัดกิ่งหลายแบบ อีกทั้งวิธีการตัดกิ่ง reduced-error pruning ที่เขาใช้ยังเป็นวิธีที่ไม่ได้มาตรฐานด้วย

Esposito, Malerba และ Semeraro (1997) ได้ศึกษาเปรียบเทียบกับอัลกอริทึมการตัดกิ่งหลายวิธีเช่นเดียวกับ Mingers (1989) รวมทั้งได้ทดสอบอัลกอริทึมที่พัฒนาขึ้นจากวิธี pessimistic error pruning ที่ชื่อว่า error-based pruning ด้วย การทดสอบของพวกเขาเป็นที่ยอมรับเนื่องจากได้เปรียบเทียบแต่ละวิธีโดยใช้ข้อมูลชุดเดียวกันในการสร้างต้นไม้ตัดสินใจ และงานวิจัยของพวกเขายังได้ใช้วิธี reduced-error pruning ตามแนวคิดของ Quinlan (1987) ในการทดสอบอีกด้วย พวกเขาทำการทดสอบกับชุดข้อมูลจำนวน 14 ชุดที่รวบรวมอยู่ใน UCI Machine Learning Repository (Blake and Merz, 1998) โดยแต่ละชุดข้อมูลจะแบ่งออกเป็น 3 ส่วนคือ growing set (49%), pruning set (21%) และ test set (30%) ผลรวมของข้อมูล growing set และ pruning set เรียกว่าชุดข้อมูลฝึกและทดสอบประสิทธิภาพของวิธีการตัดกิ่งต้นไม้ตัดสินใจต่าง ๆ โดยใช้ชุดข้อมูลทดสอบ จากผลการทดสอบข้อมูลสามารถสรุปได้ดังนี้

1) การใช้ชุดข้อมูลที่แยกไว้โดยเฉพาะเพื่อการตัดกิ่งต้นไม้ตัดสินใจ เมื่อเปรียบเทียบผลลัพธ์ที่ได้จากวิธีใช้ชุดข้อมูลฝึกเพียงชุดข้อมูลเดียวเพื่อใช้สำหรับการตัดกิ่ง จะให้ต้นไม้ตัดสินใจที่มีประสิทธิภาพไม่แตกต่างกัน
2) เทคนิคการตัดกิ่งต้นไม้ตัดสินใจ ไม่ได้ทำให้ความแม่นตรงในการจำแนกกลุ่มข้อมูลของต้นไม้ตัดสินใจมีค่าลดลง
3) กำหนดให้ค่าความผิดพลาดพื้นฐาน (base error) ซึ่งเป็นค่าร้อยละของความผิดพลาดเมื่อแบ่งกลุ่มที่มีจำนวนข้อมูลสนับสนุนมากที่สุด โดยชุดข้อมูลที่มีค่าความผิดพลาดพื้นฐานต่ำ การใช้เทคนิคการตัดกิ่งต้นไม้ตัดสินใจจะให้ประสิทธิภาพที่ดี
4) ผลที่ได้จากการตัดกิ่งด้วยวิธี pessimistic error pruning และ error-based pruning ให้ผลในแนวทางเดียวกัน ทั้งที่ทั้งสองวิธีมีวิธีการคำนวณที่แตกต่างกัน

Esposito, Malerba, Semeraro และ Tamma (1999) ได้ศึกษาเปรียบเทียบวิธีการตัดกิ่งต้นไม้ตัดสินใจรวม 6 วิธีเช่นเดียวกับการทดสอบข้อมูลครั้งก่อน (Esposito, Malerba, and Semeraro, 1997) โดยทดสอบกับชุดข้อมูลจำนวน 14 ชุด เพื่อวิเคราะห์ความแม่นตรงในการจำแนกกลุ่มของข้อมูลและขนาดของต้นไม้ตัดสินใจที่ได้รับการตัดกิ่งแล้ว โดยพวกเขาใช้วิธี 10-fold cross-validation แทนการใช้ข้อมูลทดสอบในการประเมินประสิทธิภาพของวิธีการตัดกิ่งต้นไม้ตัดสินใจต่าง ๆ ผลการทดสอบข้อมูลสามารถยืนยันข้อสรุปได้ว่า วิธีการตัดกิ่งต้นไม้ตัดสินใจไม่ได้ทำให้ความแม่นตรงในการจำแนกกลุ่มของข้อมูลใหม่ของต้นไม้ตัดสินใจมีค่าลดลงอย่างมีนัยสำคัญ และเมื่อชุดข้อมูลมีค่าอัตราความผิดพลาดที่เกิดขึ้นต่ำเมื่อตัวอย่างที่มีกลุ่มหลักถูกทำนาย (base error) วิธีการตัดกิ่งต้นไม้ตัดสินใจจะให้ผลที่ดี

Reduced-error pruning เป็นวิธีการตัดกิ่งต้นไม้ตัดสินใจที่พัฒนาขึ้นโดย Quinlan (1987) มีขั้นตอนการทำงานที่ไม่ซับซ้อนและใช้เวลาน้อยในการทำงาน แต่มีข้อเสียที่ต้องการชุดข้อมูลที่แยกไว้ต่างหากสำหรับการตัดกิ่ง ซึ่งอาจจะมีข้อมูลจำนวนน้อยเกินไปสำหรับชุดข้อมูลที่มีขนาดเล็ก แต่อย่างไรก็ตามในงานทำเหมืองข้อมูลจะประกอบด้วยข้อมูลจำนวนมาก การแบ่งชุดข้อมูลสำหรับการตัดกิ่งจึงไม่เป็นปัญหา (Esposito, et al., 1999)

Quinlan (1987) ได้อธิบายขั้นตอนการทำงานของวิธีการตัดกิ่ง reduced-error pruning แต่เนื่องจากยังมีความไม่ชัดเจนในรายละเอียด จึงทำให้เกิดการตีความหมายที่แตกต่างกันในวิธีการเข้าถึงโหนดของต้นไม้ตัดสินใจใน 2 วิธีคือ ใช้การวนซ้ำ (iterative method) หรือเข้าถึงโหนดจากล่างขึ้นบนเพียงรอบเดียว (single-scan bottom-up strategy) ความไม่ชัดเจนอีกข้อหนึ่งคือ การเลือกกลุ่มของข้อมูลที่ใบของต้นไม้เมื่อตัดกิ่งแล้ว เป็นความสับสนระหว่างการใช้กลุ่มหลักของตัวอย่างข้อมูลฝึก หรือการใช้กลุ่มหลักของตัวอย่างข้อมูลการตัดกิ่ง

Esposito et al. (1997) ได้วิเคราะห์เพื่อตรวจสอบอัลกอริทึมของ reduced-error pruning สรุปว่าการเข้าถึงโหนดของต้นไม้ตัดสินใจโดยใช้การวนซ้ำไม่สามารถให้ผลที่ดีตามที่ Quinlan ต้องการได้ เมื่อเปรียบเทียบกับการเข้าถึงโหนดจากล่างขึ้นบนเพียงรอบเดียว และผลของวิเคราะห์เปรียบเทียบของ Esposito et al. ยังยืนยันแนวคิดของ Quinlan ได้ว่า ต้นไม้ตัดสินใจที่สร้างขึ้นด้วยวิธี reduced-error pruning จะมีความเที่ยงตรงสูงเมื่อเปรียบเทียบกับชุดข้อมูลการตัดกิ่งและมีขนาดเล็กที่ค่าความเที่ยงตรงในการจำแนกนั้น

ความลำเอียงในขั้นตอนการทำงานของ reduced-error pruning ได้รับการตรวจสอบโดย Oates and Jensen (1997, 1998) พวกเขาพบว่าค่าความผิดพลาด r_L ซึ่งแทนจำนวนตัวอย่างที่ไม่ถูกต้องเมื่อแทนที่ต้นไม้ย่อยด้วยใบ จะขึ้นอยู่กับโครงสร้างของต้นไม้ที่อยู่ด้านบนของโหนด N ที่พิจารณาขึ้นไป แต่เมื่อตรวจสอบค่าความผิดพลาด r_T ซึ่งแทนจำนวนตัวอย่างที่ไม่ถูกต้องของต้นไม้

ย่อยที่มีรากเป็นโหนด N เมื่อทำการตัดกิ่งต้นไม้ตัดสินใจด้วยการทำงานแบบล่างขึ้นบน ค่าของ r_T จะมีค่าลดลง จากการแทนที่โหนดด้านล่างด้วยใบที่สามารถปรับปรุงจำนวนตัวอย่างที่ไม่ถูกต้องให้ดีขึ้นได้ ซึ่งไม่มีผลกับค่าของ r_L ดังนั้นจึงมีความเป็นไปได้ที่ค่าของ $r_T < r_L$ ทำให้ต้นไม้ย่อยจะไม่ถูกตัดออกไปทั้งที่ไม่มีความสำคัญ และความลำเอียงของค่าความผิดพลาดนี้จะมีผลมากขึ้น ถ้าขนาดของต้นไม้ตัดสินใจมีจำนวนโหนดมาก หรือจำนวนตัวอย่างในชุดข้อมูลการตัดกิ่งมีจำนวนน้อย

เนื้อหาในส่วนต่อไปของงานวิจัยนี้ จะให้รายละเอียดของวิธีดำเนินการวิจัย โดยนำเอาความรู้และงานวิจัยต่าง ๆ ที่ได้ศึกษามาปรับปรุงและพัฒนาให้สามารถสร้างต้นไม้ตัดสินใจที่มีประสิทธิภาพ โดยใช้วิธีการตัดกิ่งต้นไม้ตัดสินใจ ที่มุ่งเน้นทดสอบกับข้อมูลทางด้านวิทยาศาสตร์ที่เกี่ยวของกับงานด้านการแพทย์และการศึกษาโครงสร้างทางพันธุกรรมของมนุษย์ โดยต้นไม้ตัดสินใจที่ได้จะมีความซับซ้อนลดลง และยังคงมีความแม่นตรงในการจำแนกกลุ่มของข้อมูลใหม่ได้อย่างถูกต้อง

บทที่ 3
วิธีดำเนินการวิจัย

งานวิจัยนี้มีจุดมุ่งหมายเพื่อพัฒนาเทคนิคการตัดกิ่งต้นไม้ตัดสินใจ สำหรับการทำเหมืองข้อมูลประเภทสังเคราะห์โมเดลเพื่อใช้ในการจำแนกข้อมูล โดยเน้นการพัฒนาวิธีการตัดกิ่งต้นไม้ตัดสินใจที่เหมาะสมสำหรับข้อมูลทางด้านวิทยาศาสตร์ ที่ประกอบด้วยข้อมูลทางด้านการแพทย์และข้อมูลโครงสร้างทางพันธุกรรมของมนุษย์ รายละเอียดเนื้อหาในบทนี้ประกอบด้วยหัวข้อ 3.1 ระเบียบวิธีวิจัย ในหัวข้อ 3.2 เป็นคำอธิบายข้อมูลที่ใช้ในการทดสอบประสิทธิภาพของเทคนิคการตัดกิ่งต้นไม้ตัดสินใจที่พัฒนาขึ้น ในหัวข้อ 3.3 เป็นการกล่าวถึงเครื่องมือที่ใช้ในการวิจัย ในหัวข้อ 3.4 เป็นการพัฒนาวิธีการตัดกิ่งต้นไม้ตัดสินใจ และหัวข้อ 3.5 กล่าวถึงรายละเอียดของการทดสอบเปรียบเทียบวิธีการตัดกิ่งต้นไม้ตัดสินใจ

3.1 ระเบียบวิธีวิจัย

การค้นคว้าวิจัยแบ่งขั้นตอนการทำงานออกเป็นดังนี้

3.1.1 ศึกษาเอกสารและสรุปงานวิจัยที่เกี่ยวข้อง

3.1.2 ศึกษาการใช้งานและขั้นตอนการทำงานของระบบ WEKA (Witten and Frank, 2005) ซึ่งเป็นซอฟต์แวร์ที่พัฒนาบนระบบเปิดเผยซอร์สโค้ดที่นิยมใช้ในการทำเหมืองข้อมูล โดยเน้นงานการจำแนกข้อมูลโดยใช้ต้นไม้ตัดสินใจ

3.1.3 คัดเลือกและเตรียมข้อมูลทางด้านวิทยาศาสตร์ โดยเน้นข้อมูลทางการแพทย์และข้อมูลโครงสร้างทางพันธุกรรมของมนุษย์ เพื่อใช้ทดสอบประสิทธิภาพของวิธีการตัดกิ่งต้นไม้ตัดสินใจมีจำนวนทั้งหมด 21 ชุดข้อมูล แบ่งออกเป็นข้อมูลทางการแพทย์จำนวน 18 ชุดข้อมูล และข้อมูลโครงสร้างทางพันธุกรรมของมนุษย์จำนวน 3 ชุดข้อมูล

3.1.4 ศึกษาขั้นตอนการทำงาน และดำเนินการเปรียบเทียบประสิทธิภาพของเทคนิคการตัดกิ่งต้นไม้ตัดสินใจ รวมทั้งเปรียบเทียบประสิทธิภาพของต้นไม้ตัดสินใจที่ไม่ได้ตัดกิ่งโดยเลือกทดสอบกับเทคนิคการตัดกิ่งต้นไม้ตัดสินใจ 2 วิธีคือ

1) วิธีการตัดกิ่งแบบความผิดพลาดลดลง (Reduced-error pruning)

2) วิธีการตัดกิ่งโดยใช้ค่าความผิดพลาด (Error-based pruning)

3.1.5 พัฒนาวิธีการตัดกิ่งต้นไม้ตัดสินใจที่เหมาะสมกับชุดข้อมูลทดสอบ โดยปรับปรุงการทำงานของวิธีการตัดกิ่งที่ได้ศึกษาเปรียบเทียบไว้แล้ว

3.1.6 ทำการทดสอบข้อมูลและเปรียบเทียบประสิทธิภาพ ของวิธีการตัดกิ่งต้นไม้ตัดสินใจที่พัฒนาขึ้น โดยประมวลผลโปรแกรมด้วยเครื่องคอมพิวเตอร์ PC รุ่น Pentium4 ความเร็ว 3.20 GHz หน่วยความจำหลัก 512 MB ทำงานบนระบบปฏิบัติการ Windows XP Professional Edition Service Pack 2

3.1.7 สรุป และวิเคราะห์ผลการทดสอบข้อมูล

3.2 แหล่งที่มาและรายละเอียดของข้อมูล

ข้อมูลที่ใช้ในการวิจัยเพื่อเปรียบเทียบประสิทธิภาพของวิธีการตัดกิ่งต้นไม้ตัดสินใจนั้น เราได้คัดเลือกชุดข้อมูลมาจาก แหล่งข้อมูลของมหาวิทยาลัยแห่งรัฐแคลิฟอร์เนีย เมืองเออร์ไวน์ ประเทศสหรัฐอเมริกา (University of California at Irvine) (Blake and Merz, 1998) ซึ่งเป็นที่รวบรวมชุดข้อมูลสำหรับเป็นเกณฑ์ในการทดสอบประสิทธิภาพของอัลกอริทึมต่าง ๆ ของงานการทำเหมืองข้อมูล คัดเลือกข้อมูลเอาไว้จำนวนทั้งหมด 21 ชุดข้อมูล แบ่งออกเป็นข้อมูลทางการแพทย์จำนวน 18 ชุดข้อมูล และข้อมูลโครงสร้างทางพันธุกรรมของมนุษย์จำนวน 3 ชุดข้อมูลโดยในการคัดเลือกชุดข้อมูลจะพิจารณาลักษณะของข้อมูลดังนี้

3.2.1 เป็นชุดข้อมูลทางด้านวิทยาศาสตร์ที่เกี่ยวข้องกับทางด้านการแพทย์ หรือเป็นชุดข้อมูลโครงสร้างทางพันธุกรรมของมนุษย์

3.2.2 เป็นชุดข้อมูลที่มีจำนวนกลุ่มของข้อมูลไม่มากกว่า 6 กลุ่ม เนื่องจากเทคนิคการตัดกิ่งที่พัฒนาขึ้นต้องใช้ค่าวิกฤตของการทดสอบไคสแควร์มาเปรียบเทียบด้วย ซึ่งค่าระดับความเป็นอิสระ (degree of freedom, df) ที่ใช้สำหรับการทดสอบในงานวิจัยนี้มีค่าสูงสุดได้ไม่เกิน 30

รายชื่อชุดข้อมูล จำนวนตัวอย่าง จำนวนแอททริบิวต์จากการจำแนกคุณลักษณะต่าง ๆ และจำนวนกลุ่มในแต่ละชุดข้อมูลมีรายละเอียดดังตารางที่ 3.1

ตารางที่ 3.1 รายละเอียดของชุดข้อมูลโดยสรุปที่ใช้ในการวิจัย

ชื่อชุดข้อมูล	จำนวน	จำแนกจำนวนแอททริบิวต์				จำนวน
	ตัวอย่าง	ทั้งหมด	ข้อความ	ตัวเลข	การสูญหาย	กลุ่ม
1. Allbp	2800 (972)	30	22	7	yes	3
2. Allhyper	2800 (972)	30	22	7	yes	5
3. Allhypo	2800 (972)	30	22	7	yes	5
4. Allrep	2800 (972)	30	22	7	yes	4
5. Breast-cancer	286	10	9	0	yes	2
6. Breast-w	699	10	0	9	yes	2
7. Dermatology	366	35	33	1	yes	6
8. Diabetes	768	8	0	7	no	2
9. DNA	2000(1186)	61	60	0	no	3
10. Echocardiogram	132	12	2	9	yes	2
11. Heart-disease	920	14	8	5	yes	5
12. Heart-h	294	14	7	6	yes	2
13. Heart-Statlog	270	14	0	13	no	2
14. Hepatitis	155	20	13	6	yes	2
15. Hypothyroid	3163	26	18	7	yes	2
16. Liver-disorders	345	7	0	6	no	2
17. Lung cancer	32	57	0	56	yes	3
18. Promoters	106	58	57	0	no	2
19. Sick-euthyroid	3163	26	18	7	yes	2
20. Splice-junction	3190	61	60	0	no	3
21. Thyroid	215	6	0	5	no	3

หมายเหตุ ตัวเลขในวงเล็บเป็นจำนวนตัวอย่างของข้อมูลทดสอบที่ได้แบ่งไว้แล้วของชุดข้อมูลนั้น เพื่อใช้ทดสอบความแม่นตรงในการจำแนกกลุ่มข้อมูลของต้นไม้ตัดสินใจที่สร้างขึ้น

ข้อมูลแต่ละชุดมีรายละเอียดและการจำแนกกลุ่มของข้อมูลดังนี้

1) Allbp เป็นข้อมูลการวินิจฉัยความผิดปกติของต่อมธัยรอยด์
มีจำนวนข้อมูลฝึก 2,800 เรคคอร์ด (จำนวนข้อมูลทดสอบ 972 เรคคอร์ด) จำแนกออกเป็น 3 กลุ่ม คือ

increased binding protein	(จำนวน 124 เรคคอร์ด)
decreased binding protein	(จำนวน 9 เรคคอร์ด)
negative	(จำนวน 2,667 เรคคอร์ด)

2) Allhyper เป็นข้อมูลการวินิจฉัยความผิดปกติของต่อมธัยรอยด์
มีจำนวนข้อมูลฝึก 2,800 เรคคอร์ด (จำนวนข้อมูลทดสอบ 972 เรคคอร์ด) จำแนกออกเป็น 5 กลุ่ม คือ

hyperthyroid	(จำนวน 62 เรคคอร์ด)
T3 toxic	(จำนวน 8 เรคคอร์ด)
goitre	(จำนวน 7 เรคคอร์ด)
secondary toxic	(จำนวน 0 เรคคอร์ด)
negative	(จำนวน 2,723 เรคคอร์ด)

3) Allhypo เป็นข้อมูลการวินิจฉัยความผิดปกติของต่อมธัยรอยด์
มีจำนวนข้อมูลฝึก 2,800 เรคคอร์ด (จำนวนข้อมูลทดสอบ 972 เรคคอร์ด) จำแนกออกเป็น 5 กลุ่ม คือ

hypothyroid	(จำนวน 0 เรคคอร์ด)
primary hypothyroid	(จำนวน 64 เรคคอร์ด)
compensated hypothyroid	(จำนวน 154 เรคคอร์ด)
secondary hypothyroid	(จำนวน 2 เรคคอร์ด)
negative	(จำนวน 2,580 เรคคอร์ด)

4) Allrep เป็นข้อมูลการวินิจฉัยความผิดปกติของต่อมธัยรอยด์
มีจำนวนข้อมูลฝึก 2,800 เรคคอร์ด (จำนวนข้อมูลทดสอบ 972 เรคคอร์ด) จำแนกออกเป็น 5 กลุ่ม คือ

replacement therapy	(จำนวน 29 เรคคอร์ด)
underreplacement	(จำนวน 35 เรคคอร์ด)
overreplacement	(จำนวน 23 เรคคอร์ด)
negative	(จำนวน 2,713 เรคคอร์ด)

5) Breast-cancer เป็นข้อมูลการวินิจฉัยการเกิดขึ้นใหม่ของมะเร็งเต้านม
มีจำนวนข้อมูล 286 เรคคอร์ด จำแนกออกเป็น 2 กลุ่ม คือ

no recurrence events	(จำนวน 201 เรคคอร์ด)
recurrence events	(จำนวน 85 เรคคอร์ด)

6) Breast-w เป็นข้อมูลการวินิจฉัยมะเร็งเต้านมว่าเป็นชนิดร้ายแรงหรือไม่
มีจำนวนข้อมูล 699 เรคคอร์ด จำแนกออกเป็น 2 กลุ่ม คือ

benign	(จำนวน 458 เรคคอร์ด)
malignant	(จำนวน 241 เรคคอร์ด)

7) Dermatology เป็นข้อมูลการวินิจฉัยโรคผิวหนัง
มีจำนวนข้อมูล 366 เรคคอร์ด จำแนกออกเป็น 6 กลุ่ม คือ

class 1 (psoriasis)	(จำนวน 112 เรคคอร์ด)
class 2 (seboreic dermatitis)	(จำนวน 61 เรคคอร์ด)
class 3 (lichen planus)	(จำนวน 72 เรคคอร์ด)
class 4 (pityriasis rosea)	(จำนวน 49 เรคคอร์ด)
class 5 (cronic dermatitis)	(จำนวน 52 เรคคอร์ด)
class 6 (pityriasis rubra pilaris)	(จำนวน 20 เรคคอร์ด)

8) Diabetes เป็นข้อมูลการทดสอบว่าคนไข้ส่ออาการโรคเบาหวานหรือไม่ โดยใช้มาตรฐานขององค์การอนามัยโลก ทดสอบกับคนไข้เพศหญิงที่เป็นชนเผ่าพื้นเมืองอินเดียนแดง รัฐอริโซน่า
มีจำนวนข้อมูล 768 เรคคอร์ด จำแนกออกเป็น 2 กลุ่ม คือ

tested negative	(จำนวน 500 เรคคอร์ด)
tested positive	(จำนวน 268 เรคคอร์ด)

9) DNA เป็นข้อมูลการจำแนกโครงสร้างทางพันธุกรรมของมนุษย์
มีจำนวนข้อมูลฝึก 2,000 เรคคอร์ด (จำนวนข้อมูลทดสอบ 1,186 เรคคอร์ด) จำแนกออกเป็น 3 กลุ่ม คือ

exon/intron	(จำนวน 464 เรคคอร์ด)
intron/exon	(จำนวน 485 เรคคอร์ด)
none	(จำนวน 1,051 เรคคอร์ด)

10) Echocardiogram เป็นข้อมูลที่บันทึกว่าคนไข้โรคหัวใจสามารถมีชีวิตอยู่ได้นานกว่าหนึ่งปีหรือไม่

มีจำนวนข้อมูล 132 เรคคอร์ด จำแนกออกเป็น 2 กลุ่ม คือ

class 0 (dead) (จำนวน 88 เรคคอร์ด)

class 1 (alive) (จำนวน 43 เรคคอร์ด)

11) Heart disease เป็นข้อมูลการวินิจฉัยอาการหลอดเลือดเลี้ยงหัวใจตีบโดยใช้ภาพเอ็กซเรย์

มีจำนวนข้อมูล 920 เรคคอร์ด จำแนกออกเป็น 5 กลุ่ม คือ

class 0 (จำนวน 411 เรคคอร์ด)

class 1 (จำนวน 265 เรคคอร์ด)

class 2 (จำนวน 109 เรคคอร์ด)

class 3 (จำนวน 107 เรคคอร์ด)

class 4 (จำนวน 28 เรคคอร์ด)

12) Heart-h เป็นข้อมูลการวินิจฉัยอาการหลอดเลือดเลี้ยงหัวใจตีบโดยใช้ภาพเอ็กซเรย์ โดยทดสอบกับคนไข้ชาวฮังกาเรียน

มีจำนวนข้อมูล 294 เรคคอร์ด จำแนกออกเป็น 2 กลุ่ม คือ

< 50% diameter narrowing (จำนวน 188 เรคคอร์ด)

≥ 50% diameter narrowing (จำนวน 106 เรคคอร์ด)

13) Heart-Statlog เป็นข้อมูลการทดสอบว่าคนไข้มีอาการของโรคหัวใจหรือไม่

มีจำนวนข้อมูล 270 เรคคอร์ด จำแนกออกเป็น 2 กลุ่ม คือ

absent (จำนวน 150 เรคคอร์ด)

present (จำนวน 120 เรคคอร์ด)

14) Hepatitis เป็นข้อมูลที่บันทึกว่าคนไข้โรคตับอักเสบยังมีชีวิตอยู่หรือไม่

มีจำนวนข้อมูล 155 เรคคอร์ด จำแนกออกเป็น 2 กลุ่ม คือ

class 1 (die) (จำนวน 32 เรคคอร์ด)

class 2 (live) (จำนวน 123 เรคคอร์ด)

15) Hypothyroid เป็นข้อมูลการวินิจฉัยความผิดปกติของต่อมธัยรอยด์

มีจำนวนข้อมูล 3,163 เรคคอร์ด จำแนกออกเป็น 2 กลุ่ม คือ

hypothyroid (จำนวน 151 เรคคอร์ด)

negative (จำนวน 3,012 เรคคอร์ด)

16) Liver-disorders เป็นข้อมูลการทดสอบความผิดปกติของตับของผู้ชายโสดที่นิยมดื่มแอลกอฮอล์

มีจำนวนข้อมูล 345 เรคคอร์ด จำแนกออกเป็น 2 กลุ่ม คือ

class 1	(จำนวน 145 เรคคอร์ด)
class 2	(จำนวน 200 เรคคอร์ด)

17) Lung cancer เป็นข้อมูลการวินิจฉัยประเภทของมะเร็งปอด
มีจำนวนข้อมูล 32 เรคคอร์ด จำแนกออกเป็น 3 กลุ่ม คือ

class 1	(จำนวน 9 เรคคอร์ด)
class 2	(จำนวน 13 เรคคอร์ด)
class 3	(จำนวน 10 เรคคอร์ด)

18) Promoters เป็นข้อมูลการจำแนกโครงสร้างทางพันธุกรรมของมนุษย์
มีจำนวนข้อมูล 106 เรคคอร์ด จำแนกออกเป็น 2 กลุ่ม คือ

class +	(จำนวน 53 เรคคอร์ด)
class –	(จำนวน 53 เรคคอร์ด)

19) Sick-euthyroid เป็นข้อมูลการวินิจฉัยความผิดปกติของต่อมธัยรอยด์
มีจำนวนข้อมูล 3,163 เรคคอร์ด จำแนกออกเป็น 2 กลุ่ม คือ

sick euthyroid	(จำนวน 293 เรคคอร์ด)
negative	(จำนวน 2,870 เรคคอร์ด)

20) Splice-junction เป็นข้อมูลการจำแนกโครงสร้างทางพันธุกรรมของมนุษย์
มีจำนวนข้อมูล 3,190 เรคคอร์ด จำแนกออกเป็น 3 กลุ่ม คือ

exon -> intron (EI)	(จำนวน 767 เรคคอร์ด)
intron -> exon (IE)	(จำนวน 768 เรคคอร์ด)
neither (N)	(จำนวน 1,655 เรคคอร์ด)

21) Thyroid เป็นข้อมูลการจำแนกความผิดปกติของการวินิจฉัยต่อมธัยรอยด์
มีจำนวนข้อมูล 215 เรคคอร์ด จำแนกออกเป็น 3 กลุ่ม คือ

class 1 (normal)	(จำนวน 150 เรคคอร์ด)
class 2 (hyper)	(จำนวน 35 เรคคอร์ด)
class 3 (hypo)	(จำนวน 30 เรคคอร์ด)

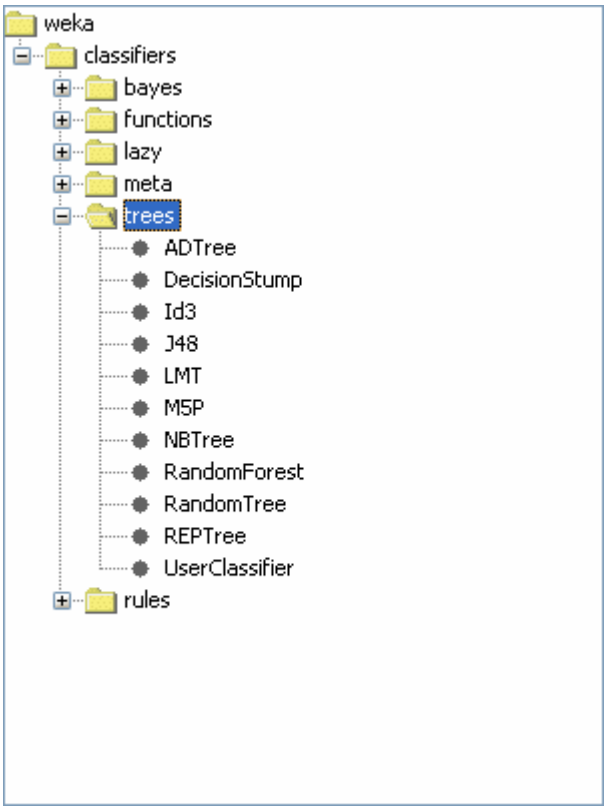
3.3 เครื่องมือที่ใช้ในการวิจัย

ระบบ WEKA

รูปที่ 3.1 หน้าจอหลักของระบบ WEKA

WEKA (http://www.cs.waikato.ac.nz/~ml/weka/) เป็นโปรแกรมที่พัฒนาขึ้นโดยมหาวิทยาลัย Waikato ประเทศนิวซีแลนด์ เพื่อใช้ในการวิเคราะห์ข้อมูลและหาความสัมพันธ์ของข้อมูลที่มีอยู่ ซึ่งจะทำให้เราสามารถวิเคราะห์แนวโน้ม หรือความเป็นไปได้ต่าง ๆ โดยผู้ใช้สามารถเลือกอัลกอริทึมที่จะใช้ในการวิเคราะห์ข้อมูลได้ โดยที่ WEKA ได้รวบรวมอัลกอริทึมที่ใช้ในการทำเหมืองข้อมูลไว้ในหลายแขนงด้วยกัน อาทิเช่น Classification, Clustering, Association, Regression เป็นต้น ซึ่งแต่ละอัลกอริทึมถูกพัฒนาขึ้นจากภาษา Java ที่สามารถนำไปใช้งานได้ ในทุก ๆ แพล็ตฟอร์ม (platform) โดยสิ่งที่เป็นจุดเด่นของ WEKA ก็คืองานในสาขาของ Classification ที่ได้รวบรวมอัลกอริทึมไว้มากมาย เช่น Bayesian Algorithm (ตัวอย่าง NaïveBayes), Lazy (ตัวอย่าง

IBk), Decision trees (ตัวอย่าง J48, ID3) และอื่น ๆ โดยในงานวิจัยนี้เราจะใช้ WEKA เวอร์ชั่น 3-4-7 ในการทดสอบข้อมูลและใช้อัลกอริทึม J48 เพื่อใช้สร้างต้นไม้ตัดสินใจ



รูปที่ 3.2 อัลกอริทึมในกลุ่มของ Classification

รูปแบบและชนิดของข้อมูลที่ใช้ในระบบ WEKA

ชุดข้อมูลที่ใช้ในระบบ WEKA จะต้องอยู่ในรูปแบบ ARFF (Attribute-Relation File Format) ซึ่ง ARFF ประกอบด้วย 2 ส่วนหลักดังนี้

1) Header หรือส่วนหัว คือส่วนที่แสดงรายละเอียดเกี่ยวกับ ชื่อของชุดข้อมูล รายการของแอททริบิวต์ที่ใช้และชนิดของแอททริบิวต์ ซึ่งจะขึ้นต้นด้วย @ และตามด้วยคำหลักอื่นๆ เช่น @relation เป็นต้น
2) Data คือส่วนที่แสดงรายการของข้อมูลทั้งหมด ซึ่งข้อมูลทั้งหมดจะแสดงอยู่หลังจากบรรทัด @data ดังแสดงตัวอย่างในรูปที่ 3.3 ซึ่งเป็นตัวอย่างของชุดข้อมูล Iris

รูปแบบของ ARFF อาจสรุปได้ดังในรูปที่ 3.4

```
% 1. Title: Iris Plants Database

% 2. Sources:
%      (a) Creator: R.A. Fisher
%      (b) Donor: Michael Marshall (MARSHALL%PLU@io.arc.nasa.gov)
%      (c) Date: July, 1988
%
@RELATION iris
@ATTRIBUTE sepallength  NUMERIC
@ATTRIBUTE sepalwidth   NUMERIC
@ATTRIBUTE petallength  NUMERIC
@ATTRIBUTE petalwidth   NUMERIC
@ATTRIBUTE class        {Iris-setosa,Iris-versicolor,Iris-
virginica}
                                                    -------→ สิ้นสุดส่วนหัว
@DATA
5.1,3.5,1.4,0.2,Iris-setosa
4.9,3.0,1.4,0.2,Iris-setosa
4.7,3.2,1.3,0.2,Iris-setosa
4.6,3.1,1.5,0.2,Iris-setosa
5.0,3.6,1.4,0.2,Iris-setosa
5.4,3.9,1.7,0.4,Iris-setosa
4.6,3.4,1.4,0.3,Iris-setosa
5.0,3.4,1.5,0.2,Iris-setosa
4.4,2.9,1.4,0.2,Iris-setosa
4.9,3.1,1.5,0.1,Iris-setosa
```

รูปที่ 3.3 แสดงตัวอย่างของข้อมูลที่อยู่ในรูปแบบ ARFF

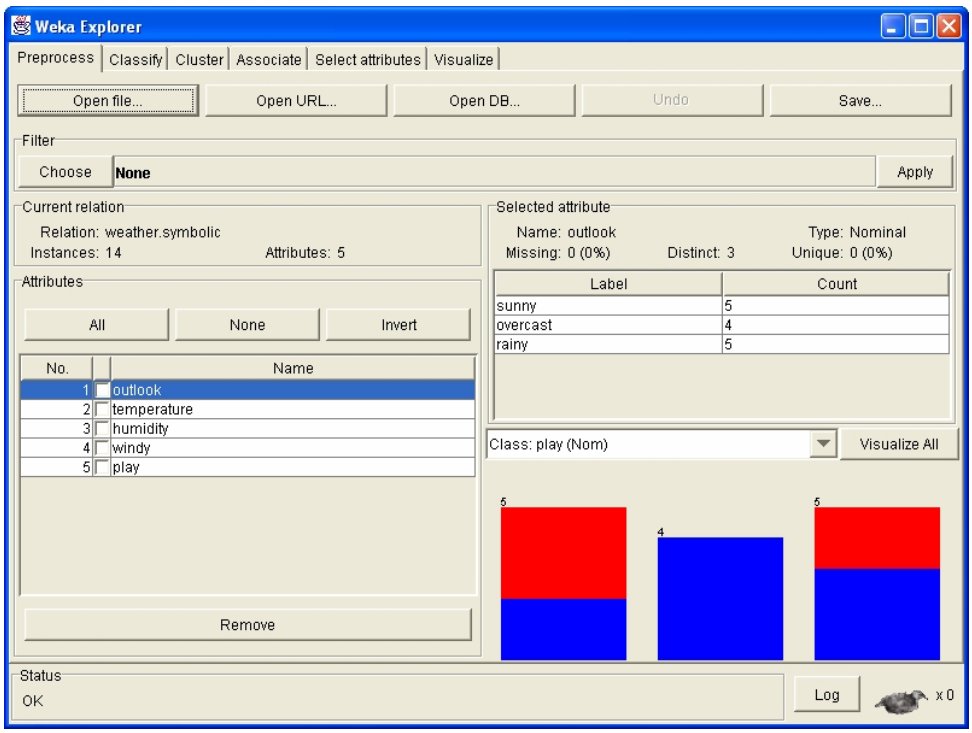
```
<ARFF-dataset>   := <header-section><data-section>

<header-section> := <relation-section><attribute-section>

<relation-section> := @relation <relation-name> \n
<attribute-section> := <attribute-declare> [<attribute-list>]
<attribute-list> := <attribute-declare> [<attribute-list>]
<attribute-declare>  := @attribute <attribute-name> <datatype> \n

<datatype> := <numeric> | <nominal-specification> | string |
            date [<date-format>]
<numeric> := real | integer
<nominal-specification> := {<nominal-name-list>}
<nominal-name-list> := <nominal-name>[, <nominal-name-list> ]

<data-section> := @data \n <data-instances>

<data-instances> := <data-instance> \n
<data-instance> := <data-value> [, <value-list>]
<value-list> := <data-value> [, <value-list>]
```

รูปที่ 3.4 สรุปรูปแบบ ARFF

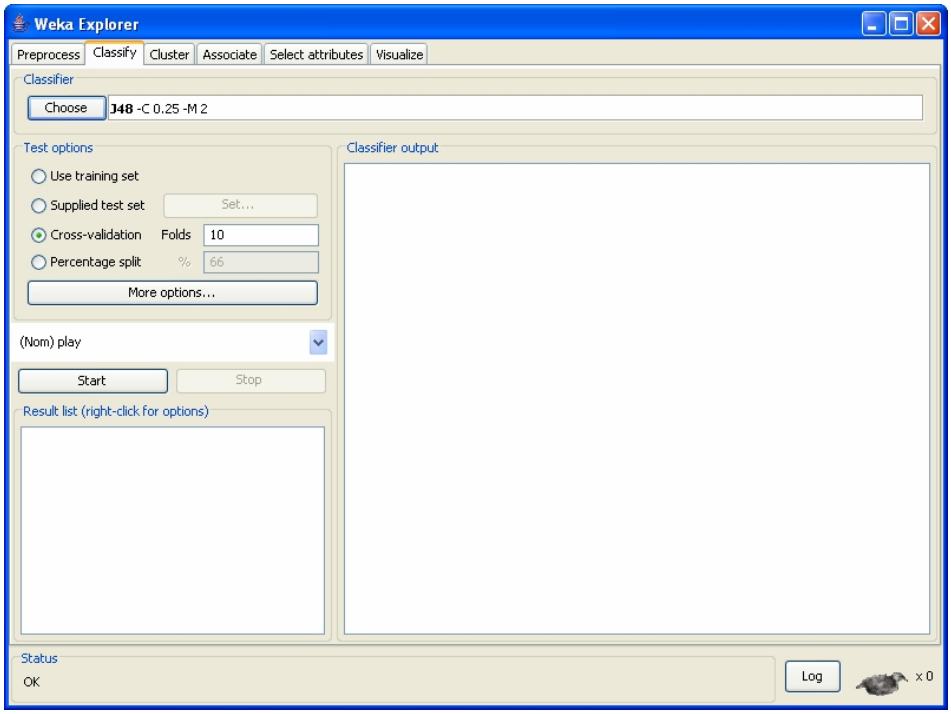
การใช้งาน Classifier ในระบบ WEKA

การใช้งาน Classifier ในระบบ WEKA เบื้องต้นนั้นทำได้โดย

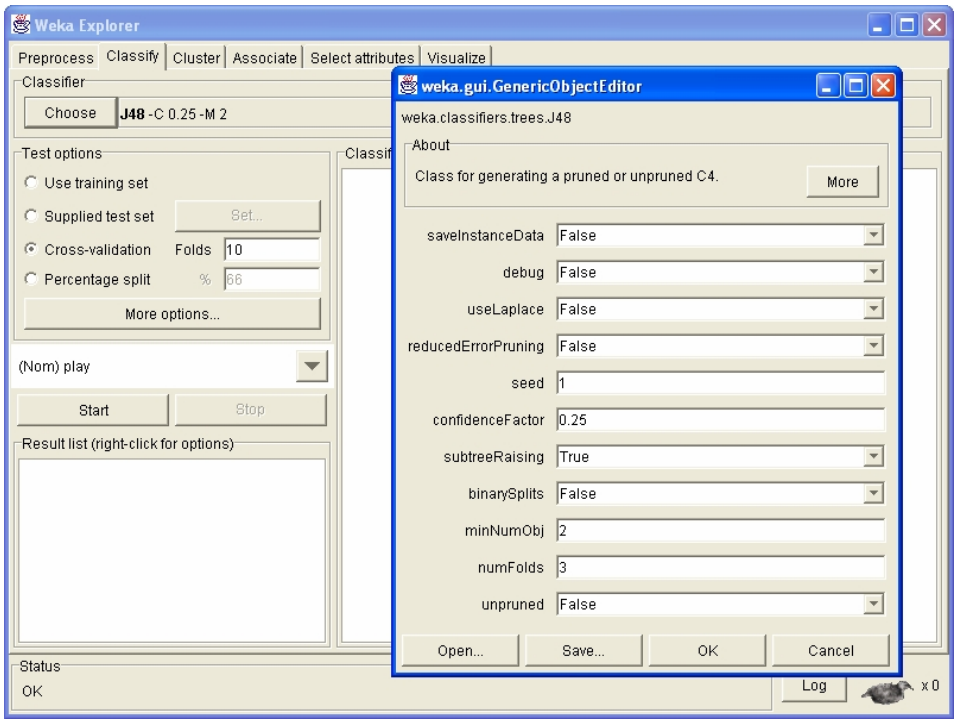
1) จากรูปที่ 3.1 คลิกที่ปุ่ม Explorer เพื่อเปิดโปรแกรม Weka Explorer
2) คลิกที่ปุ่ม Open file… เพื่อเลือกชุดข้อมูลในรูปแบบ ARFF (.arff)
3) เมื่อเลือกชุดข้อมูลแล้ว WEKA จะแสดงรายละเอียดของข้อมูลนั้น เช่น ชื่อของชุดข้อมูล, จำนวนตัวอย่างทั้งหมด, จำนวนแอททริบิวต์ และรายละเอียดอื่น ๆ ของข้อมูลโดยสังเขป (ดังตัวอย่างในรูปที่ 3.5 ที่แสดงรายละเอียดข้อมูล weather.symbolic)
4) จากนั้นคลิกที่แท็บ Classify และคลิกที่ปุ่ม Choose เพื่อเลือกอัลกอริทึมที่ใช้ในการวิเคราะห์ข้อมูล โดยในที่นี้จะใช้อัลกอริทึม J48 ซึ่งเป็นอัลกอริทึมในกลุ่มของ trees
5) โดยค่าเริ่มต้นจะเลือกแอททริบิวต์ที่อยู่ลำดับท้ายสุดเป็นกลุ่มของข้อมูล และกำหนดค่า Test options เป็น Cross-validation Folds:10 ดังแสดงในรูปที่ 3.6
6) เราสามารถปรับค่าพารามิเตอร์อื่น ๆ ของอัลกอริทึม J48 ได้ โดยการคลิกที่ชื่อของอัลกอริทึม จะปรากฏหน้าต่างสำหรับการปรับค่าพารามิเตอร์แต่ละตัวแสดงได้ในรูปที่ 3.7 ซึ่งรายละเอียดของพารามิเตอร์แต่ละตัวแสดงไว้ในตารางที่ 3.2



รูปที่ 3.5 WEKA Explorer: แสดงรายละเอียดของข้อมูลนั้นบนหน้าจอ



รูปที่ 3.6 WEKA Explorer: แสดงรายละเอียดต่าง ๆ ของแท็บ Classify



รูปที่ 3.7 WEKA Explorer: แสดงหน้าต่างสำหรับปรับค่าพารามิเตอร์ต่าง ๆ ของอัลกอริทึม J48

ตารางที่ 3.2 พารามิเตอร์ต่าง ๆ ของอัลกอริทึม J48

ชื่อพารามิเตอร์	คำสั่ง	ค่าเริ่มต้น	คำอธิบาย
saveInstanceData	-L	False	เป็นการระบุว่าต้องการที่จะเก็บชุดข้อมูลฝึกไว้สำหรับการทำ Visualization หรือไม่ ถ้าไม่ข้อมูลนั้นจะถูกลบทิ้งไปหลังจากการสร้างต้นไม้ตัดสินใจ
debug		False	ถ้ากำหนดเป็น True ทำให้ Classifier แสดงข้อมูลเพิ่มเติมบน Console ของระบบ
useLaplace	-A	False	ใช้ Laplace Smoothingในการทำนายความน่าจะเป็นของแต่ละโหนดใบ
reducedErrorPruning	-R	False	ใช้วิธีการตัดกิ่งต้นไม้ตัดสินใจ Reduced-error pruning แทนวิธีการตัดกิ่งต้นไม้ตัดสินใจ Error-based pruning
seed	-Q	1	กำหนดจำนวน seed สำหรับการสุ่ม เมื่อใช้วิธีการตัดกิ่งต้นไม้ Reduced-error pruning
confidenceFactor	-C	0.25	กำหนดค่าระดับความเชื่อมั่น ที่ใช้ประกอบการพิจารณาในการตัดกิ่งต้นไม้ Error-based pruning
subtreeRaising	-S	True	สามารถแทนที่ต้นไม้ย่อยด้วยต้นไม้ย่อยที่อยู่ด้านล่างได้ ขณะทำการตัดกิ่งต้นไม้ตัดสินใจ
binarySplits	-B	False	เพื่อใช้การแบ่งข้อมูลแบบ binary สำหรับแอททริบิวต์ที่เป็นชนิดข้อความในขณะที่ทำการสร้างต้นไม้
minNumObj	-M	2	กำหนดจำนวนตัวอย่างขั้นต่ำของโหนดใบแต่ละโหนด
numFolds	-N	3	กำหนดจำนวนโฟลด์ของข้อมูลที่ใช้สำหรับ Reduced-error pruning โดยที่หนึ่งโฟลด์ จะถูกใช้ในการตัดกิ่งของต้นไม้ และโฟลด์ที่เหลือจะใช้ในการสร้างต้นไม้
unpruned	-U	False	กำหนดค่าพารามิเตอร์นี้เป็น True เพื่อการสร้างต้นไม้ตัดสินใจโดยไม่มีการตัดกิ่งเลย และอาจทำให้ได้ต้นไม้ที่มีความซับซ้อนได้

3.4 การพัฒนาวิธีการตัดกิ่งต้นไม้ตัดสินใจ

วิธีการตัดกิ่งต้นไม้ตัดสินใจที่พัฒนาขึ้นสำหรับงานวิจัยนี้ เป็นวิธีที่นำเอาเทคนิคการตัดกิ่งต้นไม้ตัดสินใจ reduced-error pruning (REP) มาปรับปรุงและพัฒนาต่อไปให้มีประสิทธิภาพสูงขึ้น เนื่องจากเป็นเทคนิคการตัดกิ่งต้นไม้ตัดสินใจที่ทำงานได้รวดเร็ว สามารถสร้างต้นไม้ตัดสินใจที่มีขนาดเล็กและให้ค่าความแม่นตรงสูง สามารถจำแนกกลุ่มของข้อมูลใหม่ได้อย่างมีประสิทธิภาพ กำหนดให้

$E_T(N)$ แทนจำนวนตัวอย่างที่ไม่ถูกต้องของต้นไม้ย่อยที่มีรากเป็นโหนด N

$E_L(N)$ แทนจำนวนตัวอย่างที่ไม่ถูกต้อง เมื่อเปลี่ยนต้นไม้ย่อยเป็นใบ

เมื่อต้นไม้ตัดสินใจถูกทดสอบด้วยชุดข้อมูลการตัดกิ่งและให้ค่า $E_T(N) \geq E_L(N)$ แล้วจะทำการเปลี่ยนโหนด N เป็นใบ จะเห็นได้ว่า $E_L(N)$ ขึ้นอยู่กับโครงสร้างของต้นไม้ที่อยู่ด้านบน N ขึ้นไป ซึ่งเมื่อทำการตัดกิ่งต้นไม้ตัดสินใจด้วยการทำงานแบบล่างขึ้นบน $E_L(N)$ จะยังคงมีค่าคงที่แม้ว่าโครงสร้างต้นไม้ด้านล่างของ N เปลี่ยนแปลงไป ส่วนค่าของ $E_T(N)$ โดยปกติจะมีค่าลดลง เนื่องจากมีการเปลี่ยนโหนดด้านล่างเป็นใบ ที่สามารถปรับปรุงจำนวนตัวอย่างที่ไม่ถูกต้องให้ดีขึ้นได้ จากสาเหตุข้างต้นสามารถสรุปได้ดังนี้

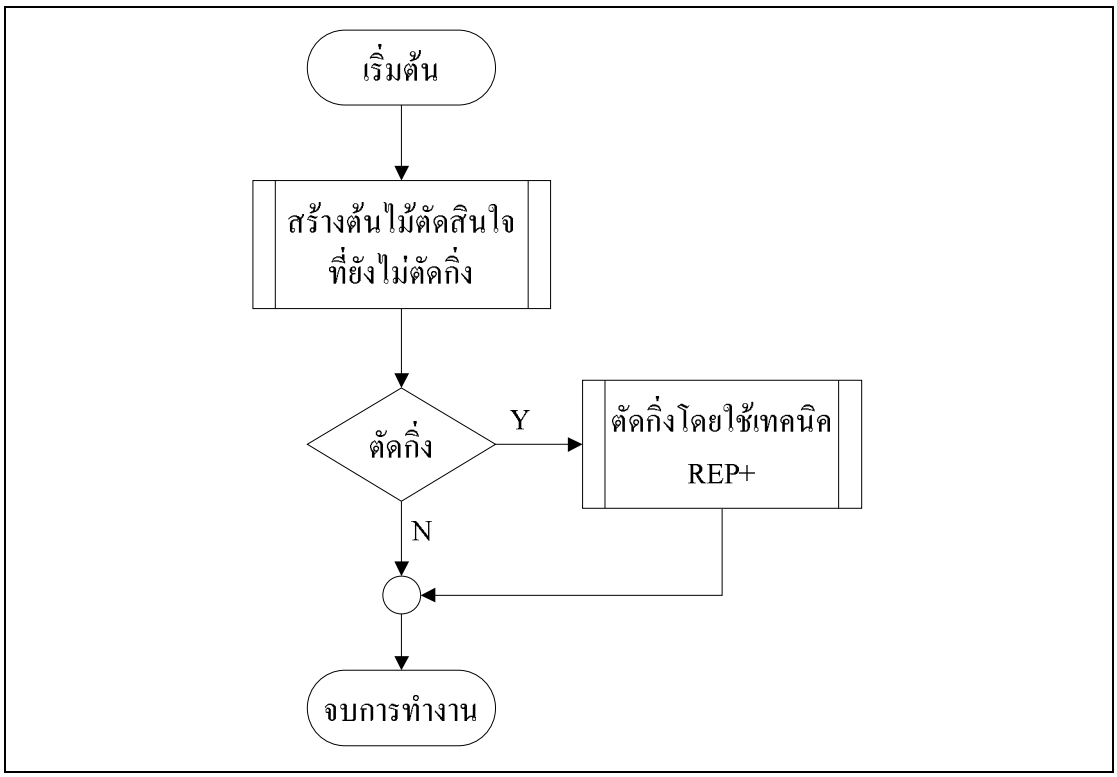
1) การตัดกิ่งต้นไม้ตัดสินใจที่อยู่ด้านล่างของโหนด N เพิ่มความเป็นไปได้ที่ทำให้ $E_T(N)$ มีค่าน้อยกว่า $E_L(N)$ ซึ่งส่งผลให้โหนด N จะไม่ถูกตัดออกไป
2) ค่าความผิดพลาด $E_T(N)$ ของต้นไม้ย่อยที่ได้ มีแนวโน้มจะถูกให้ความสำคัญที่ต่ำเกินไป (under-estimate)
3) การใช้ชุดข้อมูลการตัดกิ่งที่แยกต่างหากจากชุดข้อมูลฝึก สำหรับตัดกิ่งต้นไม้ตัดสินใจอาจทำให้เกิดปัญหาการเจาะจงโมเดลมากเกินไปกับชุดข้อมูลการตัดกิ่งได้ เช่นเดียวกับที่เกิดขึ้นในชุดข้อมูลฝึกสำหรับสร้างต้นไม้ตัดสินใจ

ดังนั้นจึงต้องทำการปรับปรุงวิธีการตัดกิ่งต้นไม้ตัดสินใจ reduced-error pruning โดยการนำค่าทางสถิติมาประกอบการตัดสินใจ เพื่อตรวจสอบว่าสามารถเปลี่ยนโหนดภายในของต้นไม้ย่อยเป็นใบได้หรือไม่ โดยกำหนดให้ต้นไม้ย่อยจะไม่เปลี่ยนแปลง ถ้ากลุ่มของข้อมูลที่ได้จากการทำนายของต้นไม้ตัดสินใจมีความสัมพันธ์กันอย่างมีนัยสำคัญ กับกลุ่มของข้อมูลที่เป็นจริงจากชุดข้อมูลฝึก เนื้อหาในหัวข้อถัดไปจะกล่าวถึง ขั้นตอนการทำงานของเทคนิคการตัดกิ่งต้นไม้ตัดสินใจที่พัฒนาขึ้นที่ให้ชื่อว่า extended reduced-error pruning (REP+)

3.4.1 การทำงานของเทคนิค Extended reduced-error pruning (REP+)

การทำงานของเทคนิคการตัดกิ่งต้นไม้ตัดสินใจ REP+ เป็นการตัดกิ่งของต้นไม้ตัดสินใจที่ถูกสร้างขึ้นสมบูรณ์แล้ว โดยตรวจสอบความต้องการว่าต้องการให้ตัดกิ่งหรือไม่ ถ้าไม่

ต้องการให้ตัดกิ่งจะได้ต้นไม้ตัดสินใจที่ยังมีโครงสร้างเช่นเดิม ซึ่งอาจจะมีความซับซ้อนมากเกินไป และจำแนกข้อมูลใหม่ได้อย่างไม่มีประสิทธิภาพ สามารถแสดงผังการทำงานได้ดังรูปที่ 3.8



รูปที่ 3.8 แสดงผังการทำงานของการสร้างต้นไม้ตัดสินใจและการตัดกิ่ง

เทคนิคการตัดกิ่งต้นไม้ตัดสินใจ REP+ พัฒนาขึ้นจากภาษาโปรแกรม Java โดยนำเทคนิคการตัดกิ่งต้นไม้ตัดสินใจ REP ที่พัฒนาบนระบบเปิดเผยซอร์สโค้ดที่ชื่อว่า WEKA มาปรับปรุงและพัฒนาต่อไปให้มีประสิทธิภาพสูงขึ้น ขั้นตอนการทำงานของเทคนิคการตัดกิ่ง REP+ สามารถแสดงผังการทำงานได้ดังรูปที่ 3.9 และ 3.10 โดยอธิบายขั้นตอนการทำงานได้ดังต่อไปนี้

1) ทำการแบ่งชุดข้อมูลฝึกไว้สำหรับใช้เป็นชุดข้อมูลการตัดกิ่ง โดยชุดข้อมูลนี้จะไม่นำไปใช้ในขั้นตอนการสร้างต้นไม้ตัดสินใจ แต่จะใช้สำหรับขั้นตอนการตัดกิ่งเท่านั้น โดยข้อมูลที่แบ่งออกแต่ละส่วนจะมีข้อมูลครบทุกกลุ่มด้วยสัดส่วนเดียวกับข้อมูลตั้งต้น

2) เข้าถึงโหนดของต้นไม้จากล่างสุดขึ้นไปยังรากของต้นไม้ โดยใช้วิธีการเข้าถึงโหนดแบบ post-order traversal

3) คำนวณค่าความผิดพลาดในการจำแนกกลุ่มข้อมูลของโหนดที่เปลี่ยนเป็นใบ

โดยการทดสอบกับชุดข้อมูลการตัดกิ่ง

4) คำนวณค่าความผิดพลาดในการจำแนกกลุ่มข้อมูลที่โหนดลูกของโหนดที่พิจารณา โดยการทดสอบกับชุดข้อมูลการตัดกิ่ง

5) เปรียบเทียบค่าความผิดพลาดที่คำนวณได้ ถ้าค่าความผิดพลาดของโหนดที่เปลี่ยนเป็นใบมีค่าน้อยกว่าหรือเท่ากับค่าความผิดพลาดที่โหนดลูกของโหนดที่พิจารณาแล้ว จะดำเนินการตัดโหนดนั้นแล้วเปลี่ยนเป็นใบ โดยที่มีกลุ่มของข้อมูลเป็นกลุ่มที่มีข้อมูลสนับสนุนมากที่สุด

6) ถ้าค่าความผิดพลาดของโหนดที่เปลี่ยนเป็นใบมีค่ามากกว่าค่าความผิดพลาดที่โหนดลูกของโหนดที่พิจารณาแล้ว จะดำเนินการตรวจสอบความสัมพันธ์กันอย่างมีนัยสำคัญระหว่างกลุ่มของข้อมูลที่ได้จากการทำนายของต้นไม้ตัดสินใจและกลุ่มของข้อมูลที่เป็นจริงจากชุดข้อมูลฝึกต่อไป

7) กำหนดค่าระดับความมีนัยสำคัญ (α) ให้กับเทคนิคการตัดกิ่ง REP+ โดยสามารถกำหนดได้ 6 ค่าได้แก่ 0.01, 0.025, 0.05, 0.1, 0.25 และ 0.5

8) สร้างตารางการณ์จรจากชุดข้อมูลฝึก โดยทางด้านคอลัมน์แทนจำนวนกลุ่มของข้อมูลที่เป็นจริงจากชุดข้อมูลฝึก ส่วนทางด้านแถวแทนจำนวนกลุ่มของข้อมูลที่ได้จากการทำนายโดยใช้ต้นไม้ตัดสินใจ

9) คำนวณค่าระดับความเป็นอิสระ (df) สำหรับข้อมูลที่จัดลงในตารางการณ์จร โดยหาค่าได้จากผลคูณของจำนวนกลุ่มข้อมูลลบออกด้วย 1

10) เปรียบเทียบค่าองศาความเป็นอิสระ ถ้ามีค่าไม่เท่ากับ 1 แล้วจะดำเนินการหาความสัมพันธ์ระหว่างกลุ่มของข้อมูลโดยใช้การทดสอบไคสแควร์ แต่ถ้าค่าองศาความเป็นอิสระเท่ากับ 1 และมีจำนวนข้อมูลทั้งหมดในตารางการณ์จรมากกว่า 20 แล้วจะแก้ไขความคลาดเคลื่อนและดำเนินการหาความสัมพันธ์ระหว่างกลุ่มของข้อมูล โดยใช้การทดสอบไคสแควร์จากสูตรของ Yates

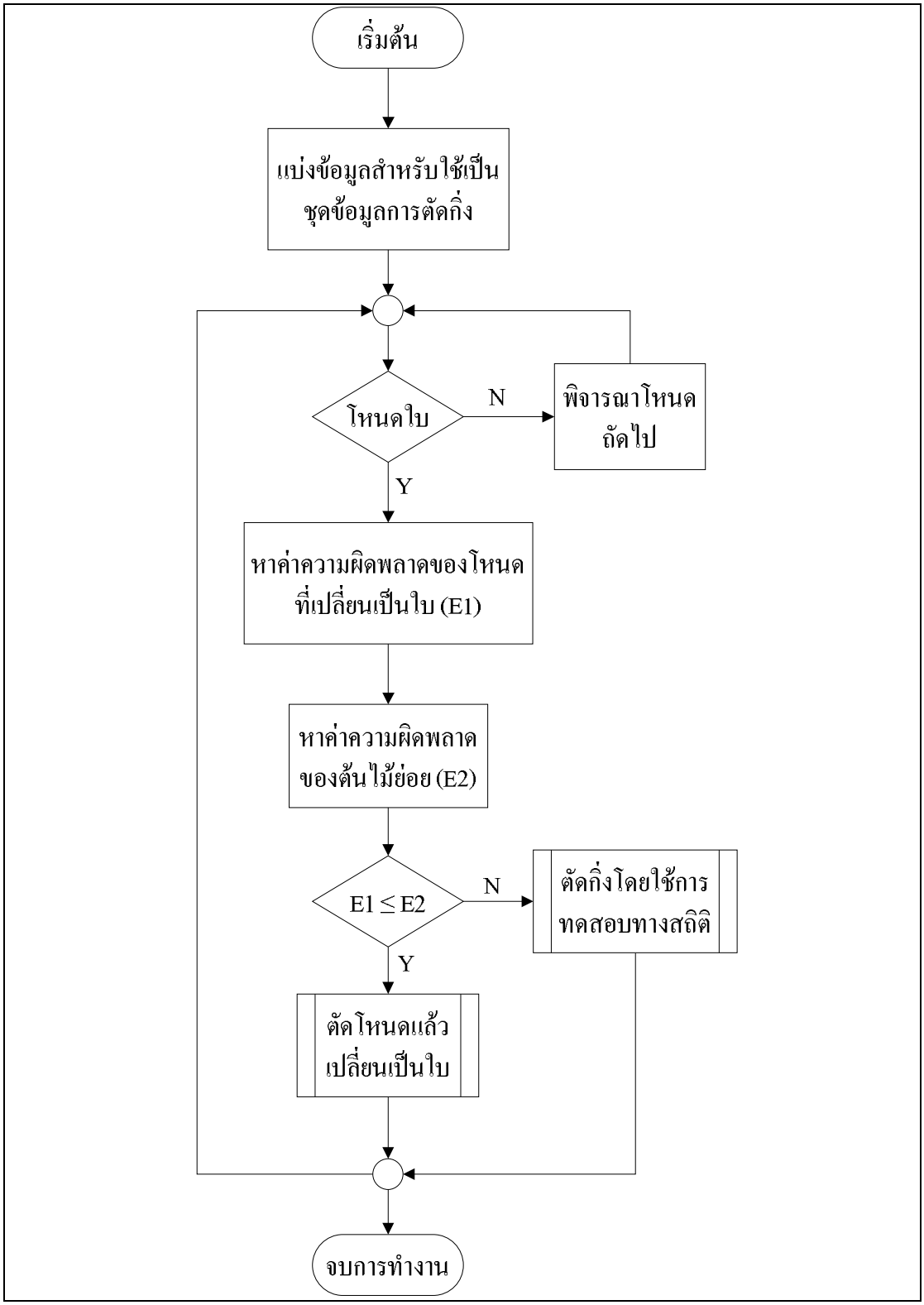
11) ถ้าค่าไคสแควร์ที่คำนวณได้ มีค่าน้อยกว่าค่าวิกฤตของการทดสอบไคสแควร์แล้วแสดงว่ากลุ่มของข้อมูลของโหนดที่พิจารณาไม่มีความสัมพันธ์กันอย่างมีนัยสำคัญ จึงสามารถดำเนินการตัดโหนดนั้นออกไปแล้วเปลี่ยนเป็นใบ โดยที่มีกลุ่มของข้อมูลเป็นกลุ่มที่มีข้อมูลสนับสนุนมากที่สุด

12) จากการเปรียบเทียบค่าองศาความเป็นอิสระ ถ้ามีค่าเท่ากับ 1 และมีจำนวนข้อมูลทั้งหมดในตารางการณ์จรน้อยกว่าหรือเท่ากับ 20 แล้วจะคำนวณค่าความน่าเป็นโดยใช้การทดสอบฟิชเชอร์

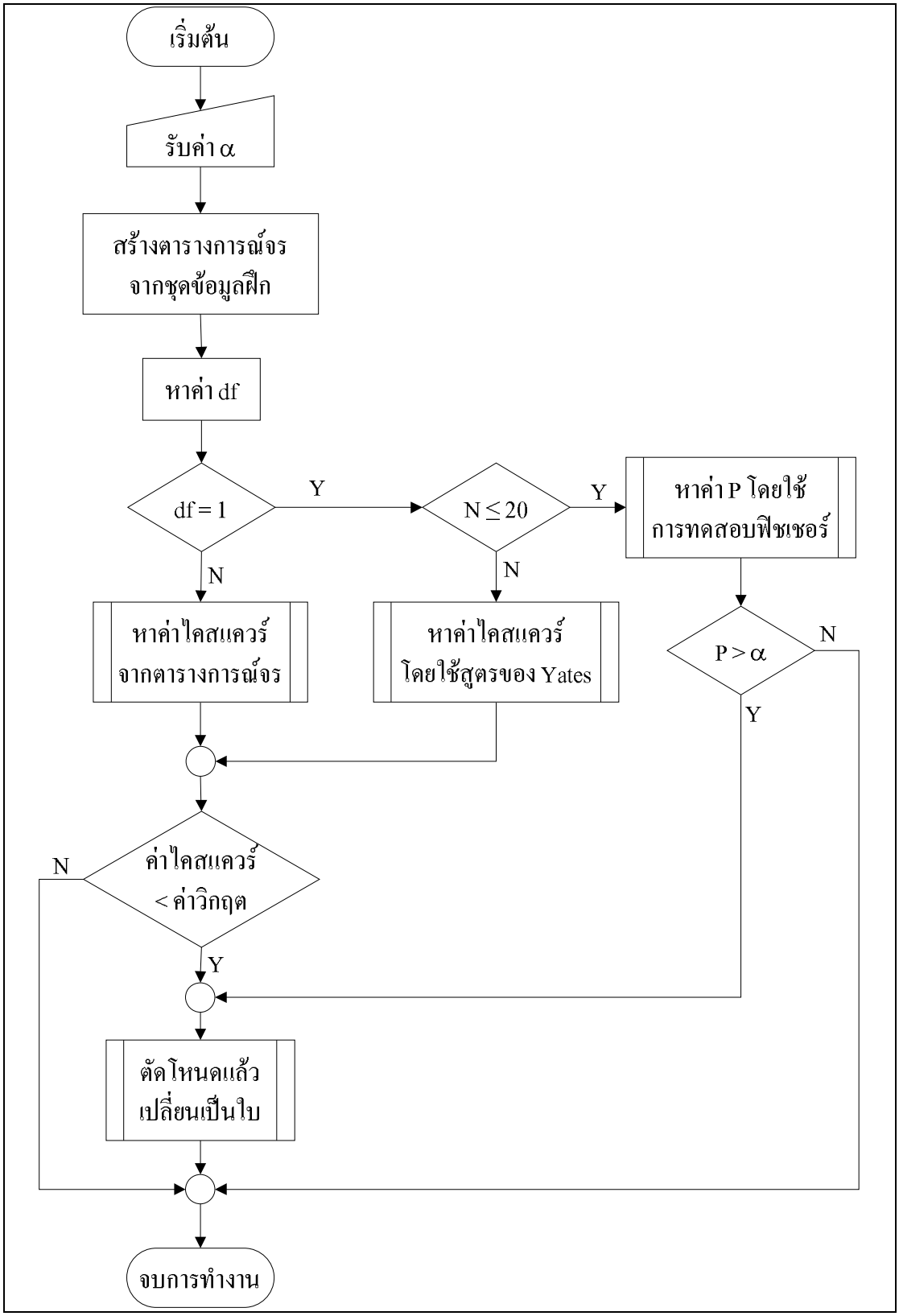
13) ถ้าค่าความน่าจะเป็นที่คำนวณได้ มีค่ามากกว่าค่าระดับความมีนัยสำคัญที่กำหนดเอาไว้ แสดงว่ากลุ่มของข้อมูลของโหนดที่พิจารณาไม่มีความสัมพันธ์กันอย่างมีนัยสำคัญ จึงสามารถดำเนินการตัดโหนดนั้นออกไปแล้วเปลี่ยนเป็นใบ โดยที่มีกลุ่มของข้อมูลเป็นกลุ่มที่มีข้อมูลสนับสนุนมากที่สุด

14) การทำงานจะวนซ้ำไปยังโหนดที่อยู่เหนือขึ้นไปจากโหนดที่พิจารณา โดยจะสิ้นสุดเมื่อโหนดที่พิจารณาเป็นโหนดรากของต้นไม้ตัดสินใจ

ในขั้นตอนการตัดกิ่งของเทคนิค REP+ ใช้การทดสอบสมมติฐานทางสถิติเพื่อตรวจสอบความสัมพันธ์กันอย่างมีนัยสำคัญ ของกลุ่มข้อมูลที่ได้จากการทำนายด้วยต้นไม้ตัดสินใจและกลุ่มข้อมูลที่เป็นจริงจากชุดข้อมูลฝึก โดยพิจารณาที่แต่ละโหนดของต้นไม้ตัดสินใจ สามารถแสดงรายละเอียดของอัลกอริทึมการทำงานของเทคนิคการตัดกิ่ง REP+ ได้ดังรูปที่ 3.11 ความซับซ้อนเชิงคำนวณของเทคนิคการตัดกิ่ง REP+ จะมีค่ามากกว่าของเทคนิคการตัดกิ่ง REP ที่เป็นแบบเชิงเส้นตามจำนวนโหนดของต้นไม้ตัดสินใจมีค่าเท่ากับ O(n) เมื่อ n เป็นจำนวนโหนดของต้นไม้ตัดสินใจ เนื่องจากเทคนิคการตัดกิ่ง REP+ จะต้องตรวจสอบความสัมพันธ์กันอย่างมีนัยสำคัญของกลุ่มข้อมูลที่แต่ละโหนดของต้นไม้ตัดสินใจด้วย ซึ่งค่ามากที่สุดที่เป็นไปได้ที่ต้องตรวจสอบมีค่าเท่ากับจำนวนโหนดของต้นไม้ตัดสินใจที่ไม่ได้ตัดกิ่ง กำหนดให้เวลาที่ใช้ในการคำนวณค่าความสัมพันธ์ทางสถิติมีค่าเท่ากับค่าคงที่ k จะได้ความซับซ้อนเชิงคำนวณของเทคนิคการตัดกิ่ง REP+ มีค่าเท่ากับ O(kn) เมื่อ n เป็นจำนวนโหนดของต้นไม้ตัดสินใจ แต่เนื่องจาก k เป็นค่าคงที่ ดังนั้นความซับซ้อนเชิงคำนวณของเทคนิคการตัดกิ่ง REP+ จึงมีค่าเท่ากับ O(n) ในหัวข้อที่ 3.4.2 และ 3.4.3 จะได้อธิบายรายละเอียดและขั้นตอนการทดสอบความเป็นอิสระต่อกันโดยใช้การทดสอบไคสแควร์ และการทดสอบความเป็นอิสระต่อกันโดยใช้การทดสอบฟิชเชอร์ตามลำดับต่อไป



รูปที่ 3.9 แสดงผังการทำงานของการตัดกิ่งด้วยเทคนิค REP+ โดยการตรวจสอบจำนวนความผิดพลาดที่โหนดของต้นไม้



รูปที่ 3.10 แสดงผังการทำงานของการตัดกิ่งด้วยเทคนิค REP+ โดยใช้การทดสอบทางสถิติ

```
DecisionTree REP+ (DecisionTree T, PruningSet S)
        for (i = 0 to S.length-1)
                classify (T, S[i])
        return prune (T)

classify (DecisionTree T, Example e)
        T.total++
        if (e.label == 1)  T.pos++     // update node counters
        if (!leaf (T))
                if (T.test (e) == 0)  classify (T.left, e)
                else classify (T.right, e)

prune (DecisionTree T)     // output classification error after pruning T
        if (leaf (T))
                if (T.label == 1)  return T.total – T.pos
                else  return T.pos
        else
                error = prune (T.left) + prune (T.right)
                if (error < min (T.pos, T.total – T.pos)
                        StatisticalTest (ContingencyTable (T), α)
                        return error
                else
                        // replace T with a leaf
                        if (T.pos > T.total – T.pos)
                                T.label = 1  return T.total – T.pos
                        else  T.label = 0  return T.pos

StatisticalTest (ContingencyTable C, Significance α)
        df = (C.row – 1) (C.col – 1)
        if (df == 1)
                if (C.total <= 20)
                        if (Fisher (C) > α)
                                replace T with a leaf
                else if (ChiSquareYates (C) < CriticalTable (df, α))
                        replace T with a leaf
        else if (ChiSquare (C) < CriticalTable (df, α))
                replace T with a leaf
```

รูปที่ 3.11 อัลกอริทึมของเทคนิคการตัดกิ่ง REP+

3.4.2 การทดสอบความเป็นอิสระต่อกันด้วยการทดสอบไคสแควร์

ในงานวิจัยนี้ได้นำค่าทางสถิติไคสแควร์ (chi-squared test) มาใช้เพื่อทดสอบสมมติฐานความเป็นอิสระต่อกันระหว่างกลุ่มของข้อมูลที่ได้จากการทำนาย กับกลุ่มของข้อมูลที่เป็นจริงจากชุดข้อมูลฝึก โดยจะทดสอบแต่ละโหนดภายในของต้นไม้ตัดสินใจ หลังจากใช้วิธีการตัดกิ่งต้นไม้ตัดสินใจ reduced-error pruning ตรวจสอบจำนวนความผิดพลาดในการทำนายกลุ่มของข้อมูลของโหนดภายในนั้นของต้นไม้ตัดสินใจแล้ว วิธี reduced-error pruning ตัดสินใจว่าไม่ตัดกิ่งของต้นไม้ย่อยนั้น เราจึงใช้การคำนวณค่าไคสแควร์เพื่อตรวจสอบความสัมพันธ์กันระหว่างกลุ่มของข้อมูลต่อไป

ในการทดสอบไคสแควร์ต้องจัดข้อมูลที่เป็นประเภทความถี่ ซึ่งเป็นจำนวนตัวอย่างจากชุดข้อมูลฝึก ณ ตำแหน่งของโหนดภายในของต้นไม้ตัดสินใจนั้น ลงในตารางการณ์จร (contingency table) โดยทางด้านคอลัมน์แทนจำนวนกลุ่มของข้อมูลที่เป็นจริงจากชุดข้อมูลฝึก ส่วนทางด้านแถวแทนจำนวนกลุ่มของข้อมูลที่ได้จากการทำนายโดยใช้ต้นไม้ตัดสินใจ แสดงโครงสร้างของตารางการณ์จรได้ดังตารางที่ 3.3

ตารางที่ 3.3 แสดงโครงสร้างของตารางการณ์จร

กลุ่มจากการทำนาย (predicted class)	กลุ่มที่เป็นจริง (actual class)		Total
	1	2	
1	o_{11}	o_{12}	R_1
2	o_{21}	o_{22}	R_2
Total	C_1	C_2	N

จากตารางที่ 3.3 เป็นรูปแบบของตารางการณ์จรขนาด 2×2 นั่นคือตารางมีจำนวนกลุ่มของข้อมูลทางด้านคอลัมน์และแถวเท่ากับ 2 ซึ่งเป็นรูปแบบเดียวกับที่ใช้ในงานวิจัยนี้ กล่าวคือตารางการณ์จรจะมีจำนวนคอลัมน์เท่ากับจำนวนแถว เนื่องจากใช้ชุดข้อมูลชุดเดียวกันในการทดสอบข้อมูล และขนาดของตารางการณ์จรจะขึ้นอยู่กับจำนวนกลุ่มของข้อมูล
จากตารางการณ์จร 2×2 ในตารางที่ 3.3 ค่าของผลรวมในแต่ละเซลล์หาได้จาก

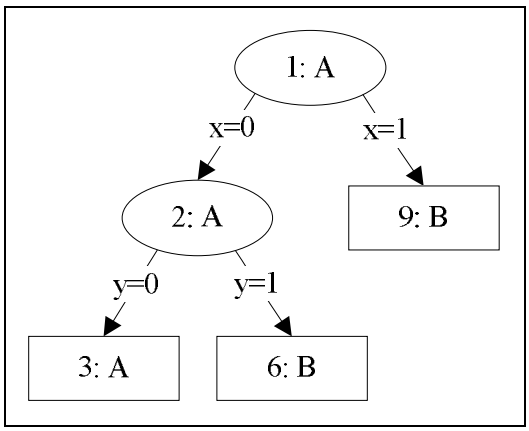
$R_1 = o_{11} + o_{12}$, $R_2 = o_{21} + o_{22}$, $C_1 = o_{11} + o_{21}$, $C_2 = o_{12} + o_{22}$ และ

$N = R_1 + R_2 = C_1 + C_2$

ตารางที่ 3.4 ต่อไปนี้แสดงตัวอย่างของชุดข้อมูลฝึกที่ระบุกลุ่มที่เป็นจริงเอาไว้ พร้อมกับกลุ่มที่ได้จากการทำนายโดยใช้ต้นไม้ตัดสินใจในรูปที่ 3.12

ตารางที่ 3.4 ตัวอย่างของชุดข้อมูลฝึกที่ระบุกลุ่มที่ได้จากการทำนาย

x	y	z	actual class	predicted class
0	0	1	A	A
0	1	1	B	B
1	1	0	B	B
1	0	0	B	B
1	1	1	A	B



รูปที่ 3.12 ตัวอย่างต้นไม้ตัดสินใจที่ใช้สำหรับทดสอบทางสถิติ

ตารางที่ 3.5 ตารางการณ์จรที่มีกลุ่มของข้อมูล 2 กลุ่ม

predicted class	actual class		Total
	A	B	
A	1	0	1
B	1	3	4
Total	2	3	5

จากตัวอย่างชุดข้อมูลฝึกในตารางที่ 3.4 สามารถนำข้อมูลมาจัดลงในตารางการณ์จรได้ดังในตารางที่ 3.5 ตารางการณ์จรที่ได้จะเห็นว่ามีลักษณะคล้ายกับ confusion matrix ที่ใช้ในอัลกอริทึมการจัดกลุ่มของข้อมูล ซึ่งช่วยในการตรวจสอบจำนวนตัวอย่างจากชุดข้อมูลฝึกที่สามารถจำแนกกลุ่มของข้อมูลได้อย่างถูกต้องเมื่อทดสอบด้วยต้นไม้ย่อยทำได้ง่ายขึ้น โดยจำนวนตัวอย่างที่จำแนกกลุ่มของข้อมูลได้อย่างถูกต้อง จะเท่ากับผลรวมของสมาชิกในแนวเส้นทแยงมุมของตาราง คือ 1+3 = 4 นั่นเอง

จากตารางการณ์จรสามารถทดสอบด้วยไคสแควร์ เพื่อทดสอบความเป็นอิสระต่อกันระหว่างกลุ่มของข้อมูลที่ได้จากการทำนายกับกลุ่มของข้อมูลจากชุดข้อมูลฝึก โดยการทดสอบไคสแควร์มีสูตรเป็นดังนี้

$$\chi^2 = \sum_{i=1}^{r}\sum_{j=1}^{c}\frac{(o_{ij} - e_{ij})^2}{e_{ij}}$$ หรือเขียนในอีกรูปแบบหนึ่งได้เป็น

$$\chi^2 = \sum_{i=1}^{r}\sum_{j=1}^{c}\frac{o_{ij}^{\ 2}}{e_{ij}} - N$$ เมื่อกำหนดให้

o_{ij} แทนความถี่ที่สังเกตได้แถวที่ i คอลัมน์ที่ j
e_{ij} แทนความถี่ที่คาดหวังแถวที่ i คอลัมน์ที่ j
N แทนจำนวนความถี่ที่สังเกตได้ทั้งหมด
โดยที่ e_{ij} หาได้จากสูตรดังนี้ $e_{ij} = \frac{R_i C_j}{N}$

เมื่อ $R_i = \sum_{j=1}^{c} o_{ij}$ และ $C_j = \sum_{i=1}^{r} o_{ij}$

จากตารางการณ์จรในตารางที่ 3.5 สามารถหาความถี่ที่คาดหวังได้เป็น $e_{11} = 2/5$, $e_{12} = 3/5$, $e_{21} = 8/5$ และ $e_{22} = 12/5$ ดังนั้นค่าไคสแควร์ของตารางการณ์จรนี้สามารถคำนวณได้ดังนี้

$$\chi^2 = \frac{(1-0.4)^2}{0.4} + \frac{(0-0.6)^2}{0.6} + \frac{(1-1.6)^2}{1.6} + \frac{(3-2.4)^2}{2.4} = 1.875$$

ถ้าค่าความถี่ที่สังเกตได้มีค่าใกล้เคียงกับความถี่ที่คาดหวัง ความแตกต่าง (o_{ij}-e_{ij}) จะมีค่าน้อย ค่าของไคสแควร์ก็จะมีค่าน้อยด้วย ถ้าค่าไคสแควร์มีค่าน้อย เราก็ไม่สามารถปฏิเสธ H_0 ที่ว่า

กลุ่มของข้อมูลที่ได้จากการทำนายกับกลุ่มของข้อมูลจากชุดข้อมูลฝึกมีความเป็นอิสระต่อกัน แต่ถ้าความแตกต่าง (o_{ij}-e_{ij}) มีค่ามาก ค่าไคสแควร์ก็จะมีค่ามากด้วย ซึ่งจะทำให้เราสามารถปฏิเสธ H_0 และยอมรับว่ากลุ่มของข้อมูลทั้งสองมีความสัมพันธ์กัน

ค่าไคสแควร์ที่คำนวณได้ ต้องนำมาเปรียบเทียบกับค่าวิกฤตของไคสแควร์จากตารางในภาคผนวก ค ซึ่งจะมีค่าแตกต่างกันตามค่าระดับความเป็นอิสระ (degree of freedom, df) และระดับความมีนัยสำคัญ (α) โดยค่า df สำหรับข้อมูลที่จัดลงในตารางการณ์จรขนาด r×c หาได้จากสูตรดังนี้

$df = (r-1)(c-1)$ เมื่อ r แทนจำนวนแถวของตาราง และ c แทนจำนวนคอลัมน์ของตาราง

เราสามารถปฏิเสธ H_0 เพื่อสรุปว่ากลุ่มของข้อมูลที่ได้จากการทำนายของต้นไม้ตัดสินใจมีความสัมพันธ์กันอย่างมีนัยสำคัญกับกลุ่มของข้อมูลจากชุดข้อมูลฝึกได้ นั่นคือจะยังคงโหนดของต้นไม้ตัดสินใจนั้นเอาไว้แทนการตัดโหนดนั้นของต้นไม้ออกไป ถ้าค่าไคสแควร์ที่คำนวณได้มีค่ามากกว่าหรือเท่ากับค่าวิกฤตของการทดสอบไคสแควร์ จากค่าระดับความเป็นอิสระและกำหนดระดับความมีนัยสำคัญที่ต้องการ โดยในงานวิจัยนี้ได้ทดสอบข้อมูลโดยใช้ระดับความมีนัยสำคัญเป็น 0.01, 0.025, 0.05, 0.1, 0.25 และ 0.5 ตามลำดับ

จากตัวอย่างตารางการณ์จรในตารางที่ 3.5 เป็นตารางการณ์จรขนาด 2×2 จะได้ค่าของ df = (2-1) (2-1) = 1 และกำหนดระดับความมีนัยสำคัญเป็น 0.05 จะได้ค่าวิกฤตของการทดสอบไคสแควร์จากตารางในภาคผนวก ค เท่ากับ 3.841 เมื่อเปรียบเทียบกับค่าไคสแควร์ที่คำนวณได้เท่ากับ 1.875 จะได้ว่าค่าไคสแควร์ที่คำนวณได้มีค่าน้อยกว่าค่าวิกฤตของไคสแควร์ จึงสามารถสรุปว่ากลุ่มของข้อมูลที่ได้จากการทำนายของต้นไม้ตัดสินใจ มีความเป็นอิสระกับกลุ่มของข้อมูลจากชุดข้อมูลฝึก ดังนั้นจึงตัดโหนดของต้นไม้ตัดสินใจนั้นออกไป

ในกรณีที่ข้อมูลจัดเป็นตารางการณ์จรขนาด 2×2 คือมีจำนวนแถวและจำนวนคอลัมน์เท่ากันและเท่ากับสอง Siegel and Castellan (1988) ได้เสนอแนะเกี่ยวกับการใช้สถิติดังนี้

1) ถ้า $N \leq 20$ ใช้การทดสอบฟิชเชอร์
2) ถ้า $20 < N \leq 40$ ให้พิจารณาเป็น 2 กรณีคือ
   (1) ถ้าความถี่ที่คาดหวังทุกตัวมากกว่าหรือเท่ากับ 5 ให้ใช้สูตร Yates' correction for continuity ดังนี้

$$\chi^2 = \sum_{i=1}^{r}\sum_{j=1}^{c}\frac{(|o_{ij}-e_{ij}|-0.5)^2}{e_{ij}}$$

(2) ถ้าความถี่ที่คาดหวังตัวที่น้อยที่สุดมีค่าน้อยกว่า 5 ใช้การทดสอบฟิชเชอร์

3) ถ้า N > 40 ใช้การทดสอบไคสแควร์โดยใช้สูตร Yates' correction for continuity

3.4.3 การทดสอบความเป็นอิสระต่อกันด้วยการทดสอบฟิชเชอร์

การทดสอบทางสถิติโดยใช้ไคสแควร์ มีข้อจำกัดในเรื่องจำนวนความถี่ของข้อมูลเมื่อจัดลงในตารางการณ์จรที่ต้องมีจำนวนความถี่มากพอ จึงจะสามารถคำนวณค่าไคสแควร์ได้อย่างถูกต้องและสามารถใช้ทดสอบความเป็นอิสระได้อย่างมีประสิทธิภาพ แต่จะเกิดปัญหาขึ้นเมื่อนำไปใช้กับข้อมูลที่มีจำนวนตัวอย่างน้อย โดยเฉพาะเมื่อนำมาใช้กับต้นไม้ตัดสินใจที่แต่ละโหนดของต้นไม้จะแบ่งจำนวนตัวอย่างที่มีขนาดลดลงเรื่อย ๆ จนกว่าจะสามารถแบ่งแยกกลุ่มของข้อมูลได้อย่างถูกต้อง ซึ่งที่ตำแหน่งใบของต้นไม้จะมีจำนวนตัวอย่างน้อย จึงไม่เหมาะที่จะใช้การทดสอบไคสแควร์จำเป็นต้องใช้การทดสอบทางสถิติวิธีอื่น เพื่อทดสอบความเป็นอิสระต่อกันระหว่างกลุ่มของข้อมูลที่ได้จากการทำนายกับกลุ่มของข้อมูลที่เป็นจริงในชุดข้อมูลฝึก การทดสอบทางสถิติวิธีหนึ่งที่สามารถนำมาใช้กับข้อมูลขนาดเล็กได้ดีคือ การทดสอบฟิชเชอร์ (Fisher exact test) โดยในงานวิจัยนี้ได้กำหนดให้ใช้การทดสอบฟิชเชอร์ เมื่อจำนวนความถี่ของข้อมูลมีค่าน้อยกว่าหรือเท่ากับ 20 (Siegel and Castellan, 1988)

การทดสอบฟิชเชอร์ใช้ทดสอบความเป็นอิสระต่อกันของกลุ่มข้อมูล จำนวน 2 กลุ่ม และสามารถจัดข้อมูลให้อยู่ในรูปตารางการณ์จรขนาด 2×2 ได้ดังตารางที่ 3.6

ตารางที่ 3.6 โครงสร้างตารางการณ์จรที่มีกลุ่มของข้อมูล 2 กลุ่ม

predicted class	actual class		Total
	1	2	
1	A	B	A+B
2	C	D	C+D
Total	A+C	B+D	N

กำหนดให้ A, B, C, D แทนจำนวนความถี่ที่ได้จากการสังเกต และ N แทนจำนวนความถี่ที่ได้จากสังเกตทั้งหมด เราสามารถหาค่าความน่าจะเป็น (exact probability, p) ที่จะเกิดความถี่ที่ได้จากการสังเกต และมีค่าใด ๆ ตกอยู่ในตารางการณ์จรนี้ได้ โดยการหาจากฟังก์ชันการแจกแจงแบบไฮเปอร์จิออมิตริก (hypergeometric distribution) ได้โดยใช้สูตรดังนี้

$$p = \frac{\binom{A+C}{A}\binom{B+D}{B}}{\binom{N}{A+B}} = \frac{\left[\frac{(A+C)!}{A!C!}\right]\left[\frac{(B+D)!}{B!D!}\right]}{\frac{N!}{(A+B)!(C+D)!}}$$

และจะได้ว่า $p = \frac{(A+B)!(C+D)!(A+C)!(B+D)!}{N!\,A!B!C!D!}$

โดยค่าความน่าจะเป็นนี้ เป็นค่าอัตราส่วนของผลคูณของแฟคทอเรียลของผลรวมทางแถวและทางคอลัมน์ต่อความถี่ของแต่ละเซลล์และความถี่ทั้งหมด

จากตัวอย่างข้อมูลในตารางที่ 3.5 แสดงตารางการณ์จรที่มีกลุ่มของข้อมูล 2 กลุ่ม เมื่อแทนค่าตัวแปรต่าง ๆ ด้วย A=1, B=0, C=1, D=3 และ N=5 สามารถคำนวณค่าความน่าจะเป็นได้ดังนี้

$$p = \frac{(1+0)!(1+3)!(1+1)!(0+3)!}{5!1!0!1!3!} = 0.4$$

ค่าความน่าจะเป็นที่คำนวณได้ ต้องนำมาเปรียบเทียบกับค่าระดับความมีนัยสำคัญ (α) เราสามารถปฏิเสธ H_0 เพื่อสรุปว่ากลุ่มของข้อมูลที่ได้จากการทำนายของต้นไม้ตัดสินใจมีความสัมพันธ์กันอย่างมีนัยสำคัญกับกลุ่มของข้อมูลจากชุดข้อมูลฝึกได้ นั่นคือจะยังคงโหนดของต้นไม้ตัดสินใจนั้นเอาไว้แทนการตัดโหนดนั้นของต้นไม้ออกไป ถ้าค่าความน่าจะเป็นที่คำนวณได้มีค่าน้อยกว่าหรือเท่ากับค่าระดับความมีนัยสำคัญที่กำหนดไว้ในการทดสอบข้อมูล โดยในงานวิจัยนี้ได้ทดสอบข้อมูลโดยใช้ค่าระดับความมีนัยสำคัญเป็น 0.01, 0.025, 0.05, 0.1, 0.25 และ 0.5

จากการคำนวณค่าความน่าจะเป็นจากตัวอย่างตารางการณ์จรในตารางที่ 3.5 และกำหนดค่าระดับความมีนัยสำคัญเป็น 0.05 เมื่อเปรียบเทียบกับค่าความน่าจะเป็นที่คำนวณได้เท่ากับ 0.4 จะได้ว่าค่าความน่าจะเป็นที่คำนวณได้มีค่ามากกว่าค่าระดับความมีนัยสำคัญที่กำหนดไว้ จึงสามารถสรุปว่ากลุ่มของข้อมูลที่ได้จากการทำนายของต้นไม้ตัดสินใจ มีความเป็นอิสระกับกลุ่มของข้อมูลจากชุดข้อมูลฝึก ดังนั้นจึงตัดโหนดของต้นไม้ตัดสินใจนั้นออกไป

3.5 การทดสอบเปรียบเทียบวิธีการตัดกิ่งต้นไม้ตัดสินใจ

จากตารางที่ 3.1 แสดงรายละเอียดของชุดข้อมูลที่ใช้ในงานวิจัย จะเห็นว่าชุดข้อมูลที่คัดเลือกสามารถแบ่งออกได้เป็น 2 ประเภทคือ

1) ชุดข้อมูลที่แบ่งข้อมูลฝึกและข้อมูลทดสอบไว้แล้ว (แสดงจำนวนตัวอย่างของข้อมูลทดสอบไว้ในวงเล็บ) สำหรับข้อมูลชุดนี้การทดลองจะใช้ข้อมูลฝึกสร้างต้นไม้ตัดสินใจและทดสอบความถูกต้องด้วยข้อมูลทดสอบ
2) ชุดข้อมูลที่ไม่มีการแบ่งข้อมูลฝึกและข้อมูลทดสอบ ในชุดข้อมูลชนิดนี้จะใช้วิธี cross-validation โดยแบ่งชุดข้อมูลออกเป็น 10 โฟลด์ (fold) เท่า ๆ กัน ในแต่ละรอบจะกำหนดให้หนึ่งโฟลด์ใช้เป็นข้อมูลทดสอบ และโฟลด์ที่เหลือจะใช้เป็นข้อมูลฝึกเพื่อสร้างต้นไม้ตัดสินใจ วนซ้ำจนชุดข้อมูลทุกโฟลด์ถูกนำมาใช้เป็นข้อมูลทดสอบ ผลลัพธ์ที่ได้จากวิธีการนี้เป็นค่าเฉลี่ยของผลลัพธ์ที่ได้จากการทำงานจำนวน 10 ครั้ง

วิธีการตัดกิ่งต้นไม้ตัดสินใจที่นำมาเปรียบเทียบประสิทธิภาพกับเทคนิคการตัดกิ่งต้นไม้ตัดสินใจวิธี REP+ คือ วิธีการตัดกิ่งแบบความผิดพลาดลดลง (reduced-error pruning) และวิธีการตัดกิ่งโดยใช้ค่าความผิดพลาด (error-based pruning) การทดสอบวิธีการตัดกิ่งต้นไม้ตัดสินใจที่พัฒนาขึ้นจะใช้ระดับความมีนัยสำคัญ (α) จำนวนทั้งสิ้น 6 ค่าคือ 0.01, 0.025, 0.05, 0.1, 0.25 และ 0.5 โดยผลการทดสอบข้อมูลด้วยเทคนิคการตัดกิ่งต้นไม้ตัดสินใจวิธี REP+ ที่นำมาเปรียบเทียบประสิทธิภาพกับเทคนิคการตัดกิ่งต้นไม้ตัดสินใจด้วยวิธี EBP, REP และต้นไม้ตัดสินใจที่ไม่ได้ตัดกิ่งจะเป็นค่าเฉลี่ยที่ได้เมื่อใช้ค่าระดับนัยสำคัญต่าง ๆ ทดสอบกับข้อมูล โดยเกณฑ์ที่ใช้ศึกษาเพื่อเปรียบเทียบประสิทธิภาพของวิธีการตัดกิ่งต้นไม้ตัดสินใจประกอบด้วย เวลาที่ใช้ในการสร้างต้นไม้ตัดสินใจ ขนาดของต้นไม้ตัดสินใจที่ได้หลังจากการตัดกิ่งแล้ว และความแม่นตรงในการจำแนกกลุ่มของข้อมูล

บทที่ 4
ผลการวิเคราะห์ข้อมูลและการอภิปรายผล

การวิเคราะห์เปรียบเทียบประสิทธิภาพของเทคนิคการตัดกิ่งต้นไม้ตัดสินใจในงานวิจัยนี้ ใช้เทคนิคการตัดกิ่งที่สำคัญ 2 วิธีได้แก่ วิธีการตัดกิ่งต้นไม้ตัดสินใจแบบความผิดผลาดลดลง (reduced-error pruning, REP) และวิธีการตัดกิ่งต้นไม้ตัดสินใจโดยใช้ค่าความผิดพลาด (error-based pruning, EBP) เปรียบเทียบกับเทคนิคการตัดกิ่งต้นไม้ตัดสินใจที่พัฒนาขึ้นในงานวิจัยนี้ที่ให้ชื่อว่า extended reduced-error pruning (REP+) ทดสอบกับข้อมูลทางด้านวิทยาศาสตร์ที่เน้นทางด้านการแพทย์และโครงสร้างทางพันธุกรรมของมนุษย์ โดยประมวลผลโปรแกรมด้วยเครื่องคอมพิวเตอร์ PC รุ่น Pentium 4 ความเร็ว 3.20 GHz หน่วยความจำหลัก 512 MB ทำงานบนระบบปฏิบัติการ Windows XP Professional Edition Service Pack 2 ผลการทดสอบค่าระดับความมีนัยสำคัญเมื่อใช้เทคนิคการตัดกิ่งต้นไม้ตัดสินใจ REP+ ปรากฏรายละเอียดในหัวข้อ 4.1 ในหัวข้อ 4.2 เป็นการเปรียบเทียบประสิทธิภาพของเทคนิคการตัดกิ่งต้นไม้ตัดสินใจด้วยวิธี REP, EBP และเทคนิคที่พัฒนาขึ้น (REP+) พร้อมทั้งแสดงผลการทดสอบกับต้นไม้ตัดสินใจที่ไม่ได้ตัดกิ่งด้วยนำมาวิเคราะห์เปรียบเทียบ โดยเกณฑ์ที่ใช้ศึกษาเพื่อเปรียบเทียบประสิทธิภาพของวิธีการตัดกิ่งต้นไม้ตัดสินใจประกอบด้วย เวลาที่ใช้ในการสร้างต้นไม้ตัดสินใจ ขนาดของต้นไม้ตัดสินใจที่ได้หลังจากการตัดกิ่งแล้ว และความแม่นตรงในการจำแนกกลุ่มของข้อมูล

4.1 การทดสอบค่าระดับความมีนัยสำคัญของเทคนิค REP+

ตารางที่ 4.1 และ 4.2 ต่อไปนี้แสดงประสิทธิภาพของเทคนิคการตัดกิ่งต้นไม้ตัดสินใจที่พัฒนาขึ้นในงานวิจัยนี้ชื่อว่า REP+ โดยเปรียบเทียบขนาดของต้นไม้ตัดสินใจหลังจากการตัดกิ่งแล้วและความแม่นตรงในการจำแนกกลุ่มของข้อมูล เมื่อทดสอบข้อมูลแต่ละชุดด้วยระดับความมีนัยสำคัญ (α) จำนวนทั้งสิ้น 6 ค่าคือ 0.01, 0.025, 0.05, 0.1, 0.25 และ 0.5 ตามลำดับ

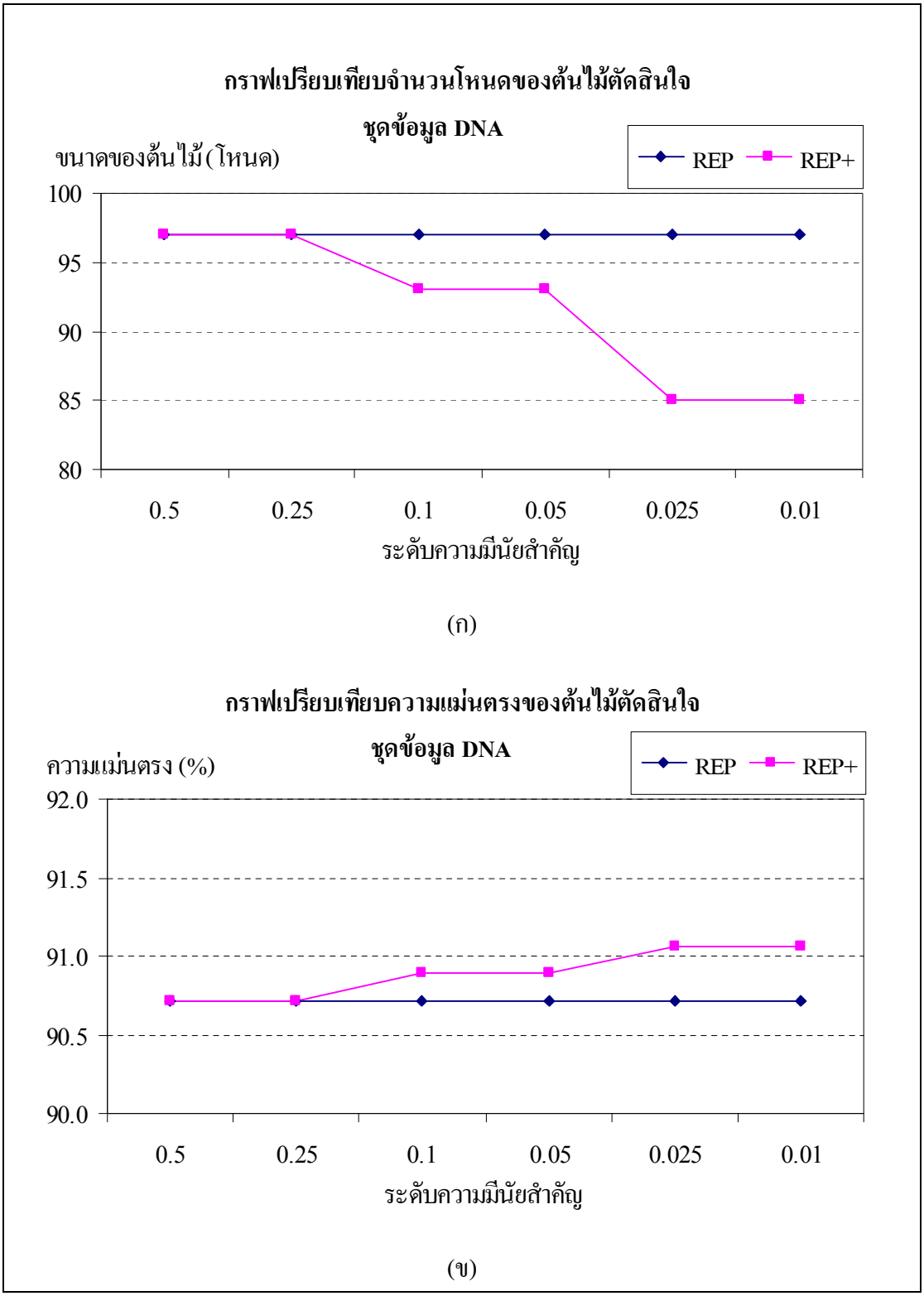
ตารางที่ 4.1 ขนาดของต้นไม้ตัดสินใจที่ได้จากเทคนิค REP+ เมื่อใช้ระดับความมีนัยสำคัญต่าง ๆ

ชื่อชุดข้อมูล	ขนาดของต้นไม้ตัดสินใจ (โหนด)					
	0.01	0.025	0.05	0.1	0.25	0.5
1. Allbp	15	35	35	35	41	47
2. Allhyper	7	7	7	7	7	7
3. Allhypo	17	17	17	17	17	21
4. Allrep	7	7	7	7	9	15
5. Breast-cancer	3	3	6	6	6	22
6. Breast-w	3	3	3	3	3	3
7. Dermatology	21	21	21	21	25	25
8. Diabetes	15	15	15	15	15	15
9. DNA	85	85	93	93	97	97
10. Echocardiogram	3	3	3	3	3	3
11. Heart-disease	5	5	5	5	5	12
12. Heart-h	8	8	8	8	8	8
13. Heart-Statlog	7	7	11	11	13	13
14. Hepatitis	1	7	7	7	9	9
15. Hypothyroid	9	9	9	9	9	9
16. Liver-disorders	5	17	17	17	17	17
17. Lung cancer	1	1	3	3	3	3
18. Promoters	5	5	5	5	5	5
19. Sick-euthyroid	7	7	7	13	13	17
20. Splice-junction	136	146	151	160	169	169
21. Thyroid	5	5	5	5	5	5

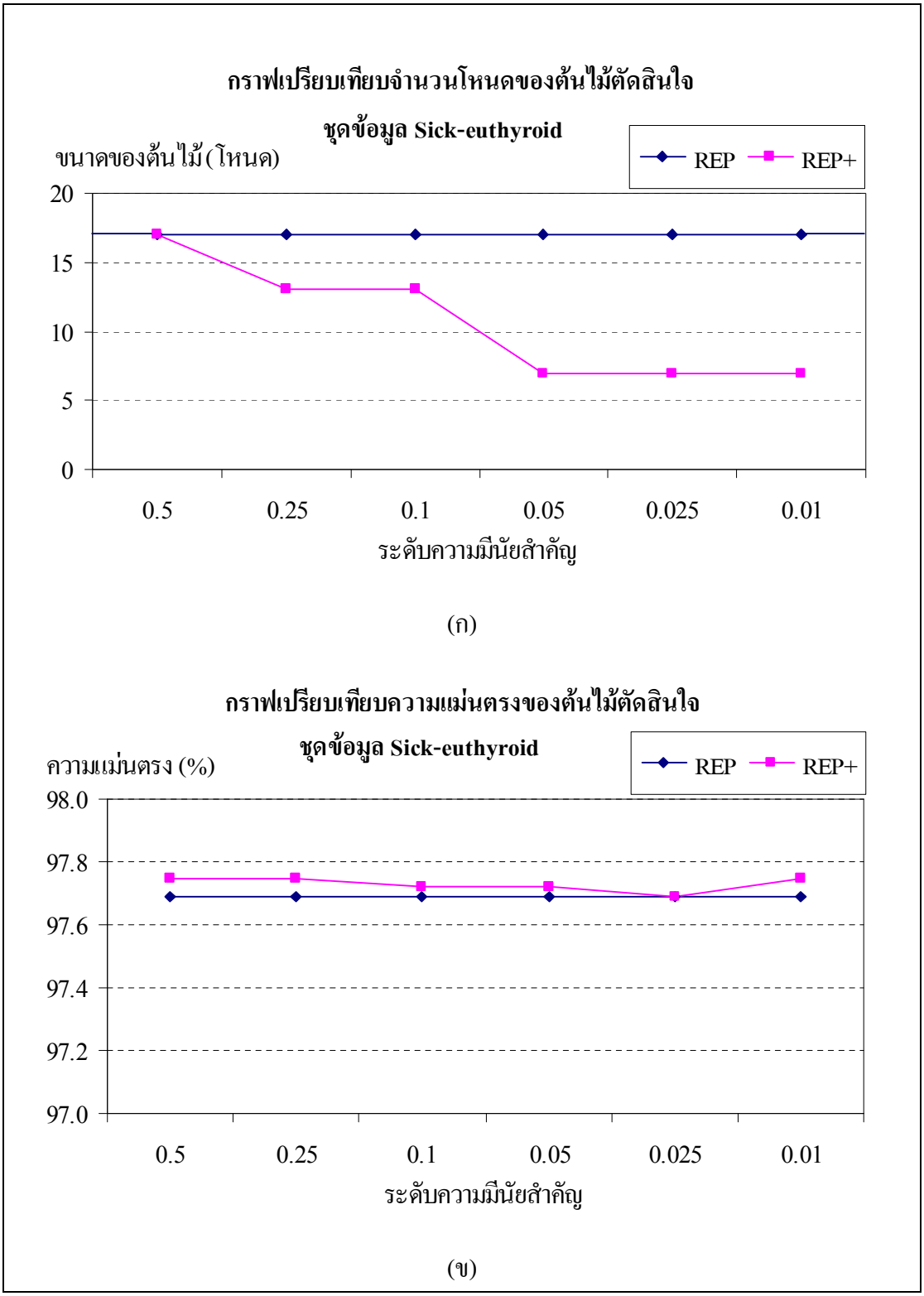
ตารางที่ 4.2 ความแม่นตรงในการจำแนกกลุ่มของข้อมูลด้วยเทคนิค REP+ เมื่อใช้ระดับความมีนัยสำคัญต่าง ๆ

ชื่อชุดข้อมูล	ความแม่นตรง (%)					
	0.01	0.025	0.05	0.1	0.25	0.5
1. Allbp	96.60	96.30	96.30	96.30	96.71	**96.81**
2. Allhyper	98.15	98.15	98.15	98.15	98.15	98.15
3. Allhypo	99.79	99.79	99.79	99.79	99.79	99.79
4. Allrep	98.05	98.05	98.05	98.05	98.46	**98.97**
5. Breast-cancer	69.23	70.63	71.33	**72.73**	72.03	72.03
6. Breast-w	91.99	91.99	92.56	**92.70**	**92.70**	**92.70**
7. Dermatology	86.34	87.70	87.98	87.98	88.80	**91.26**
8. Diabetes	**75.00**	74.87	74.87	74.87	**75.00**	**75.00**
9. DNA	**91.06**	**91.06**	90.89	90.89	90.72	90.72
10. Echocardiogram	91.60	91.60	91.60	91.60	91.60	91.60
11. Heart-disease	54.67	54.67	54.67	54.89	55.00	**55.11**
12. Heart-h	**80.61**	**80.61**	80.27	80.27	79.59	79.25
13. Heart-Statlog	74.81	77.78	77.41	77.78	78.52	**79.26**
14. Hepatitis	80.64	80.64	**81.94**	**81.94**	**81.94**	**81.94**
15. Hypothyroid	99.27	99.27	99.27	99.27	99.27	99.27
16. Liver-disorders	63.77	63.77	64.93	66.96	66.67	**68.12**
17. Lung cancer	40.63	43.75	43.75	43.75	40.63	**46.88**
18. Promoters	78.30	78.30	78.30	78.30	78.30	78.30
19. Sick-euthyroid	**97.75**	97.69	97.72	97.72	**97.75**	**97.75**
20. Splice-junction	92.63	93.10	93.29	93.70	**93.76**	**93.76**
21. Thyroid	88.37	88.37	**89.30**	88.37	88.37	88.37

หมายเหตุ ตัวเลขที่แสดงเป็นตัวหนาแสดงค่าความแม่นตรงในการจำแนกกลุ่มของข้อมูลที่สูงที่สุดเมื่อใช้ระดับความมีนัยสำคัญต่าง ๆ



รูปที่ 4.1 กราฟเปรียบเทียบประสิทธิภาพของต้นไม้ตัดสินใจจากชุดข้อมูล DNA ที่ระดับความมีนัยสำคัญต่าง ๆ



รูปที่ 4.2 กราฟเปรียบเทียบประสิทธิภาพของต้นไม้ตัดสินใจจากชุดข้อมูล Sick-euthyroid ที่ระดับความมีนัยสำคัญต่าง ๆ

จากข้อมูลในตารางที่ 4.1 และ 4.2 เมื่อนำผลการทดสอบข้อมูลที่ได้มาสร้างเป็นกราฟแสดงประสิทธิภาพของต้นไม้ตัดสินใจ เมื่อใช้เทคนิคการตัดกิ่ง REP+ ที่ระดับความมีนัยสำคัญ (α) ต่าง ๆ โดยแสดงขนาดของต้นไม้ตัดสินใจและความแม่นตรงในการจำแนกกลุ่มของข้อมูลจากการทดสอบด้วยชุดข้อมูล DNA และ Sick-euthyroid สามารถแสดงผลได้ดังรูปที่ 4.1 และ 4.2

จากกราฟในรูปที่ 4.1 และ 4.2 แสดงประสิทธิภาพของต้นไม้ตัดสินใจที่สร้างขึ้นโดยใช้ชุดข้อมูล DNA และ Sick-euthyroid โดยแสดงขนาดของต้นไม้ตัดสินใจที่ได้หลังจากการตัดกิ่ง และความแม่นตรงในการจำแนกกลุ่มของข้อมูลที่ได้จากเทคนิคการตัดกิ่ง REP+ เมื่อทดสอบข้อมูลด้วยระดับความมีนัยสำคัญจำนวนทั้งหมด 6 ค่าคือ 0.01, 0.025, 0.05, 0.1, 0.25 และ 0.5 ตามลำดับ จะเห็นได้ว่าเมื่อทดสอบกับชุดข้อมูล DNA และ Sick-euthyroid ด้วยระดับความมีนัยสำคัญที่ลดลง จะทำให้ได้ต้นไม้ตัดสินใจที่มีขนาดเล็กลงด้วย โดยเมื่อใช้ระดับความมีนัยสำคัญเท่ากับ 0.01 จะได้ต้นไม้ที่มีขนาดเล็กที่สุด และเมื่อเปรียบเทียบกับขนาดของต้นไม้ตัดสินใจที่ได้จากเทคนิคการตัดกิ่ง REP จะเห็นได้ว่าต้นไม้ตัดสินใจที่ได้จากเทคนิคการตัดกิ่ง REP+ จะมีจำนวนโหนดของต้นไม้ไม่มากไปกว่าของต้นไม้ตัดสินใจที่ได้จากเทคนิคการตัดกิ่ง REP ในทุก ๆ ระดับความมีนัยสำคัญที่ใช้ทดสอบกับชุดข้อมูล และเมื่อเปรียบเทียบความแม่นตรงในการจำแนกกลุ่มของข้อมูลโดยใช้ระดับความมีนัยสำคัญต่าง ๆ จะเห็นได้ว่าความแม่นตรงไม่ได้ลดลงอย่างมีนัยสำคัญจากการเปลี่ยนแปลงค่าระดับความมีนัยสำคัญ และยังคงให้ความแม่นตรงที่เทียบเท่าหรือสูงกว่าเมื่อเปรียบเทียบกับความแม่นตรงที่ได้จากเทคนิคการตัดกิ่ง REP

4.2 การเปรียบเทียบประสิทธิภาพของเทคนิค EBP, REP และ REP+

ในหัวข้อนี้เป็นการเปรียบเทียบประสิทธิภาพ ของเทคนิคการตัดกิ่งต้นไม้ตัดสินใจด้วยวิธี EBP, REP และ REP+ พร้อมทั้งแสดงผลการทดสอบกับต้นไม้ตัดสินใจที่ไม่ได้ตัดกิ่งด้วย โดยเกณฑ์ที่ใช้ศึกษาเพื่อเปรียบเทียบประสิทธิภาพของวิธีการตัดกิ่งต้นไม้ตัดสินใจประกอบด้วย เวลาที่ใช้ในการสร้างต้นไม้ตัดสินใจ ขนาดของต้นไม้ตัดสินใจที่ได้หลังจากการตัดกิ่งแล้ว และความแม่นตรงในการจำแนกกลุ่มของข้อมูล โดยผลการทดสอบข้อมูลด้วยเทคนิคการตัดกิ่งต้นไม้ตัดสินใจวิธี REP+ ที่นำมาเปรียบเทียบประสิทธิภาพกับเทคนิคการตัดกิ่งต้นไม้ตัดสินใจด้วยวิธี EBP, REP และต้นไม้ตัดสินใจที่ไม่ได้ตัดกิ่งจะเป็นค่าเฉลี่ยที่ได้เมื่อใช้ค่าระดับนัยสำคัญต่าง ๆ ทดสอบกับข้อมูล

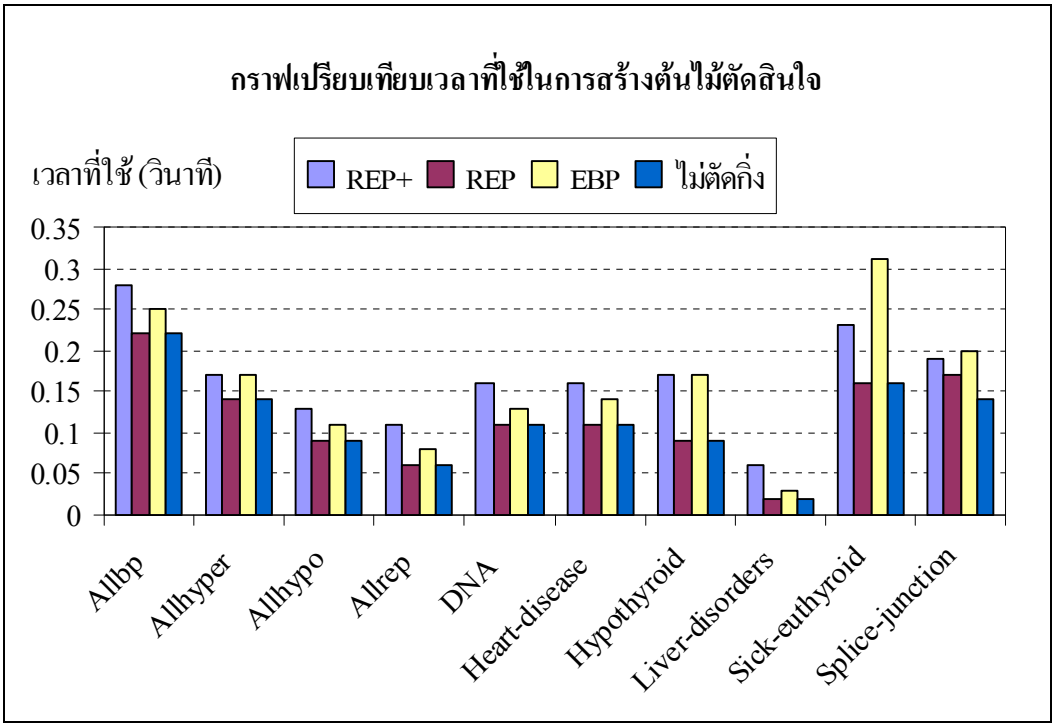
4.2.1 การเปรียบเทียบเวลาที่ใช้ในการสร้างต้นไม้ตัดสินใจ

ตารางที่ 4.3 ต่อไปนี้แสดงประสิทธิภาพของเทคนิคการตัดกิ่งต้นไม้ตัดสินใจด้วยวิธี EBP, REP และ REP+ และประสิทธิภาพของต้นไม้ตัดสินใจที่ไม่ได้ตัดกิ่ง โดยเปรียบเทียบเวลาที่ใช้ในการสร้างต้นไม้ตัดสินใจ สามารถแสดงผลการทดสอบข้อมูลได้ดังนี้

ตารางที่ 4.3 เวลาที่ใช้ในการสร้างต้นไม้ตัดสินใจของเทคนิค EBP, REP และ REP+

ชื่อชุดข้อมูล	เวลาที่ใช้ (วินาที)			
	ไม่ตัดกิ่ง	EBP	REP	REP+
1. Allbp	0.22	0.25	0.22	0.28
2. Allhyper	0.14	0.17	0.14	0.17
3. Allhypo	0.09	0.11	0.09	0.13
4. Allrep	0.06	0.08	0.06	0.11
5. Breast-cancer	0	0.02	0.02	0.05
6. Breast-w	0.03	0.03	0.03	0.06
7. Dermatology	0.02	0.03	0.02	0.05
8. Diabetes	0.05	0.06	0.05	0.08
9. DNA	0.11	0.13	0.11	0.16
10. Echocardiogram	0	0.02	0	0.03
11. Heart-disease	0.11	0.14	0.11	0.16
12. Heart-h	0.02	0.03	0.02	0.06
13. Heart-Statlog	0.02	0.03	0.02	0.06
14. Hepatitis	0.02	0.02	0.02	0.05
15. Hypothyroid	0.09	0.17	0.09	0.17
16. Liver-disorders	0.02	0.03	0.02	0.06
17. Lung cancer	0.02	0.02	0.02	0.05
18. Promoters	0.02	0.02	0.02	0.03
19. Sick-euthyroid	0.16	0.31	0.16	0.23
20. Splice-junction	0.14	0.20	0.17	0.19
21. Thyroid	0.02	0.02	0.02	0.05

จากข้อมูลในตารางที่ 4.3 เมื่อนำผลการทดสอบข้อมูลที่ได้มาวิเคราะห์โดยการสร้างเป็นกราฟเปรียบเทียบเวลาที่ใช้ในการสร้างต้นไม้ตัดสินใจ เมื่อใช้เทคนิคการตัดกิ่งต้นไม้ตัดสินใจด้วยวิธี EBP, REP และ REP+ พร้อมทั้งแสดงผลการทดสอบกับต้นไม้ตัดสินใจที่ไม่ได้ตัดกิ่งด้วยสามารถแสดงผลได้ดังรูปที่ 4.3 ดังนี้



รูปที่ 4.3 กราฟเปรียบเทียบเวลาที่ใช้ในการสร้างต้นไม้ตัดสินใจด้วยวิธี EBP, REP และ REP+

จากตารางที่ 4.3 แสดงเวลาที่ใช้ในการสร้างต้นไม้ตัดสินใจ เมื่อทดสอบข้อมูลด้วยเทคนิคการตัดกิ่ง EBP, REP และ REP+ พร้อมทั้งแสดงผลการทดสอบกับต้นไม้ตัดสินใจที่ไม่ได้ตัดกิ่งด้วย สามารถนำผลที่ได้มาสรุปและสร้างเป็นตารางเพื่อแสดงการเปรียบเทียบระหว่างเทคนิคต่าง ๆ ในแนวคอลัมน์และเทคนิคต่าง ๆ ในแนวแถวของตาราง โดยแสดงจำนวนชุดข้อมูลที่ให้ประสิทธิภาพสูงกว่าเมื่อนำแต่ละเทคนิคมาเปรียบเทียบกัน แสดงได้ในตารางที่ 4.4 ดังต่อไปนี้

ตารางที่ 4.4 ผลสรุปจำนวนชุดข้อมูลเมื่อเปรียบเทียบเวลาที่ใช้ในการสร้างต้นไม้ตัดสินใจของเทคนิค EBP, REP และ REP+

เทคนิคที่ใช้	REP+	REP	EBP	ไม่ตัดกิ่ง
REP+	-	21	17	21
REP	0	-	0	2
EBP	2	15	-	16
ไม่ตัดกิ่ง	0	0	0	-

ตารางที่ 4.4 แสดงให้เห็นว่า เทคนิคการตัดกิ่งด้วยวิธี REP ใช้เวลาน้อยกว่าในการสร้างต้นไม้ตัดสินใจเมื่อเปรียบเทียบกับเทคนิค EBP ด้วยจำนวนข้อมูล 15 ชุดข้อมูล จากข้อมูลทั้งหมด 21 ชุดข้อมูล ส่วนการใช้เทคนิคการตัดกิ่งด้วยวิธี EBP นั้นไม่มีชุดข้อมูลใดที่ใช้เวลาในการสร้างต้นไม้ตัดสินใจที่น้อยกว่าเทคนิค REP อยู่เลย และเมื่อนำต้นไม้ตัดสินใจที่ไม่ได้ตัดกิ่งเปรียบเทียบเวลาที่ใช้ในการสร้างต้นไม้ตัดสินใจกับเทคนิค REP จะเห็นว่าใช้เวลาน้อยกว่าเป็นจำนวน 2 ชุดข้อมูล และใช้เวลาน้อยกว่าเทคนิค EBP เป็นจำนวน 16 ชุดข้อมูล และเมื่อเปรียบเทียบกับต้นไม้ตัดสินใจที่ไม่ได้ตัดกิ่งจะไม่มีชุดข้อมูลใดที่เทคนิค EBP, REP และ REP+ ใช้เวลาน้อยกว่าในการสร้างต้นไม้ตัดสินใจเลย เนื่องจากเทคนิคการตัดกิ่งที่ทดสอบข้อมูลในงานวิจัยนี้เป็นวิธีการตัดกิ่งที่ทำงานหลังจากต้นไม้ตัดสินใจได้สร้างขึ้นอย่างสมบูรณ์แล้ว เวลาที่ใช้ในการสร้างต้นไม้ตัดสินใจของเทคนิค EBP, REP และ REP+ จึงเป็นเวลาที่ใช้ในการสร้างต้นไม้ตัดสินใจที่ไม่ได้ตัดกิ่งรวมกับเวลาที่ใช้ในการตัดกิ่งต้นไม้ตัดสินใจ

จากตารางที่ 4.4 ยังแสดงให้เห็นว่า เทคนิคการตัดกิ่งด้วยวิธี REP+ ใช้เวลามากกว่าในการสร้างต้นไม้ตัดสินใจเมื่อเปรียบเทียบกับเทคนิค EBP และ REP ในหลาย ๆ ชุดข้อมูล เนื่องจากเทคนิคการตัดกิ่ง REP+ ต้องใช้เวลาเพิ่มขึ้นในการคำนวณค่าทางสถิติ เพื่อตรวจสอบความสัมพันธ์กันระหว่างกลุ่มของข้อมูลที่ได้จากการทำนายของต้นไม้ตัดสินใจ และกลุ่มของข้อมูลที่เป็นจริงจากชุดข้อมูลฝึกในทุก ๆ ชุดข้อมูลที่ใช้ทดสอบ

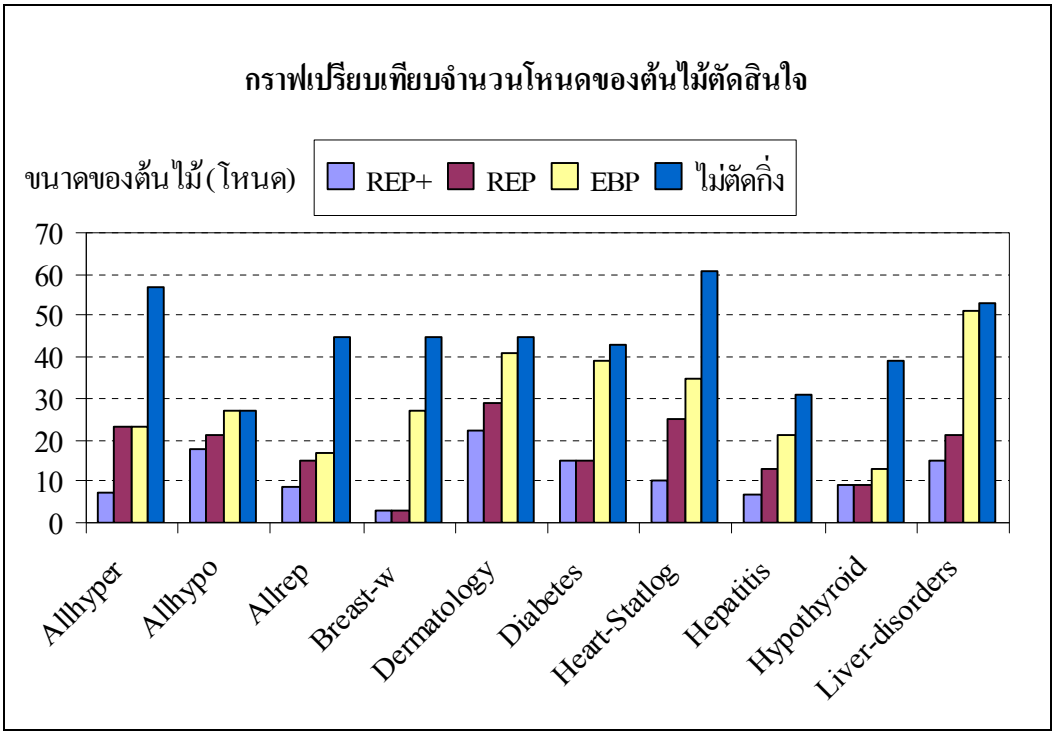
4.2.2 การเปรียบเทียบขนาดของต้นไม้ตัดสินใจ

ตารางที่ 4.5 ต่อไปนี้แสดงประสิทธิภาพของเทคนิคการตัดกิ่งต้นไม้ตัดสินใจด้วยวิธี EBP, REP และ REP+ และประสิทธิภาพของต้นไม้ตัดสินใจที่ไม่ได้ตัดกิ่ง โดยเปรียบเทียบขนาดของต้นไม้ตัดสินใจ สามารถแสดงผลการทดสอบข้อมูลได้ดังนี้

ตารางที่ 4.5 ขนาดของต้นไม้ตัดสินใจที่ได้จากเทคนิค EBP, REP และ REP+

ชื่อชุดข้อมูล	ขนาดของต้นไม้ตัดสินใจ (โหนด)			
	ไม่ตัดกิ่ง	EBP	REP	REP+
1. Allbp	127	51	49	34.7
2. Allhyper	57	23	23	7.0
3. Allhypo	27	27	21	17.7
4. Allrep	45	17	15	8.7
5. Breast-cancer	179	6	22	7.7
6. Breast-w	45	27	3	3.0
7. Dermatology	45	41	29	22.3
8. Diabetes	43	39	15	15.0
9. DNA	313	129	97	91.7
10. Echocardiogram	13	3	3	3.0
11. Heart-disease	316	59	79	6.2
12. Heart-h	47	10	8	8.0
13. Heart-Statlog	61	35	25	10.3
14. Hepatitis	31	21	13	6.7
15. Hypothyroid	39	13	9	9.0
16. Liver-disorders	53	51	21	15.0
17. Lung cancer	15	11	3	2.3
18. Promoters	37	25	5	5.0
19. Sick-euthyroid	95	25	17	10.7
20. Splice-junction	561	229	169	155.2
21. Thyroid	17	17	5	5.0

จากข้อมูลในตารางที่ 4.5 เมื่อนำผลการทดสอบข้อมูลที่ได้มาวิเคราะห์โดยการสร้างเป็นกราฟเปรียบเทียบขนาดของต้นไม้ตัดสินใจ เมื่อใช้เทคนิคการตัดกิ่งต้นไม้ตัดสินใจด้วยวิธี EBP, REP และ REP+ พร้อมทั้งแสดงผลการทดสอบกับต้นไม้ตัดสินใจที่ไม่ได้ตัดกิ่งด้วย สามารถแสดงผลได้ดังรูปที่ 4.4 ดังนี้



รูปที่ 4.4 กราฟเปรียบเทียบขนาดของต้นไม้ตัดสินใจที่ได้จากเทคนิค EBP, REP และ REP+

จากตารางที่ 4.5 แสดงขนาดของต้นไม้ตัดสินใจ เมื่อทดสอบข้อมูลด้วยเทคนิคการตัดกิ่ง EBP, REP และ REP+ พร้อมทั้งแสดงผลการทดสอบกับต้นไม้ตัดสินใจที่ไม่ได้ตัดกิ่งด้วย สามารถนำผลที่ได้มาสรุปและสร้างเป็นตารางแสดงการเปรียบเทียบระหว่างเทคนิคต่าง ๆ ในแนวคอลัมน์และเทคนิคต่าง ๆ ในแนวแถวของตาราง โดยแสดงจำนวนชุดข้อมูลที่ให้ประสิทธิภาพสูงกว่า แสดงได้ในตารางที่ 4.6 ดังต่อไปนี้

ตารางที่ 4.6 ผลสรุปจำนวนชุดข้อมูลเมื่อเปรียบเทียบขนาดของต้นไม้ตัดสินใจที่ได้จากเทคนิค EBP, REP และ REP+

เทคนิคที่ใช้	REP+	REP	EBP	ไม่ตัดกิ่ง
REP+	-	0	1	0
REP	14	-	2	0
EBP	19	17	-	0
ไม่ตัดกิ่ง	21	21	19	-

ตารางที่ 4.6 แสดงให้เห็นการเปรียบเทียบขนาดของต้นไม้ตัดสินใจ โดยใช้จำนวนโหนดทั้งหมดของต้นไม้เปรียบเทียบกัน เทคนิคการตัดกิ่งต้นไม้ตัดสินใจด้วยวิธี REP จะให้จำนวนโหนดของต้นไม้หลังจากการตัดกิ่งแล้วน้อยกว่าเทคนิค EBP เป็นจำนวน 17 ชุดข้อมูล จากการทดสอบทั้งหมด 21 ชุดข้อมูล และให้จำนวนโหนดของต้นไม้น้อยกว่าต้นไม้ที่ไม่ได้ตัดกิ่งในทุกชุดข้อมูล ส่วนการใช้เทคนิคการตัดกิ่งด้วยวิธี EBP ให้จำนวนโหนดของต้นไม้หลังจากการตัดกิ่งแล้วน้อยกว่าเทคนิค REP จำนวน 2 ชุดข้อมูล และขนาดของต้นไม้เปรียบเทียบกับต้นไม้ที่ไม่ได้ตัดกิ่งมีจำนวนโหนดน้อยกว่าเป็นจำนวน 19 ชุดข้อมูล

จากตารางที่ 4.6 ยังแสดงให้เห็นการเปรียบเทียบขนาดของต้นไม้ตัดสินใจของเทคนิคการตัดกิ่งด้วยวิธี REP+ โดยค่าที่แสดงในตารางเป็นค่าเฉลี่ยที่ได้เมื่อใช้ค่าระดับนัยสำคัญต่าง ๆ ทดสอบกับข้อมูลเปรียบเทียบกับเทคนิค EBP และ REP โดยใช้จำนวนโหนดทั้งหมดหลังจากการตัดกิ่งแล้วของต้นไม้ตัดสินใจเปรียบเทียบกัน จะเห็นว่าไม่มีชุดข้อมูลใดที่เทคนิคการตัดกิ่งต้นไม้ตัดสินใจด้วยวิธี REP ให้จำนวนโหนดของต้นไม้หลังจากการตัดกิ่งแล้วน้อยกว่าเทคนิค REP+ และเทคนิคการตัดกิ่งต้นไม้ตัดสินใจด้วยวิธี EBP ให้จำนวนโหนดของต้นไม้หลังจากการตัดกิ่งแล้วน้อยกว่าเทคนิค REP+ เป็นจำนวน 1 ชุดข้อมูลจากการทดสอบทั้งหมด 21 ชุดข้อมูล ส่วนการใช้เทคนิคการตัดกิ่งด้วยวิธี REP+ ให้จำนวนโหนดของต้นไม้หลังจากการตัดกิ่งแล้วน้อยกว่าเทคนิค REP จำนวน 14 ชุดข้อมูล และการใช้เทคนิคการตัดกิ่งด้วยวิธี REP+ ให้จำนวนโหนดของต้นไม้หลังจากการตัดกิ่งแล้วน้อยกว่าเทคนิค EBP จำนวน 19 ชุดข้อมูลจากการทดสอบทั้งหมด 21 ชุดข้อมูล

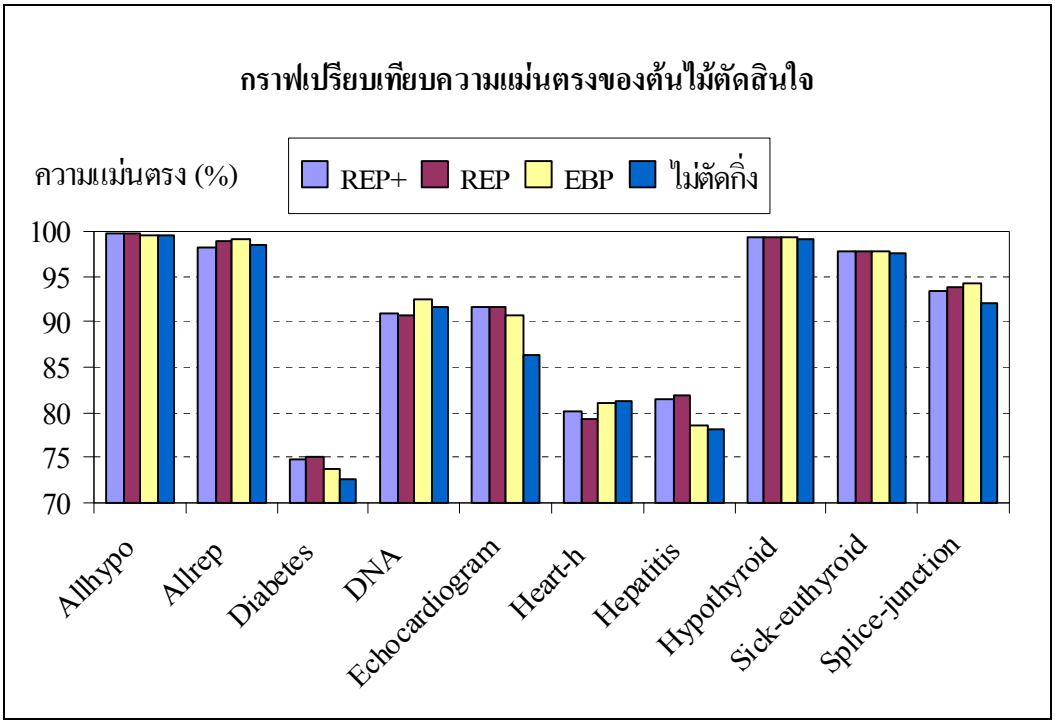
4.2.3 การเปรียบเทียบความแม่นตรงในการจำแนกกลุ่มของข้อมูล

ตารางที่ 4.7 ต่อไปนี้แสดงประสิทธิภาพของเทคนิคการตัดกิ่งต้นไม้ตัดสินใจด้วยวิธี EBP, REP และ REP+ และประสิทธิภาพของต้นไม้ตัดสินใจที่ไม่ได้ตัดกิ่ง โดยเปรียบเทียบความแม่นตรงในการจำแนกกลุ่มของข้อมูล สามารถแสดงผลการทดสอบข้อมูลได้ดังนี้

ตารางที่ 4.7 ความแม่นตรงในการจำแนกกลุ่มของข้อมูลด้วยเทคนิค EBP, REP และ REP+

ชื่อชุดข้อมูล	ความแม่นตรง (%)			
	ไม่ตัดกิ่ง	EBP	REP	REP+
1. Allbp	96.60	97.84	96.91	96.50
2. Allhyper	98.97	98.56	98.46	98.15
3. Allhypo	99.49	99.49	99.79	99.79
4. Allrep	98.56	99.07	98.97	98.27
5. Breast-cancer	69.58	75.52	70.98	71.33
6. Breast-w	93.71	94.56	92.13	92.44
7. Dermatology	94.54	93.99	93.99	88.34
8. Diabetes	72.66	73.83	75.00	74.94
9. DNA	91.65	92.41	90.72	90.89
10. Echocardiogram	86.26	90.84	91.60	91.60
11. Heart-disease	53.26	53.59	55.76	54.84
12. Heart-h	81.29	80.95	79.25	80.04
13. Heart-Statlog	74.81	76.67	79.26	77.59
14. Hepatitis	78.06	78.71	81.94	81.51
15. Hypothyroid	99.08	99.24	99.27	99.27
16. Liver-disorders	68.99	68.70	68.12	65.70
17. Lung cancer	40.63	40.63	46.88	43.23
18. Promoters	83.02	81.13	78.30	78.30
19. Sick-euthyroid	97.57	97.88	97.69	97.73
20. Splice-junction	91.97	94.36	93.73	93.37
21. Thyroid	92.09	92.09	88.37	88.53

จากข้อมูลในตารางที่ 4.7 เมื่อนำผลการทดสอบข้อมูลที่ได้มาวิเคราะห์โดยการสร้างเป็นกราฟเปรียบเทียบความแม่นตรงในการจำแนกกลุ่มของข้อมูล เมื่อใช้เทคนิคการตัดกิ่งต้นไม้ตัดสินใจด้วยวิธี EBP, REP และ REP+ พร้อมทั้งแสดงผลการทดสอบกับต้นไม้ตัดสินใจที่ไม่ได้ตัดกิ่งด้วย สามารถแสดงผลได้ดังรูปที่ 4.5 ดังนี้



รูปที่ 4.5 กราฟเปรียบเทียบความแม่นตรงในการจำแนกกลุ่มของข้อมูลด้วยวิธี EBP, REP และ REP+

จากตารางที่ 4.7 แสดงความแม่นตรงในการจำแนกกลุ่มของข้อมูล เมื่อทดสอบข้อมูลด้วยเทคนิคการตัดกิ่ง EBP, REP และ REP+ พร้อมทั้งแสดงผลการทดสอบกับต้นไม้ตัดสินใจที่ไม่ได้ตัดกิ่งด้วย สามารถนำผลที่ได้มาสรุปและสร้างเป็นตารางแสดงการเปรียบเทียบระหว่างเทคนิคต่าง ๆ ในแนวคอลัมน์และเทคนิคต่าง ๆ ในแนวแถวของตาราง โดยแสดงจำนวนชุดข้อมูลที่ให้ประสิทธิภาพสูงกว่า แสดงได้ในตารางที่ 4.8 ดังต่อไปนี้

ตารางที่ 4.8 ผลสรุปจำนวนชุดข้อมูลเมื่อเปรียบเทียบความแม่นตรงในการจำแนกกลุ่มของข้อมูลด้วยเทคนิค EBP, REP และ REP+

เทคนิคที่ใช้	REP+	REP	EBP	ไม่ตัดกิ่ง
REP+	-	11	13	10
REP	6	-	12	8
EBP	8	8	-	5
ไม่ตัดกิ่ง	11	13	13	-

ตารางที่ 4.8 เป็นผลสรุปการเปรียบเทียบความแม่นตรงในการจำแนกกลุ่มของข้อมูล โดยเมื่อทดสอบกับข้อมูลโดยใช้เทคนิคการตัดกิ่งด้วยวิธี REP ให้ความแม่นตรงสูงกว่าการใช้เทคนิค EBP เป็นจำนวน 8 ชุดข้อมูล และให้ความแม่นตรงสูงกว่าต้นไม้ตัดสินใจที่ไม่ได้ตัดกิ่งเป็นจำนวน 13 ชุดข้อมูล เมื่อทดสอบข้อมูลโดยใช้เทคนิค EBP ให้ความแม่นตรงสูงกว่าการใช้เทคนิค REP เป็นจำนวน 12 ชุดข้อมูล และให้ความแม่นตรงสูงกว่าต้นไม้ตัดสินใจที่ไม่ได้ตัดกิ่งเป็นจำนวน 13 ชุดข้อมูล

จากตารางที่ 4.8 ยังแสดงให้เห็นการเปรียบเทียบความแม่นตรงในการจำแนกกลุ่มของข้อมูลเมื่อทดสอบกับข้อมูลโดยใช้เทคนิคการตัดกิ่งด้วยวิธี REP+ โดยค่าที่แสดงในตารางเป็นค่าความแม่นตรงเฉลี่ยที่ได้เมื่อใช้ค่าระดับนัยสำคัญต่าง ๆ ทดสอบกับข้อมูลเปรียบเทียบกับเทคนิค EBP และ REP โดยเมื่อทดสอบกับข้อมูลโดยใช้เทคนิคการตัดกิ่งด้วยวิธี REP ให้ความแม่นตรงสูงกว่าการเทคนิค REP+ เป็นจำนวน 11 ชุดข้อมูล เมื่อทดสอบข้อมูลโดยใช้เทคนิค REP+ ให้ความแม่นตรงสูงกว่าการใช้เทคนิค REP เป็นจำนวน 6 ชุดข้อมูล และเมื่อทดสอบกับข้อมูลโดยใช้เทคนิคการตัดกิ่งด้วยวิธี EBP ให้ความแม่นตรงสูงกว่าการเทคนิค REP+ เป็นจำนวน 13 ชุดข้อมูล เมื่อทดสอบข้อมูลโดยใช้เทคนิค REP+ ให้ความแม่นตรงสูงกว่าการใช้เทคนิค EBP เป็นจำนวน 8 ชุดข้อมูลจากการทดสอบทั้งหมด 21 ชุดข้อมูล

4.3 การอภิปรายผล

ผลการทดสอบประสิทธิภาพของเทคนิคการตัดกิ่งต้นไม้ตัดสินใจด้วยวิธี REP EBP และเทคนิค REP+ ที่พัฒนาขึ้นในงานวิจัยนี้ โดยทดสอบกับข้อมูลทางด้านวิทยาศาสตร์ที่ได้คัดเลือกมาแล้วจำนวน 21 ชุดข้อมูล เพื่อเปรียบเทียบเวลาที่ใช้ในการสร้างต้นไม้ตัดสินใจ ขนาดของต้นไม้ตัดสินใจที่ได้หลังจากการตัดกิ่ง และความแม่นตรงในการจำแนกกลุ่มของข้อมูล โดยผลที่ได้จากการทดสอบปรากฏอยู่ในตารางและรูปที่แสดงไว้แล้วข้างต้น สามารถสรุปในประเด็นที่สำคัญได้ดังนี้

4.3.1 การทดสอบค่าระดับความมีนัยสำคัญของเทคนิค REP+

1) เทคนิคการตัดกิ่งต้นไม้ตัดสินใจ REP+ ที่พัฒนาขึ้นในงานวิจัยนี้ สามารถกำหนดค่าระดับความมีนัยสำคัญได้ทั้งหมด 6 ค่า ได้แก่ 0.01, 0.025, 0.05, 0.1, 0.25 และ 0.5 เพื่อทดสอบกับข้อมูลแต่ละชุด การใช้ค่าระดับความมีนัยสำคัญต่าง ๆ ส่งผลต่อประสิทธิภาพของต้นไม้ตัดสินใจที่สังเคราะห์ขึ้น

2) เมื่อทดสอบข้อมูลแต่ละชุดด้วยระดับความมีนัยสำคัญที่ลดลง ส่งผลต่อขนาดของต้นไม้ตัดสินใจทำให้จำนวนโหนดของต้นไม้มีจำนวนลดลง โดยการใช้ระดับความมีนัยสำคัญ

เท่ากับ 0.01 จะได้ต้นไม้ตัดสินใจที่มีขนาดเล็กที่สุด

3) การใช้ระดับความมีนัยสำคัญที่มีค่าลดลง ส่งผลต่อประสิทธิภาพของต้นไม้ตัดสินใจ ทำให้ได้ต้นไม้ตัดสินใจที่มีความซับซ้อนลดลงมาก โดยที่ความแม่นตรงในการจำแนกกลุ่มของข้อมูลมีแนวโน้มใกล้เคียงกันในแต่ละระดับความมีนัยสำคัญที่ใช้ทดสอบกับข้อมูล โดยชุดข้อมูลที่ให้ความแม่นตรงเพิ่มขึ้นได้แก่ DNA และ Heart-h ส่วนชุดข้อมูลที่ให้ความแม่นตรงลดลงมากได้แก่ Dermatology, Heart-Statlog, Liver-disorders และ Lung cancer เนื่องจากต้นไม้ตัดสินใจที่ได้จากชุดข้อมูลเหล่านี้มีขนาดเล็กเกินไป จนทำให้สูญเสียข้อมูลที่จำเป็นในการทำนายกลุ่มของข้อมูล

4.3.2 การเปรียบเทียบประสิทธิภาพของเทคนิค EBP, REP และ REP+

1) การตัดกิ่งต้นไม้ตัดสินใจด้วยวิธี REP ใช้เวลาในการสร้างต้นไม้ตัดสินใจน้อยกว่าการตัดกิ่งด้วยวิธี EBP ส่วนการตัดกิ่งด้วยวิธี EBP จะใช้เวลาในการสร้างต้นไม้ตัดสินใจค่อนข้างมากเมื่อเปรียบเทียบกับเทคนิค REP โดยใช้เวลามากกว่า 15 ชุดข้อมูล จากข้อมูลทดสอบทั้งหมด 21 ชุดข้อมูล และการตัดกิ่งต้นไม้ตัดสินใจด้วยวิธี EBP, REP และ REP+ จะใช้เวลามากขึ้นจากการสร้างต้นไม้ตัดสินใจที่ไม่ได้ตัดกิ่ง เนื่องจากเทคนิคการตัดกิ่งที่ทดสอบข้อมูลในงานวิจัยนี้เป็นวิธีการตัดกิ่งที่ทำงานหลังจากต้นไม้ตัดสินใจได้สร้างขึ้นอย่างสมบูรณ์แล้ว เวลาที่ใช้ในการสร้างต้นไม้ตัดสินใจของเทคนิค EBP, REP และ REP+ จึงเป็นเวลาที่ใช้ในการสร้างต้นไม้ตัดสินใจที่ไม่ได้ตัดกิ่งรวมกับเวลาที่ใช้ในการตัดกิ่งต้นไม้ตัดสินใจ

2) เทคนิคการตัดกิ่งด้วยวิธี REP+ ใช้เวลามากกว่าในการสร้างต้นไม้ตัดสินใจเมื่อเปรียบเทียบกับเทคนิค EBP และ REP ในหลาย ๆ ชุดข้อมูล เนื่องจากเทคนิคการตัดกิ่ง REP+ ต้องใช้เวลาเพิ่มขึ้นในการคำนวณค่าทางสถิติ เพื่อตรวจสอบความสัมพันธ์กันระหว่างกลุ่มของข้อมูลที่ได้จากการทำนายของต้นไม้ตัดสินใจและกลุ่มของข้อมูลที่เป็นจริงจากชุดข้อมูลฝึกในทุก ๆ ชุดข้อมูลที่ใช้ทดสอบ

3) เมื่อเปรียบเทียบจากขนาดของต้นไม้ตัดสินใจ โดยพิจารณาจากจำนวนโหนดทั้งหมดของต้นไม้ตัดสินใจ การใช้เทคนิคการตัดกิ่งด้วยวิธี REP ให้จำนวนโหนดของต้นไม้หลังจากการตัดกิ่งแล้วน้อยกว่าเทคนิค EBP โดยมีเพียง 2 ชุดข้อมูลที่เทคนิค EBP ให้จำนวนโหนดของต้นไม้น้อยกว่าได้แก่ชุดข้อมูล Breast-cancer และ Heart-disease

4) เมื่อเปรียบเทียบจากขนาดของต้นไม้ตัดสินใจ โดยพิจารณาจากจำนวนโหนดทั้งหมดของต้นไม้ตัดสินใจที่ได้หลังจากใช้เทคนิคการตัดกิ่งแล้ว การตัดกิ่งด้วยวิธี REP+ ให้จำนวนโหนดของต้นไม้น้อยกว่าที่ได้จากเทคนิค REP เป็นจำนวน 14 ชุดข้อมูล จากข้อมูลทดสอบทั้งหมด 21 ชุดข้อมูล และการใช้เทคนิคการตัดกิ่งด้วยวิธี REP+ ให้จำนวนโหนดของต้นไม้หลังจากการตัด

กิ่งแล้วน้อยกว่าเทคนิค EBP จำนวน 19 ชุดข้อมูล โดยจำนวนโหนดของเทคนิค REP+ ที่นำมาเปรียบเทียบประสิทธิภาพจะใช้ค่าเฉลี่ยที่ได้เมื่อใช้ค่าระดับนัยสำคัญต่าง ๆ ทดสอบกับข้อมูล

5) เมื่อเปรียบเทียบความแม่นตรงในการจำแนกกลุ่มข้อมูลของต้นไม้ตัดสินใจที่ได้จากเทคนิคการตัดกิ่งด้วยวิธี REP และ EBP จะเห็นได้ว่าทั้งสองเทคนิคให้ความแม่นตรงที่ใกล้เคียงกัน แต่การตัดกิ่งด้วยวิธี REP จะให้ต้นไม้ตัดสินใจที่มีขนาดเล็กกว่าอย่างมีนัยสำคัญในหลายชุดข้อมูล

6) เมื่อเปรียบเทียบความแม่นตรงในการจำแนกกลุ่มข้อมูลของต้นไม้ตัดสินใจที่ได้จากเทคนิคการตัดกิ่งด้วยวิธี REP และ REP+ จะเห็นได้ว่าทั้งสองเทคนิคให้ความแม่นตรงที่ใกล้เคียงกัน โดยเมื่อทดสอบกับข้อมูลโดยใช้เทคนิคการตัดกิ่งด้วยวิธี REP ให้ความแม่นตรงสูงกว่าการเทคนิค REP+ เป็นจำนวน 11 ชุดข้อมูล เมื่อทดสอบข้อมูลโดยใช้เทคนิค REP+ ให้ความแม่นตรงสูงกว่าการใช้เทคนิค REP เป็นจำนวน 6 ชุดข้อมูล และเมื่อทดสอบกับข้อมูลโดยใช้เทคนิคการตัดกิ่งด้วยวิธี EBP ให้ความแม่นตรงสูงกว่าการเทคนิค REP+ เป็นจำนวน 13 ชุดข้อมูล เมื่อทดสอบข้อมูลโดยใช้เทคนิค REP+ ให้ความแม่นตรงสูงกว่าการใช้เทคนิค EBP เป็นจำนวน 8 ชุดข้อมูล จากการทดสอบทั้งหมด 21 ชุดข้อมูล แต่การตัดกิ่งด้วยวิธี REP+ จะสามารถลดขนาดของต้นไม้ตัดสินใจลงได้ด้วย โดยค่าความแม่นตรงของเทคนิค REP+ ที่นำมาเปรียบเทียบประสิทธิภาพจะใช้ค่าเฉลี่ยที่ได้เมื่อใช้ค่าระดับนัยสำคัญต่าง ๆ ทดสอบกับข้อมูล

บทที่ 5
บทสรุป

การค้นหารูปแบบหรือโมเดลเพื่อการจำแนกข้อมูลเป็นกระบวนการหนึ่งที่สำคัญและนิยมนำมาใช้อย่างกว้างขวางในงานการทำเหมืองข้อมูล ซึ่งเป็นการค้นหาความรู้และความสัมพันธ์ที่ซ่อนอยู่ในชุดข้อมูลโดยอัตโนมัติ ในปัจจุบันได้ถูกนำไปประยุกต์ใช้ในงานหลายประเภท เช่น ด้านธุรกิจที่ช่วยในการตัดสินใจของผู้บริหาร ด้านวิทยาศาสตร์และการแพทย์ ได้แก่ โครงการศึกษาสารพันธุกรรมของมนุษย์ ใช้ในการวินิจฉัยโรคของคนไข้ ใช้ค้นหาผลข้างเคียงของการใช้ยาโดยอาศัยข้อมูลจากแฟ้มประวัติของคนไข้ เป็นต้น โดยเทคนิคหนึ่งที่สำคัญในงานประเภทนี้คือการสร้างต้นไม้ตัดสินใจ ซึ่งเป็นการเรียนรู้โดยการแยกแยะข้อมูลออกเป็นกลุ่มต่าง ๆ โดยใช้คุณสมบัติของข้อมูล ซึ่งเป็นประโยชน์ช่วยให้ผู้ใช้สามารถวิเคราะห์ข้อมูลและตัดสินใจได้ถูกต้องยิ่งขึ้น การเรียนรู้ด้วยต้นไม้ตัดสินใจเป็นเทคนิคที่ได้รับความสนใจ เนื่องจากการแสดงผลของการวิเคราะห์ข้อมูลสามารถทำความเข้าใจได้ง่าย ใช้เวลาน้อยในการวิเคราะห์ข้อมูล และให้ค่าความถูกต้องที่ดีในการจำแนกกลุ่มข้อมูลและทำนายประเภทข้อมูลที่จะเกิดขึ้นในอนาคต

แต่การนำเทคนิคการวิเคราะห์ข้อมูลอัตโนมัติแบบนี้ไปใช้กับข้อมูลจริง มักจะพบกับปัญหาข้อมูลผิดเพี้ยน ซึ่งอาจเกิดจากการบันทึกข้อมูลผิดพลาดหรือการสูญหายไปบางส่วนของข้อมูล ซึ่งทำให้ต้นไม้ตัดสินใจที่สังเคราะห์ขึ้นจะวิเคราะห์ข้อมูลผิดพลาด และอาจจะสร้างต้นไม้ตัดสินใจที่มีขนาดใหญ่และซับซ้อนเกินไป เนื่องจากพยายามที่จะขยายโครงสร้างให้สามารถอธิบายข้อมูลรบกวนเหล่านั้นให้ได้ การตัดกิ่งต้นไม้ตัดสินใจจึงเป็นวิธีที่ใช้แก้ปัญหาการเจาะจงโมเดลกับข้อมูลมากเกินไป โดยการตัดกิ่งต้นไม้ที่มีความน่าเชื่อถือน้อยออกไปจากต้นไม้ที่เติบโตเต็มที่แล้ว เพื่อลดความซับซ้อนที่เกิดขึ้นและยังคงสามารถใช้ต้นไม้ตัดสินใจจำแนกข้อมูลใหม่ได้อย่างถูกต้อง

งานวิจัยนี้มีจุดมุ่งหมายเพื่อที่จะพัฒนาวิธีการตัดกิ่งต้นไม้ตัดสินใจได้อย่างเหมาะสม โดยใช้ข้อมูลทดสอบทางด้านวิทยาศาสตร์ที่เกี่ยวข้องกับทางด้านการแพทย์และการศึกษาโครงสร้างทางพันธุกรรมของมนุษย์ โดยขั้นตอนการดำเนินงานวิจัยเริ่มจากการศึกษาค้นคว้าวิธีการตัดกิ่งต้นไม้ตัดสินใจต่าง ๆ ที่น่าสนใจ ได้แก่ การตัดกิ่งแบบความผิดพลาดลดลง (REP) และการตัดกิ่งโดยใช้ค่าความผิดพลาด (EBP) เพื่อเปรียบเทียบข้อดีและข้อเสียของแต่ละวิธี และนำเอาลักษณะเด่นของวิธีการต่าง ๆ มาประยุกต์ใช้เพื่อพัฒนาวิธีการตัดกิ่งต้นไม้ตัดสินใจที่มีประสิทธิภาพ โดยทดสอบกับข้อมูลจำนวน 21 ชุดข้อมูล เกณฑ์ที่ใช้ในการเปรียบเทียบประสิทธิภาพพิจารณาจาก เวลาที่ใช้ในการ

สร้างต้นไม้ตัดสินใจ จำนวนโหนดของต้นไม้ที่ได้หลังจากการตัดกิ่งแล้ว และความแม่นตรงในการจำแนกกลุ่มของข้อมูลได้อย่างถูกต้อง

การพัฒนาวิธีการตัดกิ่งต้นไม้ตัดสินใจ REP+ ในงานวิจัยนี้ ได้เลือกใช้วิธีการตัดกิ่งต้นไม้ตัดสินใจ REP มาปรับปรุงโดยใช้การทดสอบทางสถิติร่วมกับการตรวจสอบจำนวนความผิดพลาดของโหนดและต้นไม้ย่อยของต้นไม้ตัดสินใจ โดยสถิติที่ใช้คือการทดสอบไคสแควร์และการทดสอบฟิชเชอร์ เพื่อทดสอบสมมติฐานความเป็นอิสระต่อกันระหว่างกลุ่มของข้อมูลที่ได้จากการทำนายโดยใช้ต้นไม้ตัดสินใจและกลุ่มของข้อมูลที่เป็นจริงจากชุดข้อมูลฝึก โดยจะทดสอบแต่ละโหนดภายในของต้นไม้ตัดสินใจนั้น เพื่อเปรียบเทียบประสิทธิภาพของวิธีการตัดกิ่งต้นไม้ตัดสินใจ REP+ กับวิธีการตัดกิ่ง EBP และ REP

5.1 สรุปผลการวิจัย

5.1.1 เมื่อทดสอบเปรียบเทียบประสิทธิภาพของเทคนิคการตัดกิ่งต้นไม้ตัดสินใจ ด้วยวิธี REP และวิธีการตัดกิ่ง EBP พบว่าวิธีการตัดกิ่ง REP สามารถสร้างต้นไม้ตัดสินใจได้อย่างรวดเร็วกว่า และให้จำนวนโหนดของต้นไม้หลังจากการตัดกิ่งแล้วน้อยกว่าที่ได้จากวิธีการตัดกิ่ง EBP แต่ยังคงมีความแม่นตรงเมื่อนำไปใช้งานจำแนกข้อมูล

5.1.2 เทคนิคการตัดกิ่งต้นไม้ตัดสินใจ REP+ ที่พัฒนาขึ้นในงานวิจัยนี้ มีประสิทธิภาพที่สูงกว่าเทคนิคการตัดกิ่งต้นไม้ตัดสินใจด้วยวิธี EBP และ REP โดยทดสอบกับข้อมูลทางด้านวิทยาศาสตร์ที่เกี่ยวข้องกับข้อมูลทางการแพทย์และการศึกษาพันธุกรรมของมนุษย์ จำนวนทั้งหมด 21 ชุดข้อมูล พบว่าเทคนิคการตัดกิ่ง REP+ ให้ต้นไม้ตัดสินใจที่มีจำนวนโหนดของต้นไม้ลดลง ทำให้ได้โมเดลที่มีโครงสร้างไม่ซับซ้อนและยังคงสามารถใช้จำแนกข้อมูลใหม่ได้อย่างถูกต้อง

5.1.3 การใช้การทดสอบทางสถิติมาช่วยประกอบการตัดสินใจ เพื่อตรวจสอบการตัดกิ่งของต้นไม้ตัดสินใจ โดยพิจารณาความสัมพันธ์กันอย่างมีนัยสำคัญระหว่างกลุ่มของข้อมูลที่ได้จากการทำนายโดยใช้ต้นไม้ตัดสินใจ และกลุ่มของข้อมูลที่เป็นจริงจากชุดข้อมูลฝึก สามารถปรับปรุงประสิทธิภาพของเทคนิคการตัดกิ่งต้นไม้ตัดสินใจด้วยวิธี REP

5.1.4 เทคนิคการตัดกิ่งต้นไม้ตัดสินใจ REP+ ที่พัฒนาขึ้น สามารถปรับค่าระดับความมีนัยสำคัญ (α) ที่ใช้ทดสอบกับข้อมูลได้ โดยการใช้ระดับความมีนัยสำคัญที่มีค่าสูงจะเป็นการยอมให้เกิดความผิดพลาดในการทดสอบทางสถิติสูงขึ้น ซึ่งจะส่งผลให้เกิดการตัดกิ่งได้น้อยลงและสร้างต้นไม้ตัดสินใจที่มีขนาดเทียบเท่ากับต้นไม้ตัดสินใจที่ได้จากวิธีการตัดกิ่ง REP

5.1.5 ต้นไม้ตัดสินใจที่มีจำนวนโหนดน้อยเกินไป ส่งผลต่อประสิทธิภาพของต้นไม้ตัดสินใจที่ลดลง เมื่อใช้จำแนกกลุ่มของข้อมูลใหม่ที่ไม่เคยเห็น เนื่องจากการตัดกิ่งที่มีความสำคัญ

เหล่านั้น ส่งผลให้สูญเสียข้อมูลที่จำเป็นในการทำนายกลุ่มของข้อมูล

5.2 การประยุกต์งานวิจัย

วิธีการตัดกิ่งต้นไม้ตัดสินใจ REP+ ที่พัฒนาขึ้น สามารถนำมาใช้ภายหลังการสร้างต้นไม้ตัดสินใจที่สร้างขึ้นสมบูรณ์แล้ว โดยการตัดกิ่งที่มีความน่าเชื่อถือน้อยออกไปจากต้นไม้ ซึ่งอาจเกิดจากการเจาะจงโมเดลกับข้อมูลที่ใช้สร้างต้นไม้ตัดสินใจมากเกินไป เนื่องจากอาจมีข้อมูลรบกวนหรือความผิดปกติปะปนอยู่ในชุดข้อมูล ซึ่งสามารถนำวิธีการตัดกิ่งต้นไม้ตัดสินใจ REP+ มาประยุกต์ใช้งานในกรณีต่าง ๆ ได้

5.2.1 เมื่อต้องการต้นไม้ตัดสินใจที่มีจำนวนโหนดน้อยลงกว่าที่วิธีการตัดกิ่ง REP และ EBP ที่ศึกษาในงานวิจัยนี้ และไม่ทำให้ความแม่นตรงในการจำแนกข้อมูลสูญเสียไปมากนัก

5.2.2 เมื่อข้อมูลที่ต้องการศึกษาวิเคราะห์ อาจจะมีข้อมูลรบกวนหรือความผิดปกติปะปนอยู่ในชุดข้อมูล ควรปรับค่าระดับความมีนัยสำคัญ (α) ให้มีค่าต่ำลง เพื่อให้ได้ต้นไม้ตัดสินใจที่มีจำนวนโหนดน้อยลง

5.3 ข้อเสนอแนะ

การตัดกิ่งต้นไม้ตัดสินใจเป็นเทคนิคหนึ่งที่สำคัญ ที่ช่วยลดความซับซ้อนของต้นไม้ตัดสินใจ โดยการตัดกิ่งของต้นไม้ที่มีน่าเชื่อถือน้อยออกไป วิธีการตัดกิ่งต้นไม้ตัดสินใจ REP+ ที่พัฒนาขึ้น สามารถสร้างต้นไม้ตัดสินใจที่มีจำนวนโหนดน้อยลง และยังคงสามารถจำแนกกลุ่มของข้อมูลใหม่ได้อย่างถูกต้อง ดังนั้นเทคนิคการตัดกิ่งต้นไม้ตัดสินใจด้วยวิธี REP+ จึงน่าจะเป็นประโยชน์แก่นักวิจัยท่านอื่นที่สนใจที่จะพัฒนาองค์ความรู้เพื่องานการจำแนกกลุ่มของข้อมูล โดยแนวทางวิจัยที่จะพัฒนาต่อไปสามารถทำได้หลายแนวทางด้วยกัน ได้แก่

5.3.1 การพัฒนาวิธีการตัดกิ่งต้นไม้ตัดสินใจด้วยวิธี REP+ ต่อไป เพื่อให้สามารถใช้การทดสอบทางสถิติได้กับข้อมูลทุกกลุ่มไม่จำกัดอยู่เฉพาะข้อมูลทางด้านวิทยาศาสตร์เท่านั้น

5.3.2 การวิเคราะห์เพื่อหาแนวทางในการกำหนดค่าระดับความมีนัยสำคัญที่เหมาะสมที่สุดสำหรับข้อมูลที่ใช้ทดสอบ เพื่อให้วิธีการตัดกิ่งต้นไม้ตัดสินใจด้วยวิธี REP+ สามารถสร้างต้นไม้ตัดสินใจได้อย่างประสิทธิภาพ

รายการอ้างอิง

ก้องศักดิ์ จงเกษมวงศ์. (2543). **การตัดเล็มอย่างอ่อนสำหรับต้นไม้ตัดสินใจโดยการใช้แบ็กพรอพาเกชันนิวรอลเน็ตเวิร์ก.** วิทยานิพนธ์ปริญญามหาบัณฑิต สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย.

นิตยา เกิดประสพ. (2547). **รายงานการวิจัย: อัลกอริทึมและเทคนิคที่เหมาะสมกับการสังเคราะห์โมเดลที่ช่วยวินิจฉัยโรคได้อัตโนมัติ.** นครราชสีมา: สาขาวิชาวิศวกรรมคอมพิวเตอร์ สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี.

ยุทธ ไกยวรรณ์. (2546). **สถิติเพื่อการวิจัย.** กรุงเทพมหานคร: ศูนย์สื่อเสริมกรุงเทพ.

Blake, C.and Merz, C. J. (1998). **UCI repository of machine learning databases.** Department of Information and Computer Science, University of California, Irvine, CA. Available: http://www.ics.uci.edu/~mlearn/MLRepository.html

Breiman, L., Friedman, J., Olshen, R., and Stone, C. (1984). **Classification and Regression Trees.** Monterey, CA: Wadsworth International Group.

Breslow, L. A., and Aha, D. W. (1997). Simplifying decision trees: A survey. **Knowledge Engineering Review**, 12(1): 1-40.

Cohen, P. R. and Jensen, D. (1997). Overfitting explained. In **Proceedings of the Sixth International Workshop on Artificial Intelligence and Statistics** (pp 115-122). Ft. Lauderdale, FL.

Esposito, F., Malerba, D., and Semeraro, G. (1997). A comparative analysis of methods for pruning decision trees. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, 19(5): 476-491.

Esposito, F., Malerba, D., Semeraro, G., and Tamma, V. (1999). The effects of pruning methods on the predictive accuracy of induced decision trees. **Applied Stochastic Models in Business and Industry**, 15: 277-299.

Frank, E. (2000). **Pruning decision trees and lists.** Ph.D. thesis, University of Waikato, Department of Computer Science, Hamilton, New Zealand.

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

Minger, J. (1989). An empirical comparison of pruning methods for decision tree induction. **Machine Learning**, 4(2): 227-243.

Murthy, S. K. (1998). Automatic construction of decision trees from data: A multi-disciplinary survey. **Data Mining and Knowledge Discovery**, 2(4): 345-389.

Oates, T. and Jensen, D. (1997). The effects of training set size on decision tree complexity. In D. H. Fisher (ed.). **Proceedings of the Fourteenth International Conference on Machine Learning** (pp 254-261). San Francisco, CA: Morgan Kaufmann.

Oates, T. and Jensen, D. (1998). Large datasets lead to overly complex models: An explanation and a solution. In R. Agrawal, P. Stolorz, and G. Piotetsky-Shapiro (eds.). **Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining** (pp 294-298). Menlo Park, CA: AAAI Press.

Oates, T. and Jensen, D. (1999). Toward a theoretical understanding of why and when decision tree pruning algorithms fail. In **Proceedings of the Sixteenth National Conference on Artificial Intelligence** (pp 372-378). Menlo Park, CA: AAAI Press.

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

Quinlan, J. R. (1987). Simplifying decision trees. **International Journal of Man-Machine Studies**, 27: 221-234.

Quinlan, J. R. and Rivest, R. L. (1989). Inferring decision trees using the minimum description length principle. **Information and Computation**, 80(3): 227-248.

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

Siegel, S. and Castellan, N. J. (1988). **Nonparametric Statistics for the Behavioral Sciences.** New York: McGraw-Hill.

Witten, I. H., and Frank, E. (2005). **Data Mining: Practical Machine Learning Tools and Techniques.** (2nd ed.). San Francisco, CA: Morgan Kaufmann.

ภาคผนวก ก

บทความผลงานวิจัยที่นำเสนอในการประชุมวิชาการวิทยาศาสตร์และเทคโนโลยีแห่งประเทศไทย ครั้งที่ 31

ณ มหาวิทยาลัยเทคโนโลยีสุรนารี

18-20 ตุลาคม 2548

การศึกษาเพื่อเปรียบเทียบประสิทธิภาพของเทคนิคการตัดกิ่งต้นไม้ตัดสินใจ

A COMPARATIVE STUDY OF METHODS FOR PRUNING DECISION TREES

นฤพนต์ ว่องประชานุกูล, นิตยา เกิดประสพ และ กิตติศักดิ์ เกิดประสพ
Narupon Wongprachanukul, Nittaya Kerdprasop and Kittisak Kerdprasop

Data Engineering and Knowledge Discovery (DEKD) Research Unit, School of Computer Engineering, Suranaree University of Technology, Nakhon Ratchasima, Thailand,
E-mail address: narupon@nsrc.or.th, nittaya@sut.ac.th, kerdpras@sut.ac.th

บทคัดย่อ: งานวิจัยนี้เป็นการศึกษาเพื่อเปรียบเทียบประสิทธิภาพของเทคนิคการตัดกิ่งต้นไม้ตัดสินใจที่มีชื่อเสียงสองวิธีคือ Reduced-error pruning และ Error-based pruning โดยมีจุดมุ่งหมายเพื่อวิเคราะห์ค่าความเที่ยงตรงในการจำแนกคลาสข้อมูล เวลาที่ใช้ในการสร้างโมเดล และขนาดของต้นไม้ตัดสินใจ เราทำการทดลองกับสิบชุดข้อมูลด้วยเทคนิคการตัดกิ่งเหล่านี้ เพื่อลดขนาดของต้นไม้และแก้ไขปัญหา “overfitting” ต้นไม้ที่ได้รับการตัดกิ่งแล้วจะใช้เวลาในการสร้างลดลงเนื่องจากขนาดที่เล็กลง และยังคงสามารถจำแนกข้อมูลใหม่ได้อย่างถูกต้อง

Abstract: We make a comparative study of two well-known pruning methods, reduced-error pruning and error-based pruning. The predictive accuracy, the time taken to build the model, and size of the pruned trees are evaluated for each pruning method. We conduct the experiments on ten data sets. Pruning methods aim at simplifying decision trees to avoid overfitting problem. The pruned trees result in faster classification and do not decrease their predictive accuracy.

Introduction: Decision tree is one of the tools used for data mining. The main application area is classification task. The model is built from a set of records, called training set. Each record consists of a number of attribute-value pairs. One of these attributes represents class of the record. We also have a test set for evaluating the performance of a decision tree.

When a decision tree is built, many of the branches may be overly expanded due to noise or outliers in the training set. The built model is too complex, since it tries to classify all records in the training set including noise and outliers. This problem is called “overfitting”. We use tree pruning method to remove the least reliable branches, generally resulting in faster classification and improvement in the ability of the tree to correctly classify unknown data.

We study the performance of the post-pruning approach. A tree node is pruned by removing its branches from a fully grown tree (T_{max}) [1]. In the following subsections, we summarize the concepts of two pruning methods whose performances are evaluated in this paper.

Reduced-error pruning (REP): This method is probably the simplest pruning technique. It uses the pruning set to evaluate the goodness of a subtree of the complete tree. It starts with T_{max} and runs the test data through it. For each internal node, the number of classification errors is counted if the subtree is kept compared with if the subtree is pruned. The decision on whether or not to prune the subtree is based on which alternative yields a minimum error.

A pruning operation involves replacing a subtree by a leaf. REP will perform this operation if it does not increase the total number of classification errors. Traversing the tree in a bottom-up strategy ensures that the result is the smallest pruned tree that has minimum error on the pruning data.

Error-based pruning (EBP): This method is implemented by the well-known decision tree inducer C4.5 [3]. Unlike REP, EBP uses the training set for building and simplifying trees. It visits nodes of T_{max} according to a bottom-up traversal strategy and uses the certainty factor (CF) parameter to control the pruning. CF is used to estimate the upper limit of the probability that an error occurs over the population at a leaf.

The subtree replacement is performed if the error estimate for the expected leaf is not greater than the sum of the error estimates for the current leaf nodes of the subtree. EBP also performs a pruning operation called "subtree raising" that replaces a subtree with its most populated branch if this does not increase the estimated error.

Methodology: We conducted experiments and used the decision tree on ten data sets from UCI Machine Learning Repository [2] with the above pruning methods and use the decision tree inducer C4.5. Model accuracy was tested with ten-fold cross-validation technique. The main characteristics of the data sets are presented in Table 1.

Table 1. The main characteristics of the data sets used for experiments

Data set	No. of Instances	No. of Attributes	No. of Nominal attributes	No. of Numeric attributes	Missing values	No. of Classes
Anneal	898	38	32	6	yes	5
Audiology	226	69	69	0	yes	24
Glass	214	9	0	9	no	7
Glass-2	163	9	0	9	no	2
Hepatitis	155	19	13	6	yes	2
Ionosphere	351	34	0	34	no	2
Iris	150	4	0	4	no	3
Labor	57	16	8	8	yes	2
Soybean	683	35	35	0	yes	19
Vote	435	16	16	0	yes	2

Results, Discussion and Conclusion: We compare the predictive accuracy, the time taken to build the model and size of the pruned trees with the unpruned trees. These results are reported in Table 2.

Table 2. Accuracy, time taken to build the model and size of the pruned trees compare with the unpruned trees

	Time (s)			Tree sizes			Accuracy (%)		
Data set	1	2	3	1	2	3	1	2	3
Anneal	0.55	1.32	2.14	113	78	155	92.87	90.98	93.10
Audiology	0.11	0.22	0.22	47	54	62	71.24	77.43	76.55
Glass	0.11	0.22	0.17	17	59	59	71.50	66.82	65.89
Glass-2	0	0.05	0.05	15	17	17	74.23	80.37	80.37
Hepatitis	0.05	0.11	0.06	13	21	31	81.94	78.71	78.06
Ionosphere	1.65	2.14	1.70	13	35	35	90.60	88.03	88.32
Iris	0	0.06	0	9	9	9	94.67	96.0	96.0
Labor	0	0.05	0	7	5	22	82.46	73.68	78.95
Soybean	0.27	0.55	0.38	120	93	175	87.55	91.51	91.36
Vote	0.06	0.06	0.06	9	11	37	95.63	96.32	96.32

1 = REP, 2 = EBP, and 3 = unpruned

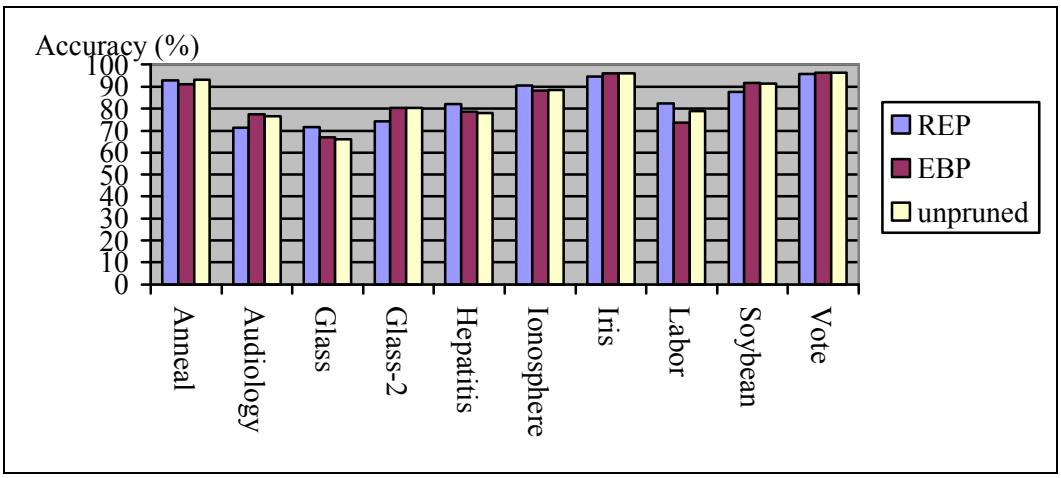


Figure 1. The predictive accuracy of the pruned trees compares with the unpruned trees

Both REP and EBP reduce the size of a fully grown tree by removing some unnecessary branches and do not significantly decrease the predictive accuracy of most final trees. REP produces the pruned tree in the shortest period of time, since it must not estimate classification errors. These experiments are still preliminary and need more systematic and extensive studies including additional comparative studies to other pruning methods.

References: [1]. Esposito, F., Malerba, D., and Semeraro, G. *A Comparative Analysis of Methods for Pruning Decision Trees*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 19, 5, 476-491. 1997.

[2]. Merz, C.J., and Murphy, P.M. *UCI Repository of machine learning databases.* http://www.ics.uci.edu/~mlearn/MLRepository.html. 1996.

[3]. Quinlan, J.R. *C4.5: Programs for Machine Learning*. 1993.

Keywords: decision trees, pruning methods, reduced-error pruning, error-based pruning.

ภาคผนวก ข

โครงสร้างและการทำงานของอัลกอริทึม J48

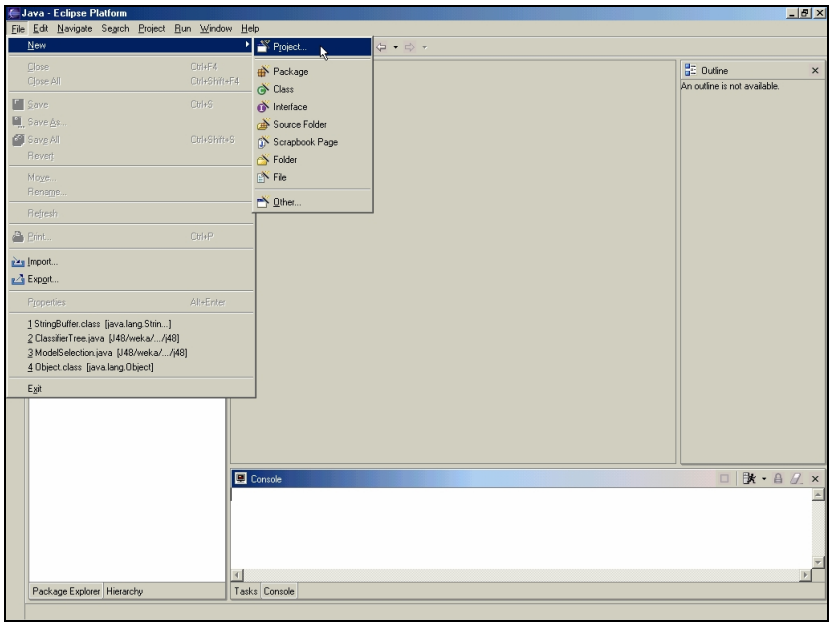
ในการศึกษาอัลกอริทึม J48 ของระบบ WEKA ซึ่งพัฒนาด้วยภาษา Java แบ่งออกเป็น 2 ส่วนด้วยกัน คือ การแยกเอาเฉพาะคลาสที่เกี่ยวข้องกับอัลกอริทึม J48 เท่านั้นออกมาจากทั้งระบบของ WEKA เพื่อศึกษาโครงสร้างของอัลกอริทึม และอีกส่วนก็คือการศึกษารายละเอียดภายในซอร์สโค้ดของอัลกอริทึม และเพื่อให้ง่ายต่อการศึกษาเราจึงเลือกใช้ Eclipse เวอร์ชัน 2.1.3 ซึ่งเป็นแพล็ตฟอร์มที่พัฒนาขึ้นแบบ open source ใช้สำหรับออกแบบและพัฒนา Java application และ web-based application ที่มีประสิทธิภาพ นำมาเป็นเครื่องมือช่วยในการศึกษาโครงสร้างและการทำงานของอัลกอริทึมที่ใช้ในการสร้างต้นไม้ตัดสินใจ

1. โครงสร้างของอัลกอริทึม J48

ในการศึกษาโครงสร้างของอัลกอริทึม เราทำการแยกเอาเฉพาะจาวาคลาสที่เกี่ยวข้องกับการทำงานของอัลกอริทึม J48 ออกมาต่างหากจากระบบ WEKA ซึ่งขั้นตอนการดำเนินงานมีดังนี้

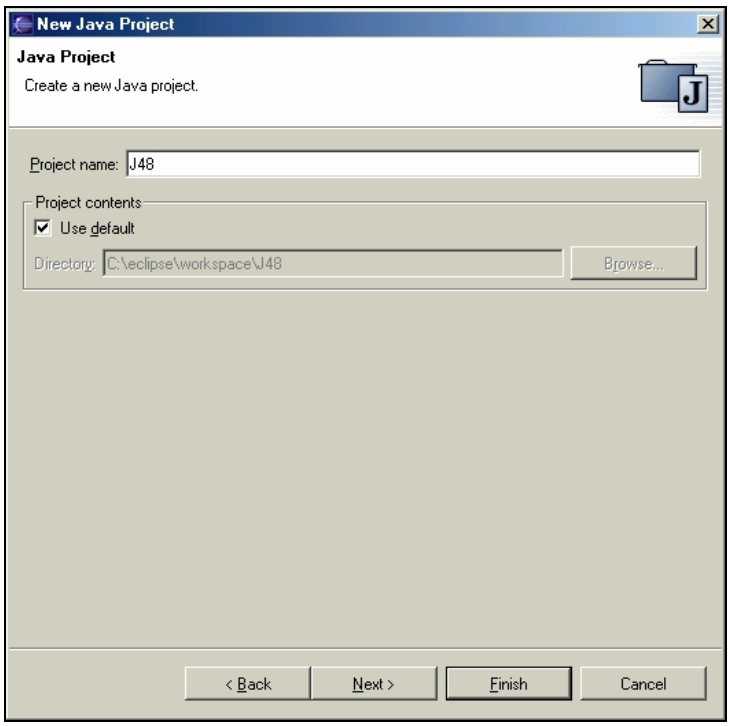
1.1 การสร้างโปรเจ็คใหม่

เริ่มต้นด้วยการสร้างโปรเจ็คใหม่สำหรับจัดเก็บไฟล์ที่เกี่ยวข้องกับการทำงานของอัลกอริทึม J48 โดยคลิกเลือกที่เมนู File > New > Project… ดังรูปที่ ข.1



รูปที่ ข.1 แสดงการเลือกเมนูเพื่อสร้างโปรเจ็คใหม่

เลือกสร้างโปรเจ็คชนิด Java Project และกำหนดชื่อของโปรเจ็คเป็น J48 ดังรูปที่ ข.2



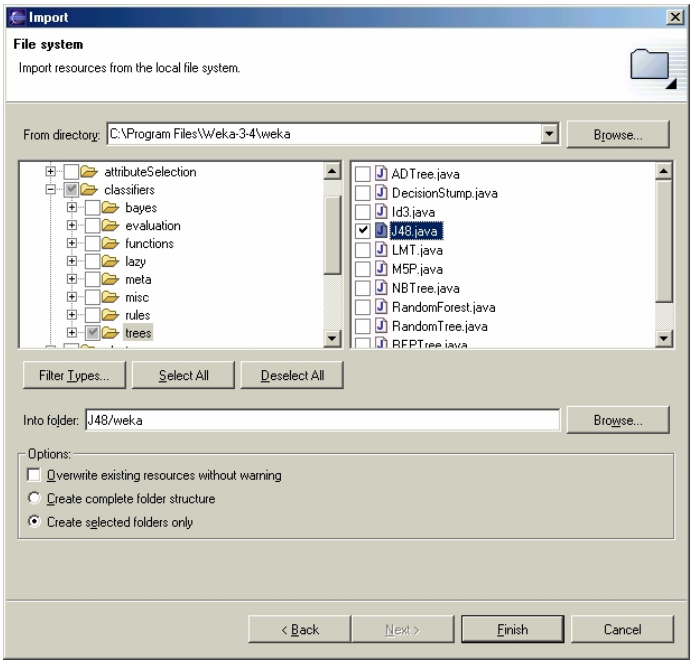
รูปที่ ข.2 แสดงการสร้างโปรเจ็คชนิด Java Project

1.2 การนำเข้า (Import) ไฟล์ที่เกี่ยวข้อง

ขั้นตอนต่อไป เราจะนำเข้าไฟล์ที่เกี่ยวข้องกับการทำงานของอัลกอริทึม J48 โดยใช้ซอร์ส-โค้ดของระบบ WEKA ที่ชื่อว่า weka-src.jar โดยสามารถหาไฟล์นี้ได้ในโฟลเดอร์ที่ติดตั้งระบบ WEKA ในขั้นตอนนี้จะต้องทำการแตกไฟล์ weka-src.jar เสียก่อน โดยใช้คำสั่งต่อไปนี้

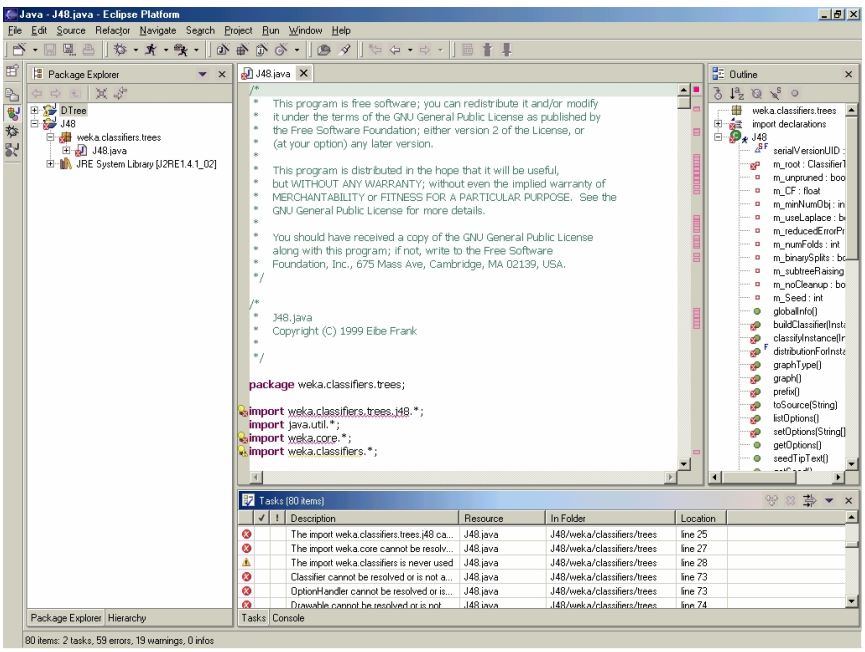
jar – xvf weka-src.jar

สร้าง Package ใหม่ชื่อว่า weka ภายในโปรเจ็ค J48 ของโปรแกรม Eclipse และนำเข้าไฟล์โดยใช้ Import… เลือก source เป็นชนิด File system โดยในลำดับแรกจะนำเข้าไฟล์ J48.java เพียงไฟล์เดียวเท่านั้น ซึ่งอยู่ในโฟล์เดอร์ weka\classifiers\trees\ และระบุ Info folder เป็น J48/weka ดังรูปที่ ข.3 โดยข้อควรระวังสำหรับการเขียนโปรแกรมภาษาจาวาคือตัวพิมพ์ใหญ่กับตัวพิมพ์เล็กจะให้ความหมายต่างกัน (case sensitive)



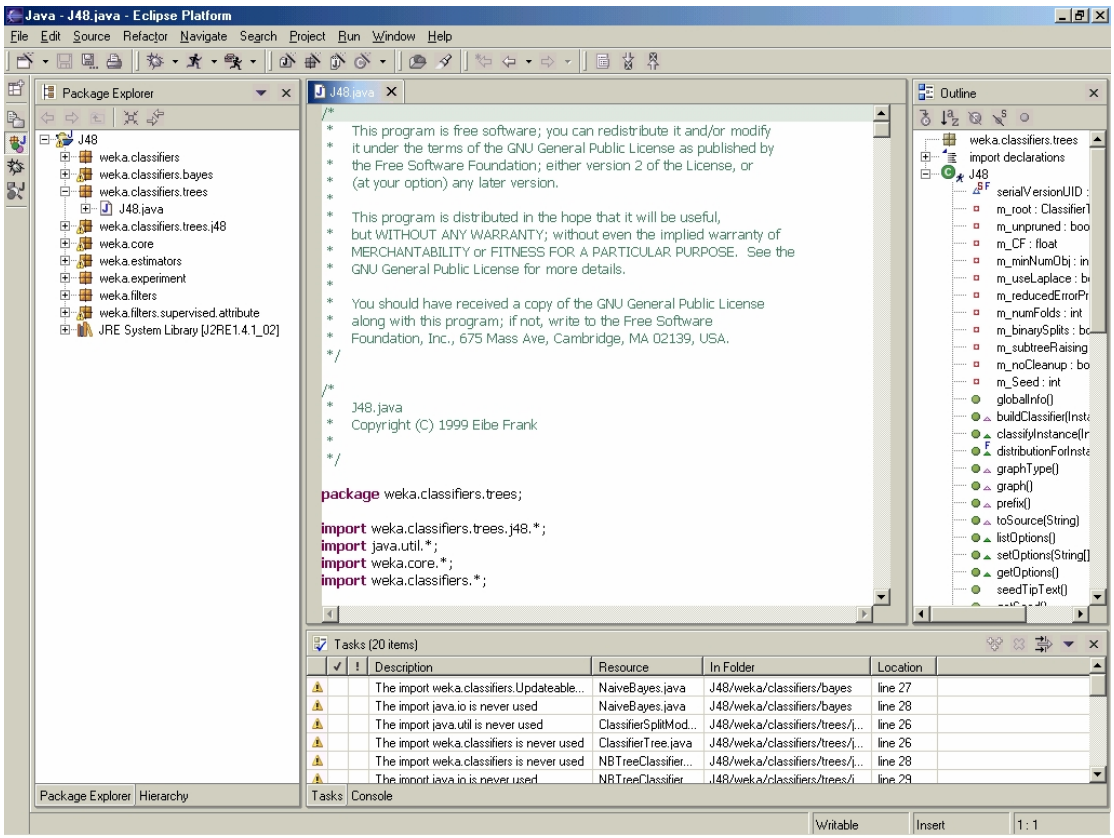
รูปที่ ข.3 แสดงการนำเข้าไฟล์ J48.java

เมื่อนำเข้าไฟล์ J48.java แล้ว จะได้ผลดังแสดงในรูปที่ ข.4



รูปที่ ข.4 แสดงผลการนำเข้าไฟล์ J48.java

จะเห็นได้ว่ามีความผิดพลาดเกิดขึ้นในหลายจุดแสดงในหน้าต่าง Tasks ด้านล่างของรูปที่ ข.4 เนื่องจากภายในคลาส J48 จะมีการเรียกใช้งานคลาสอื่น ๆ ที่เกี่ยวข้องอีกหลายคลาส เราจำเป็นต้องนำเข้าไฟล์ที่เกี่ยวข้องทั้งหมดเข้าสู่โปรเจ็ค J48 โดยใช้วิธีการเดียวกับการนำเข้าไฟล์ J48.java และเมื่อนำเข้าไฟล์ที่เกี่ยวข้องทั้งหมดกับการทำงานของคลาส J48 จนไม่มีความผิดพลาดเกิดขึ้น จะได้ผลดังแสดงในรูปที่ ข.5

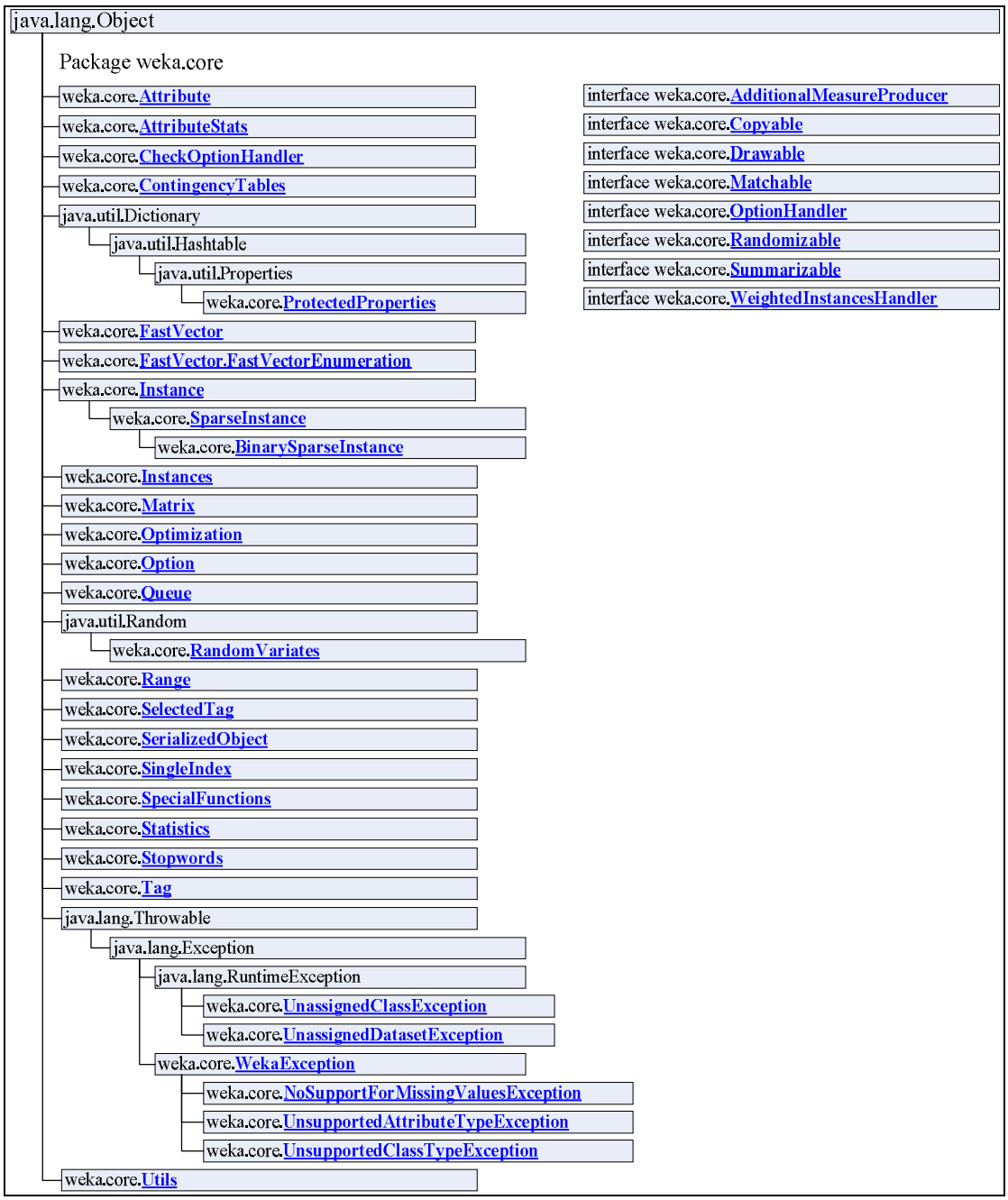


รูปที่ ข.5 แสดงการนำเข้าไฟล์ที่เกี่ยวข้องกับคลาส J48

แพ็คเกจทั้งหมดที่เกี่ยวข้องกับการทำงานของคลาส J48 สามารถสรุปได้ดังนี้

- weka.classifiers
- weka.classifiers.bayes
- weka.classifiers.trees
- weka.classifiers.trees.j48
- weka.core
- weka.estimators
- weka.experiment
- weka.filters
- weka.filters.supervised.attribute

คลาสที่เกี่ยวข้องทั้งหมดของอัลกอริทึม J48



รูปที่ ข.6 คลาสที่เกี่ยวข้องทั้งหมดของอัลกอริทึม J48

รูปที่ ข.6 คลาสที่เกี่ยวข้องทั้งหมดของอัลกอริทึม J48 (ต่อ)

2. การทำงานของอัลกอริทึม J48

การประมวลผลอัลกอริทึม J48 มีการเรียกใช้คลาสในการทำงานหลายคลาส ในที่นี้จะยกมาอธิบายเฉพาะในส่วนของคลาสที่มีความสำคัญเท่านั้น ได้แก่ คลาส J48, คลาส Evaluation, คลาส Instances, คลาส ClassifierTree, และคลาส C45PruneableClassifierTree ดังต่อไปนี้

Class J48

java.lang.Object

└ weka.classifiers.Classifier

└ **weka.classifiers.trees.J48**

เป็นคลาสที่อยู่ในแพ็คเกจ weka.classifiers.trees ได้รับการถ่ายทอดมาจากคลาส Classifier ทำหน้าที่สร้างต้นไม้ตัดสินใจด้วยอัลกอริทึม C4.5 โดยสามารถเลือกให้มีการตัดหรือไม่ตัดกิ่งของต้นไม้ได้ คลาสนี้เป็นคลาสหลักที่ให้ความสำคัญเพื่อศึกษาการทำงานของอัลกอริทึม J48 โดยสามารถกำหนด option หรือพารามิเตอร์เฉพาะให้กับอัลกอริทึมการเรียนรู้นี้ได้หลายแบบซึ่งจะได้อธิบายต่อไป

Class Evaluation

java.lang.Object

└ **weka.classifiers.Evaluation**

เป็นคลาสที่อยู่ในแพ็คเกจ weka.classifiers ทำหน้าที่ประเมินโมเดลการเรียนรู้ ตามค่า option ทั่วไปที่ระบุไว้ในการประมวลผลอัลกอริทึมแบบ Command-line

Class Instances

java.lang.Object

└ **weka.core.Instances**

เป็นคลาสที่อยู่ในแพ็คเกจ weka.core ทำหน้าที่จัดการกับข้อมูลรูปแบบ ARFF file เช่น อ่านไฟล์ข้อมูลฝึก และกำหนดตำแหน่งคอลัมน์ของกลุ่มข้อมูล เป็นต้น

Class ClassifierTree

java.lang.Object

└ **weka.classifiers.trees.j48.ClassifierTree**

เป็นคลาสที่อยู่ในแพ็คเกจ weka.classifiers.trees.j48 ทำหน้าที่จัดการกับโครงสร้างของต้นไม้ ที่สร้างขึ้นสำหรับงาน Classification นอกจากนี้ยังมีเมธอดที่สำคัญในการสร้างต้นไม้ และคืนเนื้อที่หน่วยความจำเมื่อไม่ได้ใช้งาน

Class C45PruneableClassifierTree

java.lang.Object

└ weka.classifiers.trees.j48.ClassifierTree

└ **weka.classifiers.trees.j48.C45PruneableClassifierTree**

เป็นคลาสที่อยู่ในแพ็คเกจ weka.classifiers.trees.j48 ได้รับการถ่ายทอดมาจากคลาส ClassifierTree ทำหน้าที่จัดการกับโครงสร้างต้นไม้ที่เลือกให้มีการตัดกิ่งด้วย โดยใช้กระบวนการของอัลกอริทึม C4.5

2.1 ขั้นตอนการทำงานของอัลกอริทึม J48

1) เริ่มต้นการทำงานที่เมธอด main() ในคลาส J48 ด้วยการสร้างออบเจ็กต์ของคลาส J48 แล้วส่งเป็นค่าพารามิเตอร์ให้กับเมธอด evaluateModel(Classifier classifier, String [] options) ในคลาส Evaluation โดย options เป็นพารามิเตอร์ที่ระบุเมื่อรันคลาส J48 (เช่น –t weather.arff –T weather-test.arff)
2) เรียกใช้เมธอด evaluateModel(Classifier classifier, String [] options) ทำหน้าที่ประเมิน classifier ด้วยค่า option ที่กำหนดให้กับอัลกอริทึมทำงาน
3) เรียกใช้เมธอด getOption() ในคลาส Utils ทำหน้าที่ตรวจสอบและรับค่า option ทั่วไปที่กำหนดไว้ โดยจะทำหน้าที่แยก option แต่ละตัวออกจาก String [] options ที่ส่งมา เพื่อให้ค่ากับตัวแปรของอัลกอริทึม
4) สร้างบัฟเฟอร์เตรียมไว้สำหรับใช้อ่านข้อมูลจากไฟล์ของชุดข้อมูลฝึก และชุดข้อมูลทดสอบ (ถ้ามีการระบุ)
5) สร้างออบเจ็กต์โดยเรียกใช้คอนสตรัคเตอร์ Instances() ทำหน้าที่อ่านข้อมูลในรูปแบบ ARFF file

6) เรียกใช้เมธอด setOptions() ในคลาส J48 ทำหน้าที่ตรวจสอบและรับค่า option เฉพาะของอัลกอริทึมที่กำหนดไว้
7) สร้างออบเจ็กต์ของคลาส Evaluation ทำหน้าที่ประเมินโมเดลการเรียนรู้ ของชุดข้อมูลฝึกและชุดข้อมูลทดสอบ
8) เรียกใช้เมธอด currentTimeMillis() ในคลาส java.lang.System ทำหน้าที่ให้ค่าเวลาเมื่อเริ่มต้นการสร้างโมเดล
9) เรียกใช้เมธอด buildClassifier() ในคลาส J48 ทำหน้าที่สร้างโมเดล โดยเรียกใช้ buildClassifier() ในคลาส C45PruneableClassifierTree ทำหน้าที่สร้างต้นไม้ที่เลือกให้มีการตัดกิ่งได้
10) เรียกใช้เมธอด deleteWithMissingClass() ในคลาส Instances ทำหน้าที่ตรวจสอบแต่ละอินสแตนซ์ของชุดข้อมูล ถ้า class label ไม่มีอยู่ ข้อมูลในอินสแตนซ์นั้นจะไม่ถูกพิจารณา
11) เรียกใช้เมธอด buildTree() ในคลาส ClassifierTree เพื่อค้นหาแอททริบิวต์ที่ทำหน้าที่เป็นโหนดรากของต้นไม้
12) จากโหนดรากสามารถแบ่งข้อมูลในแต่ละอินสแตนซ์ได้เป็น subset ทำการวนซ้ำเรียกใช้เมธอด getNewTree() ในคลาส C45PruneableClassifierTree เพื่อสร้างต้นไม้ของโหนดลูกต่อไป โดยเรียกใช้เมธอด buildTree() ในคลาส ClassifierTree
13) เรียกใช้เมธอด collapse() และเมธอด prune() ในคลาส C45PruneableClassifierTree เพื่อตัดกิ่งของต้นไม้ โดยพิจารณาจากค่าความผิดพลาดของต้นไม้ที่สร้างขึ้นจากชุดข้อมูล
14) เรียกใช้เมธอด cleanup() ในคลาส ClassifierTree เพื่อคืนเนื้อที่หน่วยความจำเมื่อไม่ได้ใช้งานแล้ว
15) เรียกใช้เมธอด currentTimeMillis() ในคลาส java.lang.System อีกครั้งหนึ่งเพื่อนำไปลบกับค่าเวลาเมื่อเริ่มต้นการสร้างโมเดล ได้เวลาที่ใช้ไปในขั้นตอนการสร้างโมเดล
16) เรียกใช้เมธอด currentTimeMillis() ในคลาส java.lang.System ทำหน้าที่ให้ค่าเวลาเมื่อเริ่มต้นการทดสอบโมเดล
17) เรียกใช้เมธอด evaluateModel() ในคลาส Evaluation ทำหน้าที่ทดสอบโมเดลโดยใช้ข้อมูลแต่ละอินสแตนซ์ของชุดข้อมูลฝึก
18) เรียกใช้เมธอด currentTimeMillis() ในคลาส java.lang.System อีกครั้งหนึ่งเพื่อ

นำไปลบกับค่าเวลาเมื่อเริ่มต้นการทดสอบโมเดล ได้เวลาที่ใช้ไปในขั้นตอนการทดสอบโมเดล

19) ถ้าไม่ได้กำหนด option การทดสอบโมเดลเป็นอย่างอื่น อัลกอริทึมจะใช้วิธี cross-validation กับชุดข้อมูลฝึกซึ่งเป็นค่า default โดยเรียกใช้เมธอด crossValidateModel() ในคลาส Evaluation
20) สุดท้ายแสดงผลลัพธ์ทั้งหมดที่ได้จากการสร้างโมเดลออกสู่หน้าจอหลัก

ตารางที่ ข.1 พารามิเตอร์ทั่ว ๆ ไปสำหรับการใช้งานอัลกอริทึมเรียนรู้บนระบบ WEKA

พารามิเตอร์	คำอธิบาย
-t <name of training file>	ระบุแฟ้มข้อมูลที่ใช้ในการฝึก
-T <name of test file>	ระบุแฟ้มข้อมูลที่ใช้ในการทดสอบ
-c <class index>	ระบุลำดับที่ของคลาสแอททริบิวต์ (default: last)
-x <number of folds>	ระบุจำนวน Fold สำหรับการทดสอบแบบ cross-validation (default: 10)
-s <random number seed>	ระบุจำนวน seed สำหรับการทดสอบแบบ cross-validation (default: 1)
-m <name of file with cost matrix>	ระบุแฟ้มที่มี cost matrix
-l <name of input file>	ระบุไฟล์อินพุตของโมเดล
-d <name of output file>	ระบุไฟล์เอาต์พุตของโมเดล
-v	เอาต์พุตที่ไม่มีค่าทางสถิติของชุดข้อมูลฝึก
-o	เอาต์พุตที่มีเฉพาะค่าทางสถิติของชุดข้อมูลฝึกเท่านั้น
-i	เอาต์พุตรายละเอียดเกี่ยวกับการเข้าถึงข้อมูลสำหรับแต่ละคลาส
-k	เอาต์พุต information-theoretic statistics
-p <attribute range>	แสดงผลลัพธ์ของการทดสอบการทำนายสำหรับแอททริบิวต์ที่ระบุ (0 for none)
-r	แสดงเอาต์พุต cumulative margin distribution
-z <class name>	แสดงเอาต์พุต source representation ของ Classifier จากคลาสที่ระบุ
-g	แสดงเอาต์พุต graph representation ของ Classifier

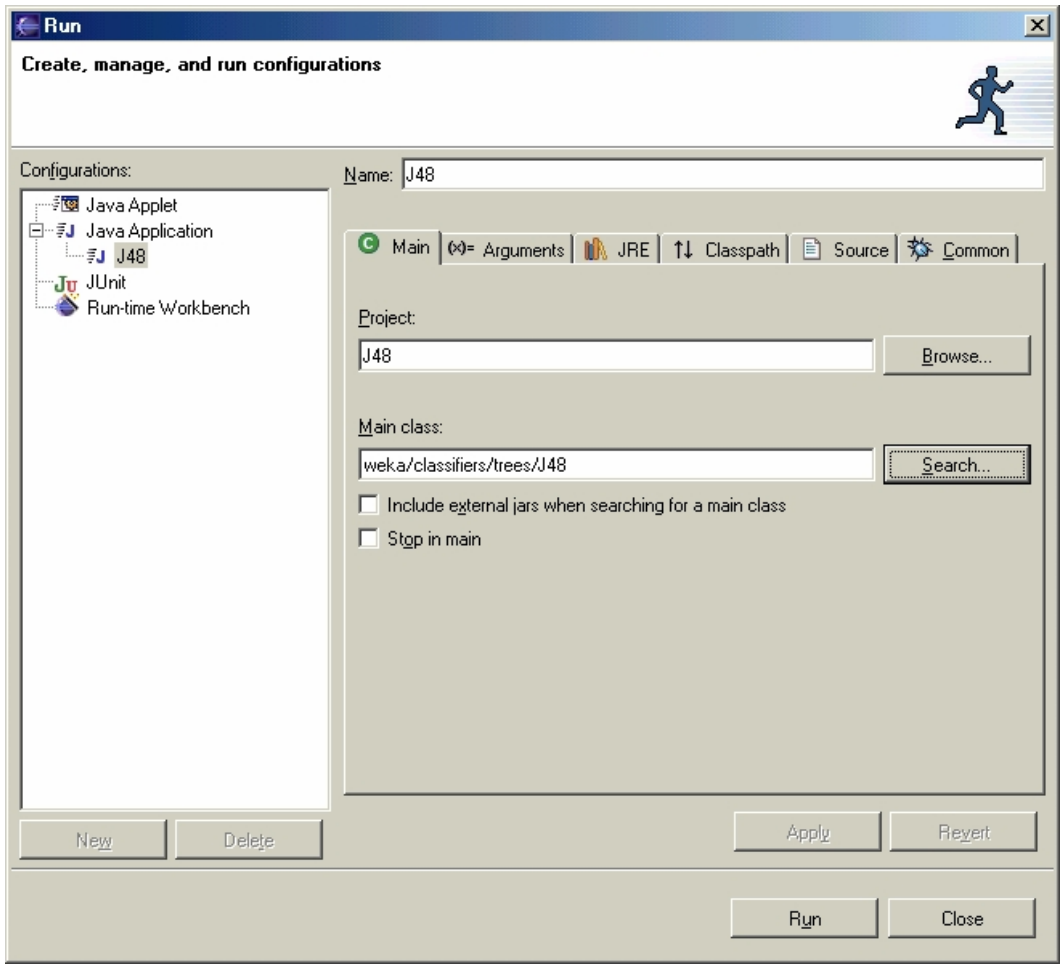
3. การใช้งานอัลกอริทึม J48 ใหม่ที่ถูกแยกออกมาจากระบบ WEKA

การประมวลผลอัลกอริทึมสามารถทำได้ทั้งในโปรแกรม Eclipse เอง และบน Command Prompt โดยใช้ *.jar file ดังมีรายละเอียดดังนี้

3.1 การประมวลผลอัลกอริทึม J48 บน โปรแกรม Eclipse

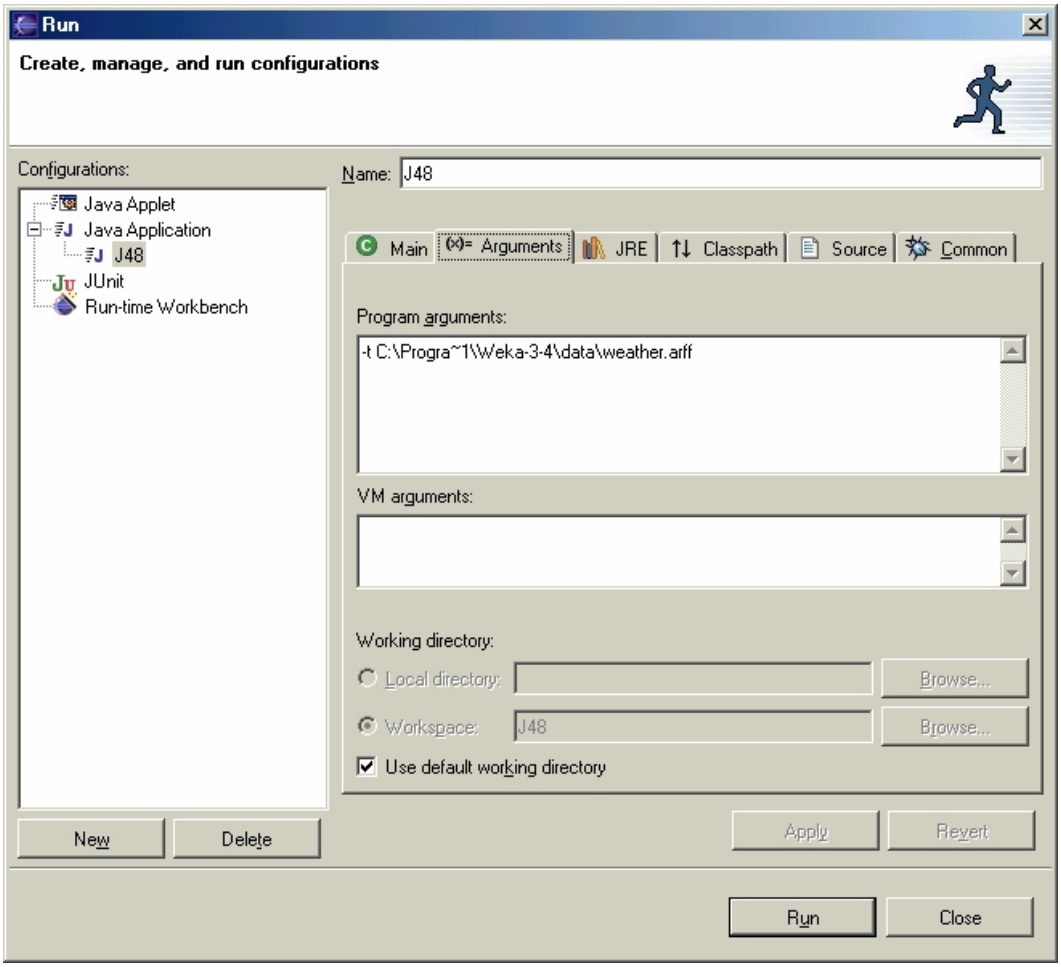
วิธีการรันอัลกอริทึม J48 โดยใช้โปรแกรม Eclipse แสดงเป็นขั้นตอนได้ดังนี้

1) เลือกเมนู Run > Run… โปรแกรมจะแสดงหน้าต่าง Run configurations คลิกเลือกที่ Java Application กำหนดชื่อเป็น J48 ในแท็บ Main ระบุ Main class ของโปรแกรมเป็น weka/classifier/trees/J48 ดังรูปที่ ข.7



รูปที่ ข.7 แสดงการกำหนดค่า Run configurations ในแท็บ Main

2) ในแท็บ Arguments กำหนดค่า Program arguments เป็น -t C:\Progra~1\Weka-3-4\data\weather.arff ซึ่งเป็นการกำหนดค่า option ให้กับอัลกอริทึม J48 โดยระบุตำแหน่งที่เก็บไฟล์ข้อมูลฝึกให้กับอัลกอริทึมเพื่อใช้ในการวิเคราะห์



รูปที่ ข.8 แสดงการกำหนดค่า Run configurations ในแท็บ Arguments

3) เมื่อกำหนดค่าต่าง ๆ ของ Run configurations คลิกที่ปุ่ม Run อัลกอริทึม J48 จะทำงานและแสดงผลลัพธ์ของการสร้างต้นไม้ตัดสินใจในหน้าต่าง console ของโปรแกรม Eclipse ซึ่งได้ผลลัพธ์เช่นเดียวกับการใช้งานผ่านระบบ WEKA แต่มีสิ่งที่ต่างกันอยู่เพียงเล็กน้อยคือ การรันด้วยวิธีนี้จะแสดงข้อมูลการทดสอบโมเดลเป็นสองส่วน โดยส่วนแรกคือการทดสอบโมเดลโดยใช้ข้อมูลฝึกเป็นตัวทดสอบ และอีกส่วนเป็นการทดสอบตามการตั้งค่าพารามิเตอร์ –T <test-set> ซึ่งถ้าไม่มีการระบุอัลกอริทึมจะใช้วิธีทดสอบเป็นแบบ cross-validation ดังแสดงผลลัพธ์ที่ได้ดังต่อไปนี้

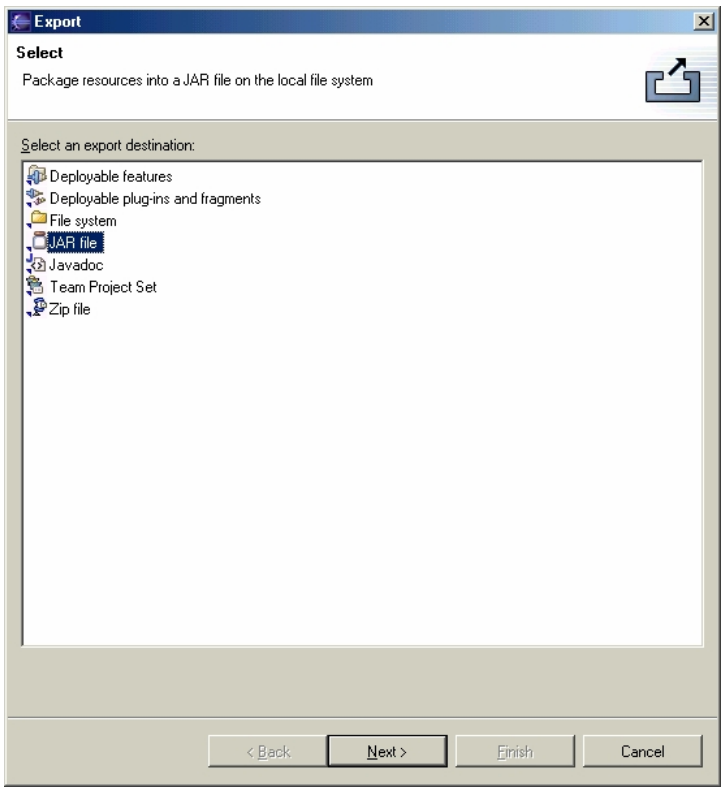
```
J48 pruned tree
------------------

outlook = sunny
|   humidity <= 75: yes (2.0)
|   humidity > 75: no (3.0)
outlook = overcast: yes (4.0)
outlook = rainy
|   windy = TRUE: no (2.0)
|   windy = FALSE: yes (3.0)

Number of Leaves  :     5
Size of the tree :      8

Time taken to build model: 0.22 seconds
Time taken to test model on training data: 0 seconds

=== Error on training data ===
Correctly Classified Instances          14               100      %
Incorrectly Classified Instances         0                 0      %
Kappa statistic                          1
Mean absolute error                      0
Root mean squared error                  0
Relative absolute error                  0      %
Root relative squared error              0      %
Total Number of Instances               14

=== Confusion Matrix ===
 a b   <-- classified as
 9 0 | a = yes
 0 5 | b = no


=== Stratified cross-validation ===
Correctly Classified Instances           9               64.2857 %
Incorrectly Classified Instances         5               35.7143 %
Kappa statistic                          0.186
Mean absolute error                      0.2857
Root mean squared error                  0.4818
Relative absolute error                 60      %
Root relative squared error             97.6586 %
Total Number of Instances               14

=== Confusion Matrix ===
 a b   <-- classified as
 7 2 | a = yes
 3 2 | b = no
```

รูปที่ ข.9 ผลลัพธ์ที่ได้จากการรันอัลกอริทึม J48 ด้วยโปรแกรม Eclipse

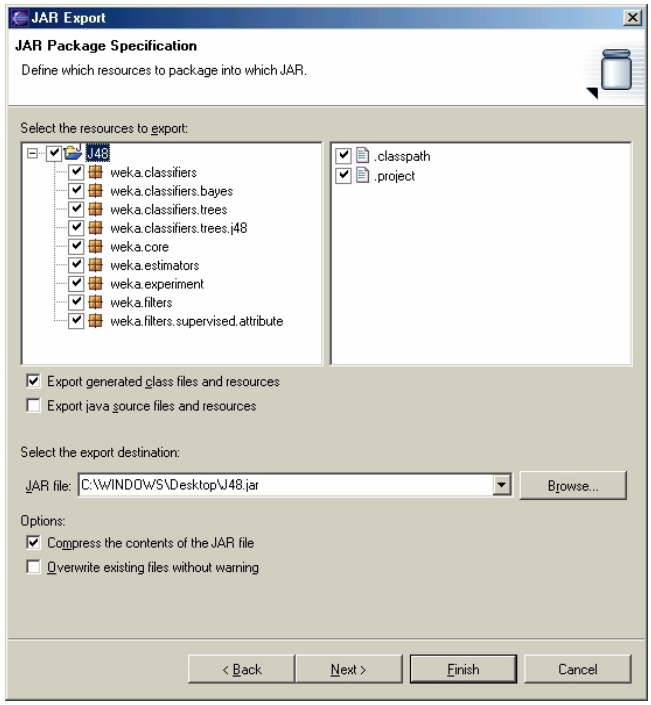
3.2 การประมวลผลอัลกอริทึม J48 บน Command Prompt

เราสามารถนำเอาคลาสต่าง ๆ ที่เกี่ยวข้องกับการทำงานของอัลกอริทึม J48 รวมเข้าไว้ด้วยกันโดยการ Export โปรเจ็คออกไปเป็น JAR file เพื่อนำไปรันบน Command Prompt โดยแสดงเป็นขั้นตอนได้ดังนี้

1) เลือกที่เมนู File > Export… ในหน้าต่าง Export คลิกเลือก Export destination เป็น JAR file ดังรูปที่ ข.10
2) ในหน้าต่าง JAR Export เลือกคลาสทั้งหมดในทุกแพ็คเกจเพื่อทำการ export จากนั้นคลิกเลือก Export generated class files and resources ระบุชื่อและตำแหน่งของ JAR file ที่ต้องการ และคลิกที่ปุ่ม Finish เป็นการเสร็จสิ้นการสร้าง JAR file จากโปรเจ็ค ดังรูปที่ ข.11



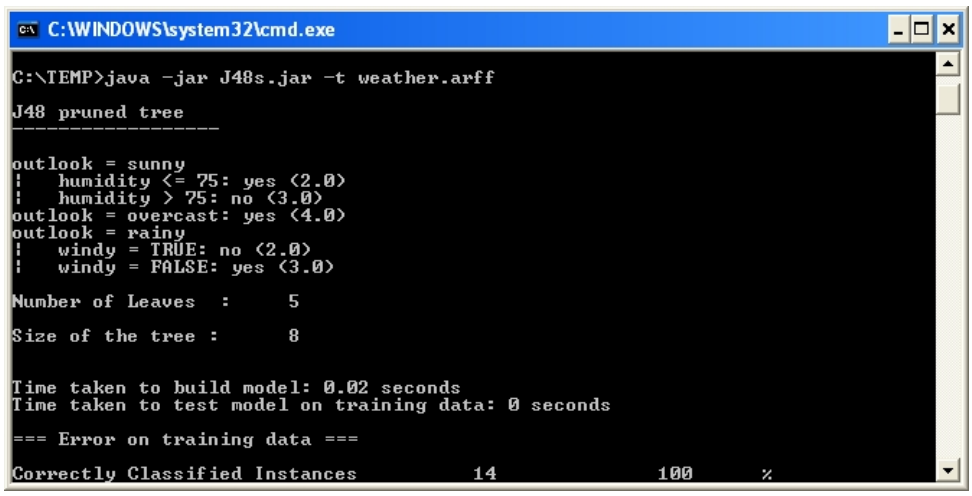
รูปที่ ข.10 การสร้าง JAR file สำหรับรันบน Command Prompt



รูปที่ ข.11 การกำหนดค่าต่าง ๆ เพื่อสร้าง JAR file

3) ทำการรัน JAR file ที่ Command Prompt โดยใช้คำสั่งดังนี้

java –jar J48.jar –t C:\Progra~1\Weka-3-4\data\weather.arff



รูปที่ ข.12 แสดงการรันอัลกอริทึม J48 ด้วย Command Prompt

ผลลัพธ์ที่ได้จากการรันอัลกอริทึม J48 ด้วย Command Prompt จะเหมือนกับการรันด้วยโปรแกรม Eclipse

```
C:\TEMP>java -jar J48.jar -t weather.arff

J48 pruned tree
------------------
outlook = sunny
|   humidity <= 75: yes (2.0)
|   humidity > 75: no (3.0)
outlook = overcast: yes (4.0)
outlook = rainy
|   windy = TRUE: no (2.0)
|   windy = FALSE: yes (3.0)

Number of Leaves  :     5
Size of the tree :      8

Time taken to build model: 0.02 seconds
Time taken to test model on training data: 0 seconds

=== Error on training data ===
Correctly Classified Instances          14               100      %
Incorrectly Classified Instances         0                 0      %
Kappa statistic                          1
Mean absolute error                      0
Root mean squared error                  0
Relative absolute error                  0      %
Root relative squared error              0      %
Total Number of Instances               14

=== Confusion Matrix ===
 a b   <-- classified as
 9 0 | a = yes
 0 5 | b = no

=== Stratified cross-validation ===
Correctly Classified Instances           9               64.2857 %
Incorrectly Classified Instances         5               35.7143 %
Kappa statistic                          0.186
Mean absolute error                      0.2857
Root mean squared error                  0.4818
Relative absolute error                 60      %
Root relative squared error             97.6586 %
Total Number of Instances               14

=== Confusion Matrix ===
 a b   <-- classified as
 7 2 | a = yes
 3 2 | b = no
```

รูปที่ ข.13 ผลลัพธ์ที่ได้จากการรันอัลกอริทึม J48 ด้วย Command Prompt

ภาคผนวก ค

ตารางแสดงค่าวิกฤตของการทดสอบไคสแควร์

ตารางที่ ค.1 ค่าวิกฤตของการทดสอบไคสแควร์ (χ^2)

ระดับความเป็นอิสระ (df)	ระดับความมีนัยสำคัญ (α)					
	0.01	0.025	0.05	0.1	0.25	0.5
1	6.635	5.024	3.841	2.706	1.323	0.455
2	9.210	7.378	5.991	4.605	2.773	1.386
3	11.345	9.348	7.815	6.251	4.108	2.366
4	13.277	11.143	9.488	7.779	5.385	3.357
5	15.086	12.833	11.071	9.236	6.626	4.351
6	16.812	14.449	12.592	10.645	7.841	5.348
7	18.475	16.013	14.067	12.017	9.037	6.346
8	20.090	17.535	15.507	13.362	10.219	7.344
9	21.666	19.023	16.919	14.684	11.389	8.343
10	23.209	20.483	18.307	15.987	12.549	9.342
11	24.725	21.920	19.675	17.275	13.701	10.341
12	26.217	23.337	21.026	18.549	14.845	11.340
13	27.688	24.736	22.362	19.812	15.984	12.340
14	29.141	26.119	23.685	21.064	17.117	13.339
15	30.578	27.488	24.996	22.307	18.245	14.339
16	32.000	28.845	26.296	23.542	19.369	15.339
17	33.409	30.191	27.587	24.769	20.489	16.338
18	34.805	31.526	28.869	25.989	21.605	17.338
19	36.191	32.852	30.144	27.204	22.718	18.338
20	37.566	34.170	31.410	28.412	23.828	19.337
21	38.932	35.479	32.671	29.615	24.935	20.337
22	40.289	36.781	33.924	30.813	26.039	21.337
23	41.638	38.076	35.172	32.007	27.141	22.337
24	42.980	39.364	36.415	33.196	28.241	23.337
25	44.314	40.646	37.652	34.382	29.339	24.337
26	45.642	41.923	38.885	35.563	30.435	25.336
27	46.963	43.195	40.113	36.741	31.528	26.336

ตารางที่ ค.1 ค่าวิกฤตของการทดสอบไคสแควร์ (χ^2) (ต่อ)

ระดับความเป็นอิสระ (df)	ระดับความมีนัยสำคัญ (α)					
	0.01	0.025	0.05	0.1	0.25	0.5
28	48.278	44.461	41.337	37.916	32.620	27.336
29	49.588	45.722	42.557	39.087	33.711	28.336
30	50.892	46.979	43.773	40.256	34.800	29.336

ภาคผนวก ง

รหัสต้นฉบับของเทคนิคการตัดกิ่งด้วยวิธี REP+

```
package narupon.trees.pruning;

import weka.core.*;
import weka.classifiers.trees.j48.*;

public class ChiSquared {

	/** Array for storing the confusion matrix. */
	private double [][] m_ConfusionMatrix;
	private float m_significance = 0.5f;  // Significance level
	private static boolean m_usePValue = false;
	private double m_chival;

	/** Array for storing the critical values of Chi-squared.
	 *  (Significance level = 0.01, 0.025, 0.05, 0,1, 0.25, and
	 *    0.5, df is between 1 and 30)
	 */
	private static final double m_Critical[][] = {
		// Significance level = 0.01
		{   6.635,  9.210, 11.345, 13.277, 15.086,
		   16.812, 18.475, 20.090, 21.666, 23.209,
		   24.725, 26.217, 27.688, 29.141, 30.578,
		   32.000, 33.409, 34.805, 36.191, 37.566,
		   38.932, 40.289, 41.638, 42.980, 44.314,
		   45.642, 46.963, 48.278, 49.588, 50.892
		},
		// Significance level = 0.025
		{   5.024,  7.378,  9.348, 11.143, 12.832,
		   14.449, 16.013, 17.535, 19.023, 20.483,
		   21.920, 23.337, 24.736, 26.119, 27.488,
		   28.845, 30.191, 31.526, 32.852, 34.170,
		   35.479, 36.781, 38.076, 39.364, 40.646,
		   41.923, 43.194, 44.461, 45.722, 46.979
		},
		// Significance level = 0.05
		{   3.841,  5.991,  7.815,  9.488, 11.070,
		   12.592, 14.067, 15.507, 16.919, 18.307,
		   19.675, 21.026, 22.362, 23.685, 24.996,
		   26.296, 27.587, 28.869, 30.144, 31.410,
		   32.671, 33.924, 35.172, 36.415, 37.652,
		   38.885, 40.113, 41.337, 42.557, 43.773
		},
		// Significance level = 0.1
		{   2.706,  4.605,  6.251,  7.779,  9.236,
		   10.645, 12.017, 13.362, 14.684, 15.987,
		   17.275, 18.549, 19.812, 21.064, 22.307,
		   23.542, 24.769, 25.989, 27.204, 28.412,
		   29.615, 30.813, 32.007, 33.196, 34.382,
		   35.563, 36.741, 37.916, 39.087, 40.256
		},
		// Significance level = 0.25
		{  1.32330,  2.77259,  4.10834,  5.38527,  6.62568,
		   7.84080,  9.03715, 10.21885, 11.38875, 12.54886,
		  13.70069, 14.84540, 15.98391, 17.11693, 18.24509,
		  19.36886, 20.48868, 21.60489, 22.71781, 23.82769,
```

```
              24.93478, 26.03927, 27.14134, 28.24115, 29.33885,
              30.43457, 31.52841, 32.62049, 33.71091, 34.79974
            },
            // Significance level = 0.5
            {  0.45494,  1.38629,  2.36597,  3.35669,  4.35146,
               5.34812,  6.34581,  7.34412,  8.34283,  9.34182,
              10.34100, 11.34032, 12.33976, 13.33927, 14.33886,
              15.33850, 16.33818, 17.33790, 18.33765, 19.33743,
              20.33723, 21.33704, 22.33688, 23.33673, 24.33659,
              25.33646, 26.33634, 27.33623, 28.33613, 29.33603
            }
};


      public ChiSquared(Distribution data, float significance) {

            int maxIndex;
            int numBags = data.numBags();
            int numClasses = data.numClasses();
            m_ConfusionMatrix = new double [numClasses][numClasses];
            m_significance = significance;

            for (int i = 0; i < numBags; i++)
              for (int j = 0; j < numClasses; j++) {
                  maxIndex = data.maxClass(i);
                  m_ConfusionMatrix[maxIndex][j] +=
                  data.perClassPerBag(i, j);
              }

      }

      public ChiSquared(Distribution train, Distribution test, float
      significance) {

            int maxIndex;
            int numBags = train.numBags();
            int numClasses = train.numClasses();
            double [][]tempMatrix = new double [numBags][numClasses];
            m_ConfusionMatrix = new double [numClasses][numClasses];
            m_significance = significance;

            for (int i = 0; i < numBags; i++)
              for (int j = 0; j < numClasses; j++) {
                  tempMatrix[i][j] = train.perClassPerBag(i, j) +
                  test.perClassPerBag(i, j);
              }

            Distribution temp = new Distribution(tempMatrix);

            for (int i = 0; i < numBags; i++)
              for (int j = 0; j < numClasses; j++) {
                  maxIndex = temp.maxClass(i);
                  m_ConfusionMatrix[maxIndex][j] +=
                  temp.perClassPerBag(i, j);
              }

      }
```

```
/**
 * Statistical tests of independence.
 */
public boolean compare() {

      int index;
      if (m_significance == 0.01f)  index = 0;
      else if (m_significance == 0.025f)  index = 1;
      else if (m_significance == 0.05f)   index = 2;
      else if (m_significance == 0.1f)    index = 3;
      else if (m_significance == 0.25f)   index = 4;
      else index = 5;

      int df = (m_ConfusionMatrix.length - 1) *
      (m_ConfusionMatrix[0].length - 1);
      m_chival = chiVal(m_ConfusionMatrix, false);

      if (df == 1 && m_usePValue) {
            if (Utils.smOrEq(m_chival, m_significance))
                  return false;
            else
                  return true;
      }

      else {
            if (Utils.gr(m_chival, 0)) {
                  if (Utils.grOrEq(m_chival,
                  m_Critical[index][df-1]))
                        return false;
                  else
                        return true;
            }
            else
                  return false;
      }

}

/**
 * Computes chi-squared statistic for a contingency table.
 */
public static double chiVal(double [][] matrix, boolean
useYates) {

  int df, nrows, ncols, row, col;
  double[] rtotal, ctotal;
  double expect = 0, chival = 0, n = 0;
  boolean yates = false;

  nrows = matrix.length;
  ncols = matrix[0].length;
  rtotal = new double [nrows];
  ctotal = new double [ncols];

  for (row = 0; row < nrows; row++) {
      for (col = 0; col < ncols; col++) {
            rtotal[row] += matrix[row][col];
            ctotal[col] += matrix[row][col];
```

```
                n += matrix[row][col];
            }
        }

        df = (nrows - 1) * (ncols - 1);

        if (df == 1) {
            yates = true;
            if (Utils.smOrEq(n, 20)) {
                m_usePValue = true;
                return pValue(matrix);
            }
        }
        else if (df <= 0) {
            m_usePValue = false;
            return 0;
        }

        m_usePValue = false;
        chival = 0.0;
        for (row = 0; row < nrows; row++) {
            if (Utils.gr(rtotal[row], 0)) {
                for (col = 0; col < ncols; col++) {
                    if (Utils.gr(ctotal[col], 0)) {
                        expect = (ctotal[col] * rtotal[row]) /
                        n;
                        chival += chiCell (matrix[row][col],
                        expect, yates);
                    }
                }
            }
        }
        return chival;
    }

    /**
     * Computes chi-value for one cell in a contingency table.
     */
    private static double chiCell(double freq, double expected,
    boolean yates){

        // Cell in empty row and column?
        if (Utils.smOrEq(expected, 0)) {
            return 0;
        }

        // Compute difference between observed and expected value
        double diff = Math.abs(freq - expected);
        if (yates) {

            // Apply Yates' correction if wanted
            diff -= 0.5;

            // The difference should never be negative
            if (diff < 0) {
              diff = 0;
            }
        }
```

```
  // Return chi-value for the cell
  return (diff * diff / expected);
}

public static double sum(double a, double b) {

      return a + b;
}

/**
 * Computes the value of a factorial function.
 */
public static double facVal(double n) {

      double n_factorial = 1;

      for(int count = 1; count <= n; count++)
            n_factorial *= count;

      return n_factorial;
}


/**
 * Computes p-value of contingency tables.
 */
public static double pValue(double [][] matrix) {

      double f11, f12, f21, f22, r1, r2, c1, c2, a=0, b=0, c=0,
      d=0;
      double ab, ac, bd, cd, abcd;
      double x1, x2, x3, x4, y1, y2, y3, y4, ra1, ra2, ra3,
      ra4, ra5;
      double p1=0, p2=0, temp=0, ratio=0, z=0, mode=0, aaron=0;
      boolean flag=false, revflag=false;

      f11 = matrix[0][0];
      f12 = matrix[0][1];
      f21 = matrix[1][0];
      f22 = matrix[1][1];

      r1=sum(f11,f12);
      r2=sum(f21,f22);
      c1=sum(f11,f21);
      c2=sum(f12,f22);

      if (r1<=r2) { if (c1<=c2) { a=f11; b=f12; c=f21;
      d=f22; } }
      if (r1<=r2) { if (c2<=c1) { a=f12; b=f22; c=f11;
      d=f21; } }
      if (r2<=r1) { if (c1<=c2) { a=f21; b=f11; c=f22;
      d=f12; } }
      if (r2<=r1) { if (c2<=c1) { a=f22; b=f21; c=f12;
      d=f11; } }
      if (b<a) { z=c; c=a; a=b; b=d; d=z; }
      else if (c<a) { z=b; b=a; a=c; c=d; d=z; }
      temp=a;
```

```
for (int i=0; i<100; i++) {
  ab=sum(a,b); ac=sum(a,c); bd=sum(b,d); cd=sum(c,d);
  abcd=sum(ab,cd);
  x1=facVal(ab); x2=facVal(ac); x3=facVal(bd);
  x4=facVal(cd);
  y1=facVal(a); y2=facVal(b); y3=facVal(c); y4=facVal(d);
  for (i=0; i<2; i++) {
      if (x2>x1) { z=x1; x1=x2; x2=z; }
      if (x3>x2) { z=x2; x2=x3; x3=z; }
      if (x4>x3) { z=x3; x3=x4; x4=z; }
  }
  for (i=0; i<2; i++) {
      if (y2>y1) { z=y1; y1=y2; y2=z; }
      if (y3>y2) { z=y2; y2=y3; y3=z; }
      if (y4>y3) { z=y3; y3=y4; y4=z; }
  }
  ra1=x1/y1; ra2=x2/y2; ra3=x3/y3; ra4=x4/y4;
  ra5=1/facVal(abcd);
  ratio=ra1*ra2*ra3*ra4*ra5;
  if (p1==0) { aaron=ratio; }
  p1=sum(p1,ratio);
  if (ratio>mode) { mode=ratio; }
  a=sum(a,-1); b=sum(b,1); c=sum(c,1); d=sum(d,-1);
  if (a<0) { break; }
}

if (mode>aaron) { p1=1-p1+aaron; revflag=true; }

if (r1<=r2) { if (c1<=c2) { a=f11; b=f12; c=f21;
d=f22; } }
if (r1<=r2) { if (c2<=c1) { a=f12; b=f22; c=f11;
d=f21; } }
if (r2<=r1) { if (c1<=c2) { a=f21; b=f11; c=f22;
d=f12; } }
if (r2<=r1) { if (c2<=c1) { a=f22; b=f21; c=f12;
d=f11; } }
if (b<a) { z=c; c=a; a=b; b=d; d=z; }
else if (c<a) { z=b; b=a; a=c; c=d; d=z; }

for (int j=0; j<100; j++) {
  if (revflag==false) { a=sum(a,1); b=sum(b,-1);
  c=sum(c,-1); d=sum(d,1); }
  else if (revflag==true) { a=sum(a,-1); b=sum(b,1);
  c=sum(c,1); d=sum(d,-1); }
  if (a<0) { break; }
  if (b<0) { break; }
  if (c<0) { break; }
  if (d<0) { break; }
  ab=sum(a,b); ac=sum(a,c); bd=sum(b,d); cd=sum(c,d);
  abcd=sum(ab,cd);
  x1=facVal(ab); x2=facVal(ac); x3=facVal(bd);
  x4=facVal(cd);
  y1=facVal(a); y2=facVal(b); y3=facVal(c); y4=facVal(d);
  for (j=0; j<2; j++) {
      if (x2>x1) { z=x1; x1=x2; x2=z; }
      if (x3>x2) { z=x2; x2=x3; x3=z; }
      if (x4>x3) { z=x3; x3=x4; x4=z; }
  }
```

```
            for (j=0; j<2; j++) {
                if (y2>y1) { z=y1; y1=y2; y2=z; }
                if (y3>y2) { z=y2; y2=y3; y3=z; }
                if (y4>y3) { z=y3; y3=y4; y4=z; }
            }
            ra1=x1/y1; ra2=x2/y2; ra3=x3/y3; ra4=x4/y4;
            ra5=1/facVal(abcd);
            ratio=ra1*ra2*ra3*ra4*ra5;
            if (ratio<=(aaron+.000000001)) { flag=true; }
            if (flag==true) { p2=sum(p2,ratio); }
          }

          return p1+p2;
      }

      /**
       * Main method for testing this class.
       */
      public static void main(String[] args) {

          double[] firstRow = {4, 2};
          double[] secondRow = {3, 6};
          double[][] m_ConfusionMatrix = new double[2][0];

          m_ConfusionMatrix[0] = firstRow; m_ConfusionMatrix[1] =
          secondRow;

          for (int i = 0; i < m_ConfusionMatrix.length; i++) {
            for (int j = 0; j < m_ConfusionMatrix[i].length; j++) {
                System.out.print(m_ConfusionMatrix[i][j] + " ");
            }
            System.out.println();
          }

          System.out.println("Chi-squared value: " +
          chiVal(m_ConfusionMatrix, false));

          pValue(m_ConfusionMatrix);
      }
}
```

ประวัติผู้เขียน

นายนฤพนต์ ว่องประชานุกูล เกิดเมื่อวันที่ 25 มีนาคม พ.ศ. 2521 เกิดที่อำเภอดำเนินสะดวก จังหวัดราชบุรี สำเร็จการศึกษาระดับปริญญาตรีสาขาวิชาวิศวกรรมคอมพิวเตอร์ จากมหาวิทยาลัยเทคโนโลยีสุรนารี จังหวัดนครราชสีมา เมื่อปี พ.ศ. 2543 ภายหลังสำเร็จการศึกษาได้เข้าทำงานในศูนย์คอมพิวเตอร์ของมหาวิทยาลัย สังกัดฝ่ายประมวลผลข้อมูลด้วยคอมพิวเตอร์ ต่อมาในปี พ.ศ. 2544 ได้เข้าทำงานในศูนย์ปฏิบัติการวิจัยเครื่องกำเนิดแสงซินโครตรอนแห่งชาติ หน่วยงานในสังกัดกระทรวงวิทยาศาสตร์และเทคโนโลยี ทำหน้าที่พัฒนาและดูแลระบบควบคุมด้วยคอมพิวเตอร์ของฝ่ายเทคโนโลยีเครื่องเร่งอนุภาค จากการทำงานด้านวิศวกรรมจึงเป็นแรงจูงใจที่จะศึกษาต่อในระดับปริญญาโท โดยได้ขอลาศึกษาต่อในสาขาวิชาวิศวกรรมคอมพิวเตอร์ สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี ในปีการศึกษา 2547 โดยในระหว่างการศึกษาได้ช่วยงานวิจัยของคณาจารย์ในสาขาวิชา และเป็นผู้สอนปฏิบัติการในรายวิชา Computer Programming และ Event-Driven Programming