# การพัฒนาระบบเหมืองข้อมูลแบบ Classification โดยใช้เทคนิค C4.5 Decision Tree

# DATA MINING DEVELOPMENT FOR CLASSIFICATION USING C4.5 DECISION TREE



โดย

ปริญา คำเกลี้ยง

PARIYA KHAMKHLIANG

อาจารย์ที่ปรึกษา

รศ. ดร. วรพจน์ กรีสุระเดช



รายงานนี้เป็นส่วนหนึ่งของวิชาการศึกษาอิสระ
หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
คณะเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ภาคเรียนที่ 2 ปีการศึกษา 2553

# DATA MINING DEVELOPMENT FOR CLASSIFICATION USING C4.5 DECISION TREE

PARIYA KHAMKHLIANG

A REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS OF THE COURSE
INDEPENDENT STUDY
MASTER OF SCIENCE PROGRAM IN INFORMATION TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
2/ 2010

| | |
|---|---|
| **หัวข้อ** | การพัฒนาระบบเหมืองข้อมูลแบบ Classification โดยใช้เทคนิค C4.5 Decision Tree |
| **นักศึกษา** | นางสาว ปริญา คำเกลี้ยง |
| **รหัสนักศึกษา** | 51066538 |
| **ปริญญา** | วิทยาศาสตรมหาบัณฑิต |
| **สาขาวิชา** | เทคโนโลยีสารสนเทศ |
| **แขนงวิชา** | วิทยาการสารสนเทศ |
| **ปีการศึกษา** | 2553 |
| **อาจารย์ที่ปรึกษา** | รศ.ดร.วรพจน์ กรีสุระเดช |

## บทคัดย่อ

การศึกษาอิสระฉบับนี้ถูกจัดทำขึ้นเพื่อศึกษาและทำความเข้าใจเกี่ยวกับการพัฒนาเหมืองข้อมูล โดยใช้แนวคิดแบบ Classification โดยนำหลักการของ C4.5 Decision Tree เข้ามาดำเนินการเพื่อประโยชน์สำหรับเป็นแนวทางในการวิเคราะห์ และพัฒนาเหมืองข้อมูลต่อไป

| | |
|---|---|
| **Title** | Data mining Development for Classification using C4.5 Decision Tree |
| **Student** | Miss. Pariya KhamKhliang |
| **Student ID.** | 51066538 |
| **Degree** | Master of Science |
| **Program** | Information Technology |
| **Major** | Information Technology Management |
| **Academic Year** | 2010 |
| **Advisor** | Assoc.Prof. Dr. Worapoj Kreesuradej |


## ABSTRACT

This Independent study is intended for education and understanding about the development of data mining using the principles of the C4.5 Decision Tree of Classification concept. For a way to benefit analysis and to develop data mining.

# กิตติกรรมประกาศ

การพัฒนาระบบงานฉบับนี้สามารถสำเร็จลุล่วงได้เป็นอย่างดี ด้วยคำแนะนำ และการให้คำปรึกษาจาก รศ.ดร.วรพจน์ กรีสุระเดช ซึ่งเป็นอาจารย์ที่ปรึกษา และผู้รับผิดชอบในโครงการพัฒนาระบบงานนี้ ข้าพเจ้ารู้สึกทราบซึ้งในความอนุเคราะห์จากท่านอาจารย์ และขอขอบพระคุณเป็นอย่างสูง ที่ท่านกรุณาสละเวลาให้คำแนะนำ และเสนอแนวคิดต่างๆ อันเกิดประโยชน์แก่ข้าพเจ้า และเป็นที่ปรึกษาในการแก้ปัญหาต่างๆที่เกิดขึ้นจนผ่านพ้นไปได้ด้วยดี สำเร็จตามวัตถุประสงค์ดังที่ตั้งใจไว้ รวมทั้งยังเป็นผู้ตรวจสอบความถูกต้องของเนื้อหาให้สมบูรณ์ยิ่งขึ้น

ขอขอบพระคุณคณาจารย์ สาขาวิชาเทคโนโลยีสารสนเทศ คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุก ๆ ท่านที่ได้ประสิทธิ์ประสาทวิชาให้กับข้าพเจ้า

ขอขอบคุณเจ้าหน้าที่ ในคณะเทคโนโลยีสารสนเทศ ที่ให้ความช่วยเหลือ และคำแนะนำในเรื่องต่างๆ

สุดท้ายนี้ข้าพเจ้าขอขอบพระคุณ บิดา มารดา และครอบครัวของข้าพเจ้าที่เป็นกำลังใจ และให้การสนับสนุนในทุกๆเรื่อง ทำให้ข้าพเจ้าสามารถทำโครงการพัฒนาระบบงานฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี

คุณค่าและประโยชน์อันพึงมาจากโครงการพัฒนาระบบฉบับนี้ ข้าพเจ้าขอมอบแด่ผู้มีพระคุณทุกท่าน

ปริญา คำเกลี้ยง

# สารบัญ

หน้า

# สารบัญ (ต่อ)

หน้า

# สารบัญตาราง

# สารบัญภาพ

# สารบัญภาพ (ต่อ)

# สารบัญภาพ (ต่อ)

ภาพที่ หน้า

# บทที่ 1
# บทนำ

## 1.1 ความเป็นมาและความสำคัญของปัญหา

เนื่องจากภาคธุรกิจและหน่วยงานต่างๆได้มีการจัดเก็บข้อมูลที่เกี่ยวข้องกับกิจกรรม หรือพฤติกรรมของลูกค้าเอาไว้ในปริมาณมาก และในสภาวะเศรษฐกิจปัจจุบันที่มีความผันผวนและมีการแข่งขันกันสูง หากสามารถสกัดหรือดึงเอาสารสนเทศหรือข้อความรู้ที่ซ่อนอยู่ภายในของข้อมูลที่มีขนาดใหญ่มากเหล่านี้มาช่วยในการตัดสินใจในการดำเนินงานของหน่วยงานได้ ก็จะทำให้ผู้บริหารสามารถบริหารงานได้ดียิ่งขึ้น การเข้าใจถึงพฤติกรรมของข้อมูลเหล่านั้น และสามารถนำไปใช้ได้อย่างถูกต้อง จะสามารถสร้างโอกาสให้กับธุรกิจมากขึ้น ส่งผลให้หน่วยงานมีความได้เปรียบในการวางแผน และเชิงการแข่งขัน เพื่อที่จะสามารถครองส่วนแบ่งทางการตลาดและทำกำไรได้สูงสุด รวมถึงการสร้างความพึงพอใจสูงสุดให้กับลูกค้า, การพยายามรักษาฐานลูกค้าเก่าและเพิ่มกลุ่มลูกค้าใหม่ ซึ่งเป็นสิ่งที่ทุกธุรกิจขาดไม่ได้

จากการที่ธุรกิจเหล่านี้มีข้อมูลที่ถูกจัดเก็บไว้ในระบบเป็นปริมาณมาก และต้องการวิธีการที่จะช่วยทำการค้นหารูปแบบ หรือข้อความรู้ที่ซ่อนอยู่ในฐานข้อมูลขนาดใหญ่ที่ซับซ้อนนั้นได้อย่างสะดวกและมีประสิทธิภาพ โดยใช้เวลาและค่าใช้จ่ายที่ไม่สูงจนเกินไป จึงได้มีความพยายามที่จะคิดค้นและพัฒนาเทคนิคต่างๆขึ้น เพื่อใช้ช่วยในการตัดสินใจทางธุรกิจ เข้ามาช่วยในการวิเคราะห์ข้อมูลเพื่อให้ทราบถึงความสัมพันธ์ในรูปแบบต่างๆที่ซ่อนอยู่ในฐานข้อมูล เพื่อให้ธุรกิจมีความได้เปรียบ สามารถแข่งขันได้ในตลาด ดังนั้นการทำเหมืองข้อมูลจึงเป็นสิ่งจำเป็น เนื่องจากเป็นวิธีการที่ช่วยตอบสนองความต้องการนี้ได้เป็นอย่างดียิ่ง ซึ่งวิธีการสอบถามข้อมูล และวิธีการวิเคราะห์เชิงสถิติโดยทั่วไปอาจไม่สามารถตอบสนองได้ในลักษณะเดียวกัน เช่น การใช้รูปแบบพฤติกรรมของลูกค้าที่วิเคราะห์ได้ไปใช้ในการวางแผนการตลาด, การวางมาตรการลดความเสี่ยง, การลดต้นทุนการดำเนินงานหรือเพิ่มโอกาสการแสวงหากำไร, การค้นหาข้อมูลเพื่อตอบคำถามว่าลูกค้าใดในกลุ่มลูกค้าของกิจการมีโอกาสสูงสุดที่จะซื้อสินค้าที่ออกใหม่ หรือลูกค้าคนใดของกิจการมีโอกาสสูงในการเลิกเป็นลูกค้า หรือการเบิกเงินรายการใดเป็นการเบิกที่น่าจะเป็นการทุจริต เป็นต้น

## 1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

โครงการศึกษาและพัฒนาระบบเหมืองข้อมูลแบบ Classification โดยใช้เทคนิค C4.5 มีวัตถุประสงค์ดังต่อไปนี้

1. เพื่อศึกษาการทำงานระบบเหมืองข้อมูลแบบ Classification โดยใช้เทคนิค C4.5 Decision Tree ที่มีวิธีการตัดกิ่งแบบ EBP

2. เพื่อศึกษาความเป็นไปได้ในการนำระบบสารสนเทศมาใช้ในการทำนายผลข้อมูล และรูปแบบของการประยุกต์ใช้ที่เหมาะสม

3. เพื่อทำการออกแบบระบบ พร้อมทั้งทำการพัฒนาต้นแบบหน้าจอที่สำคัญ

4. เพื่อพัฒนาระบบเหมืองข้อมูลแบบ Classification โดยใช้เทคนิค C4.5 โดยให้ระบบทำการแสดงผลโมเดลต้นไม้ที่ได้จากการทำดาต้าไมนิ่งในลักษณะของรูปภาพ

5. เพื่อพัฒนาระบบให้สามารถบันทึก หรืออ่านโมเดลต้นไม้ในรูปแบบของเอกสาร PMML โดยใช้เทคนิค TreeModel ในการอธิบายโมเดลต้นไม้ที่ได้จากการทำดาต้าไมนิ่ง และมีวิธีการจัดการแบบ nullPrediction เมื่อการประเมินผลของ Node มีค่าเป็น UNKNOWN ในระหว่างทำนายผลข้อมูล

6. เพื่อนำเอาเทคนิคของดาต้าไมนิ่ง C4.5 Decision tree มาใช้ในการวิเคราะห์ลักษณะของข้อมูล เพื่อให้สามารถนำสารสนเทศที่ได้ไปใช้ประกอบการตัดสินใจในการดำเนินงานต่างๆ รวมทั้งเป็นแนวทางในการนำไปประยุกต์ใช้ เพื่อพิจารณาปัจจัยอื่นๆที่ต้องการ

## 1.3 ขอบเขตของการศึกษา

โครงการนี้เป็นการศึกษาถึงการนำเอาเทคนิคของดาต้าไมนิ่งมาประยุกต์ใช้ โดยอาศัยหลักการของอัลกอริทึม C4.5 ซึ่งเป็นอัลกอริทึมหนึ่งใน Classification ในการแบ่งกลุ่มข้อมูล โดยอาศัยหลักการนำเสนอผลลัพธ์ในรูปของกฎ และ Decision Tree ในลักษณะของรูปภาพ เพื่อให้ง่ายต่อการวิเคราะห์ และทำความเข้าใจลักษณะของข้อมูลในกลุ่ม รวมไปถึงความแม่นยำของโมเดลต้นไม้ในการทำนายผลข้อมูล และมีการจัดเก็บโมเดลต้นไม้เป็นเอกสาร PMML ซึ่งเอกสาร PMML ที่ได้นั้น สามารถนำไปใช้กับระบบอื่นที่รองรับเอกสาร PMML ได้ โดยไม่จำเป็นต้องใช้กับระบบของโครงการพัฒนาระบบฉบับนี้เท่านั้น ส่งผลให้เกิดความยืดหยุ่นในการนำโมเดลไปใช้งานมากขึ้น

## 1.4 ขั้นตอนของการศึกษา

1. ศึกษาขั้นตอน และวิธีการพัฒนาเหมืองข้อมูล โดยอาศัยเทคนิค C4.5 และศึกษาโครงสร้างเอกสาร PMML ในการอธิบายแบบจำลอง โดยเลือกศึกษาในส่วนของเทคนิค TreeModel เท่านั้น

2. วิเคราะห์และออกแบบขั้นตอนการทำงานของระบบ และส่วนต่อประสานผู้ใช้

3. พัฒนาโปรแกรมระบบที่ใช้ในการสร้างโมเดล โดยเทคนิค C4.5 จากนั้นแสดงผลโมเดลเป็นรูปภาพ และจัดเก็บโมเดลในลักษณะเอกสาร PMML

4. ทดสอบระบบกับข้อมูลที่ได้รับมา และแก้ไข ประเมินและสรุปผลการทดสอบ
5. จัดทำเอกสารประกอบโครงงาน

## 1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. เข้าใจหลักการ และวิธีการของการทำดาต้าไมนิ่งโดยอาศัยเทคนิค C4.5
2. เป็นแนวทางในการพัฒนาต่อไปในอนาคต เช่น การเพิ่มความสามารถของฟังก์ชันให้มีประสิทธิภาพมากยิ่งขึ้น หรือเป็นแนวทางในการนำไปสร้างให้เกิดระบบใหม่ ๆ ต่อไป
3. ได้โปรแกรมระบบ ที่มีความสามารถในการทำนายผลข้อมูลที่ไม่เคยพบเห็นมาก่อนได้อย่างถูกต้อง ภายใต้จำนวน Training Data ที่มีมากพอสำหรับใช้สร้างโมเดลต้นไม้การตัดสินใจเพื่อให้ต้นไม้สามารถดำเนินการทำนายผลได้ทุกกรณี
4. ได้โปรแกรมระบบที่มีความสามารถในการนำเสนอผลลัพธ์ด้วยแผนภาพต้นไม้ เพื่อจะช่วยให้ผู้ใช้สามารถเข้าใจความหมายของกฎได้ง่าย และดียิ่งขึ้น เกิดความผิดพลาดในวิเคราะห์น้อยลง และสามารถตัดสินใจนำสารสนเทศที่ได้จากการทำนาย หรือสารสนเทศที่ซ่อนอยู่ไปประยุกต์ใช้ให้เกิดประโยชน์ต่อไปได้อย่างรวดเร็วยิ่งขึ้น ส่งผลให้เกิดการสร้างโอกาส หรือข้อได้เปรียบกับเจ้าของข้อมูลเอง
5. ได้โปรแกรมที่มีความสามารถรองรับการต่อฐานข้อมูลได้มากกว่า 1 ฐานข้อมูล ทำให้ผู้ใช้สามารถดึงข้อมูลได้สะดวกมากยิ่งขึ้น หากมีการจัดเก็บข้อมูลที่มีลักษณะโครงสร้างแบบเดียวกัน แต่ถูกเก็บอยู่ในฐานข้อมูลต่างชนิดกัน
6. โมเดลต้นไม้ที่บันทึกไว้หลังจากทำดาต้าไมนิ่ง ซึ่งอยู่ในรูปของเอกสาร PMML สามารถนำไปใช้ในระบบอื่นๆที่รองรับเอกสาร PMML ที่ใช้เทคนิค TreeModel และมีวิธีการจัดการแบบ nullPrediction ได้

# บทที่ 2
# ดาต้าไมนิ่ง (Data Mining)

ดาต้าไมนิ่งเป็นกระบวนการที่กระทำกับข้อมูลขนาดใหญ่เพื่อค้นหาสารสนเทศ รูปแบบ กฎเกณฑ์ แนวโน้ม พฤติกรรม และความสัมพันธ์ที่ซ่อนอยู่ในชุดข้อมูลซึ่งมีอยู่จริงในฐานข้อมูลขนาดใหญ่ ซึ่งสาระเหล่านี้อาจไม่เคยถูกค้นพบมาก่อน เนื่องจากปริมาณข้อมูลที่มีอยู่อย่างมากมายทำให้สาระ เหล่านี้ไม่แสดงความสำคัญหรือความโดดเด่นออกมา ดาต้าไมนิ่งเป็นการประมวลผลข้อมูลในเชิงวิเคราะห์ขั้นสูงจากฐานข้อมูล เพื่อนำข้อความรู้ที่ได้ไปใช้ประโยชน์ในการตัดสินใจ ช่วยในการค้นหรือแสวงหาโอกาสทางธุรกิจใหม่ และสามารถปรับเปลี่ยนกลยุทธ์ในการดำเนินธุรกิจได้ทันต่อสถานการณ์การแข่งขันและการเปลี่ยนแปลงได้อย่างมีประสิทธิภาพยิ่งขึ้น ในการทำดาต้าไมนิ่งนั้นจะอาศัยหลักสถิติ การรู้จำ การเรียนรู้ผ่านระบบคอมพิวเตอร์ และหลักคณิตศาสตร์ เพื่อให้ได้ข้อมูลสารสนเทศที่เราไม่รู้ออกมา และให้อยู่ในรูปแบบที่เต็มไปด้วยความหมาย และอยู่ในรูปของกฎ สารสนเทศที่ได้อาจถูกนำมาใช้ในการสร้างการพยากรณ์ หรือสร้างตัวแบบสำหรับการจำแนกหน่วยหรือกลุ่ม หรือค้นหาความสัมพันธ์ที่มีอยู่ในฐานข้อมูล หรือให้ข้อสรุปของสาระในฐานข้อมูล ซึ่งตัวอย่างการประยุกต์ใช้ดาต้าไมน์นิ่งกับงานด้านต่างๆ ได้แก่

- งานด้านการตลาด เช่น การทำโปรโมชั่นส่งเสริมการขาย, การโฆษณาสินค้าได้อย่างเหมาะสมและตรงตามเป้าหมาย
- กลุ่มประกันชีวิต เช่น แผนประกันชีวิตแบบต่างๆ
- งานด้านการแพทย์ เช่น การหาสาเหตุความผิดปกติที่ทำให้เกิดโรค, การวินิจฉัยโรคและการรักษา
- กลุ่มธุรกิจการเงินการธนาคาร เช่น ระบบตรวจจับการทุจริตทางการเงิน, การวิเคราะห์การให้สินเชื่อแก่ลูกค้า, การทำนายอัตราการจ่ายเงินกู้, การแบ่งกลุ่มลูกค้าเพื่อหาเป้าหมายทางการตลาด
- ธุรกิจค้าปลีก ในการพิจารณาหากลยุทธ์ให้เป็นที่สนใจกับผู้บริโภคในรูปแบบต่าง ๆ เช่น ที่ว่างในชั้นวางของจะจัดการอย่างไรถึงจะเพิ่มยอดขายได้
- การวิเคราะห์บัตรเครดิต เช่น การตัดสินใจในการที่จะให้เครดิตการ์ดกับลูกค้าหรือไม่, ป้องกันปัญหาเรื่องการทุจริตบัตรเครดิต

## 2.1 หน้าที่ของ Data Mining

โดยทั่วไปแล้วสามารถแบ่งกลุ่มได้ 2 ประเภทใหญ่ๆ คือ supervised และ unsupervised

1.) Supervised functions จะถูกนำมาใช้ในการทำนายค่า และต้องการการระบุผลลัพธ์การเรียนรู้ไว้ล่วงหน้าสำหรับแต่ละกรณี เพื่อที่จะใช้ในระหว่างการสร้างโมเดล ตัวอย่างเช่น ซื้อ/ไม่ซื้อ สำเร็จ/ล้มเหลว

2.) Unsupervised functions ไม่ต้องการการระบุผลลัพธ์การเรียนรู้ แต่จะค้นหาโครงสร้าง หรือความสัมพันธ์ที่มีอยู่ในตัวข้อมูล

ในอีกมุมมอง ผลที่ได้จากการทำดาต้าไมนิ่งมี 2 แบบ คือ descriptive หรือ predictive

1.) Descriptive data mining เป็นการอธิบายข้อมูล และแสดงให้เห็นถึงลักษณะที่น่าสนใจของข้อมูล

2.) Predictive data mining คือ การสร้างโมเดลที่สามารถดำเนินการสรุปผลข้อมูลที่มีอยู่ได้ แล้วพยายามที่จะนำโมเดลนั้นมาทำนายผลให้กับข้อมูลชุดใหม่

## 2.2 งานของ Data Mining

1. Association rule Discovery หลักการทำงาน คือ การค้นหาหรือแสดงความสัมพันธ์ของข้อมูลที่เกิดขึ้นจากข้อมูลขนาดใหญ่ที่มีอยู่ เพื่อนำไปใช้ในการวิเคราะห์ หรือทำนายปรากฏการณ์ต่าง ๆ เช่น การวิเคราะห์ข้อมูลการขายสินค้า โดยเก็บข้อมูลจากระบบ ณ จุดขาย หรือระบบขายสินค้าออนไลน์ แล้วทำการพิจารณาสินค้าที่อยู่ในตระกร้าเดียวกัน หรือสินค้าที่ลูกค้ามักซื้อพร้อมกัน เช่น พบว่าลูกค้าที่ซื้อเทปวิดีโอมักจะซื้อเทปกาวด้วย ทางร้านค้าก็อาจจะดำเนินการจัดร้าน โดยให้สินค้าสองอย่างนั้นวางอยู่ใกล้กันเพื่อเพิ่มยอดขาย หรืออาจจะพบว่าลูกค้าที่ซื้อหนังสือ A แล้ว หลังจากนั้นก็มักจะซื้อหนังสือ B ก็สามารถนำกฎนี้ไปแนะนำลูกค้าที่กำลังซื้อหนังสือ A ได้

2. Classification & Prediction

Classification การจัดหมวดหมู่ถือว่าเป็นงานธรรมดาทั่วไปของการทำดาต้าไมนิ่ง เป็นกระบวนการสร้างโมเดลจัดการข้อมูลให้อยู่ในกลุ่มที่กำหนดมาให้ ตัวอย่างเช่น จัดกลุ่มนักเรียนว่า ดีมาก ดี ปานกลาง ไม่ดี โดยพิจารณาจากประวัติและผลการรียน หรือแบ่งประเภทของลูกค้าว่ามีความเชื่อถือได้หรือไม่ โดยพิจารณาจากข้อมูลที่มีอยู่ โดยการจัดหมวดหมู่ ประกอบด้วย การสำรวจจุดเด่นของวัตถุที่ปรากฏออกมา และทำการกำหนดจุดเด่นนั้นๆ เป็นตัวที่ใช้แบ่งหมวดหมู่ งานในการแบ่งหมวดหมู่ คือการบ่งบอกลักษณะ โดยการอธิบายจุดเด่นที่เป็นที่รู้จักดีในหมวดหมู่นั้น และTraining Set ของตัวอย่างในแต่ละหมวดหมู่ ซึ่งมีภาระหน้าที่ในการสร้างโมเดลของบางชนิดที่ไม่สามารถจะจัดหมวดหมู่ของข้อมูลได้ ให้สามารถจัดเป็นหมวดหมู่ได้

Prediction การทำนายล่วงหน้าก็เป็นงานที่มีลักษณะคล้ายกับการจัดหมวดหมู่ แต่จะใช้สถิติการบันทึกของการจัดหมวดหมู่ในการทำนายอนาคตของพฤติกรรม หรือการประเมินค่าที่จะเกิดขึ้นในอนาคต ตัวอย่างของงานการทำนายล่วงหน้า เช่น การทำนายการเปลี่ยนแปลงพฤติกรรมของตลาด หรือการทำนายจำนวนลูกค้าที่จะออกจากธุรกิจของเราใน 6 เดือนข้างหน้า เป็นต้น

3. Database Clustering Or Segmentation เป็นเทคนิคการแบ่งข้อมูลที่มีลักษณะคล้ายกันออกเป็นกลุ่ม ด้วยการรวมกลุ่มตัวแปรที่มีลักษณะเดียวกันไว้ด้วยกัน เพื่อนำข้อมูลที่ได้ไปวิเคราะห์ คืองานที่ทำการรวมส่วนต่างๆ ในแต่ละส่วนที่ต่างชนิดกันให้อยู่รวมกันเป็นกลุ่มย่อย เช่น การแบ่งกลุ่มผู้ป่วยที่เป็นโรคเดียวกันตามลักษณะ อาการ เพื่อนำไปใช้ประโยชน์ในการวิเคราะห์หาสาเหตุของโรค โดยพิจารณาจากผู้ป่วยที่มีอากการคล้ายคลึงกัน ซึ่งความแตกต่างของการรวมตัวจากการจัดหมวดหมู่ คือ การรวมตัวจะไม่พึ่งพาอาศัยการกำหนดหมวดหมู่ล่วงหน้าและไม่ใช้ตัวอย่าง ข้อมูลจะรวมตัวกันบนพื้นฐานของความคล้ายในตัวเอง

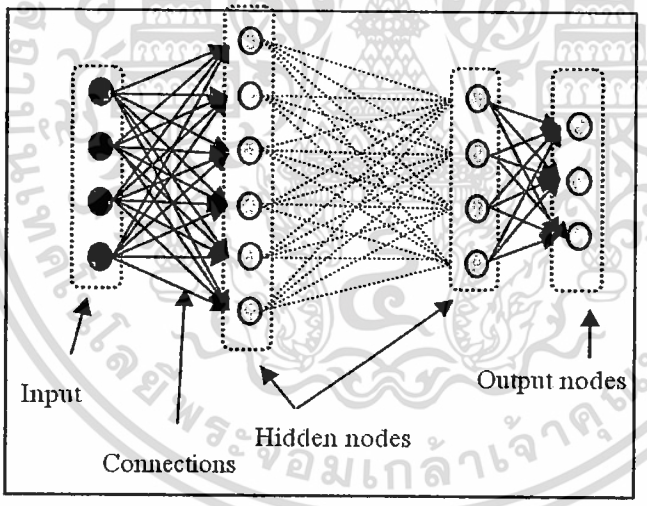
Clustering วิธีนี้เป็นวิธีที่อาจจะเรียกได้ว่าเป็นการทำดาต้าไมนิ่งแบบอ้อมๆ เนื่องจากการหาผลลัพธ์ในแต่ละครั้งนั้น ตัวผู้หาเองก็ยังไม่อาจทราบได้ว่าสิ่งที่ต้องการจะหาคืออะไร ต้องรอจนกว่าการค้นหาจะดำเนินการจนเสร็จสมบูรณ์จึงจะทราบสาระข้อมูล หรือลักษณะเด่นที่ซ่อนเร้นอยู่ภายในข้อมูล

4. Deviation Detection เป็นกรรมวิธีในการหาค่าที่แตกต่างไปจากมาตรฐาน หรือค่าที่คาดคิดไว้ ว่าต่างไปมากน้อยเพียงใด โดยทั่วไปมักใช้วิธีทางสถิติ หรือการแสดงให้เห็นภาพ (Visualization) ตัวอย่างการนำเทคนิคนี้ไปใช้ เช่น การตรวจสอบลายเซ็นปลอม, บัตรเครดิตปลอม, การหาจุดบกพร่องของชิ้นงานในโรงงานอุตสาหกรรม

## 2.3 เทคนิคของ Data Mining ที่ถูกใช้อย่างแพร่หลาย

1.) Decision Tree เป็นแบบจำลองที่มีลักษณะคล้ายกับต้นไม้ จะมีการสร้างกฎต่างๆ ขึ้นเพื่อใช้ในการตัดสินใจ ดีซีชันทรีเป็นวิธีที่ได้รับความนิยม เนื่องจากเป็นลักษณะที่คนจำนวนมากคุ้นเคย และสามารถทำความเข้าใจได้ง่าย เพราะความไม่ซับซ้อนของอัลกอริทึม ส่งผลให้เครื่องมือที่ใช้ในการทำดาต้าไมนิ่งที่วางขายกันอยู่ในท้องตลาดต่างก็ใช้วิธีนี้ โดยที่แต่ละโหนดจะแสดงถึงแอตทริบิวต์ (attribute), แต่ละกิ่ง (branch) แสดงถึงผลลัพธ์ที่ได้จากการทดสอบ และแต่ละลีฟโหนด (leaf node) แสดงถึงกลุ่มข้อมูล (class) ที่กำหนดไว้ ข้อดีของวิธีนี้คือ สามารถตีความและเข้าใจลักษณะของรูปแบบข้อมูลได้ง่าย เพราะมีการแยกออกเป็นกฎ หรือข้อกำหนดต่างๆ แต่ก็ยังคงมีปัญหาในเรื่องของการให้น้ำหนักความน่าเชื่อถือ หรือการให้ค่าน้ำหนักในแต่ละโหนด ซึ่งถ้าให้น้ำหนักผิดไป อาจจะทำให้การตีความผิดไปได้ ตัวอย่างของวิธีการ คือ Classification and Regression Trees (CART), Chi Square Automatic Interaction Detection (CHAID), C4.5

2.) Neural Network เครือข่ายเส้นประสาท เป็นแบบจำลองระบบการประมวลผลที่เหมือนเครือข่ายเส้นใยประสาทของสมองมนุษย์ที่เชื่อมโยงกัน เป็นเทคโนโลยีที่มีที่มา จากงานวิจัยด้านปัญญาประดิษฐ์ Artificial Intelligence(AI) และถูกพัฒนาขึ้นโดยโมเดลทางคณิตศาสตร์ของกระบวนการเรียนรู้ของมนุษย์ โดยเลียนแบบการทำงานของสมอง และจะเรียนรู้จากชุดข้อมูลของชุดความรู้เทรนนิ่งเซต (training set) เพื่อใช้ในการคำนวณค่าฟังก์ชันจากกลุ่มข้อมูล นิวรอลเน็ตเวิร์กประกอบด้วยหน่วยความจำจำนวนมากเรียกว่า นิวรอน (Neurons), เซล (Cells) หรือโหนด (Nodes) แต่ละนิวรอนต่อกันโดยคอนเน็กชั่นลิงค์ (Connection Link) ที่มีค่าน้ำหนักของมันอยู่ในแต่ละการเชื่อมต่อ โดยค่าน้ำหนักจะแสดงรายละเอียดที่เน็ตเวิร์กใช้ในการแก้ปัญหา วิธีการของนิวรอลเน็ตเวิร์กเป็นวิธีการที่ให้เครื่องเรียนรู้จากตัวอย่างต้นแบบ แล้วทำการฝึก ( training ) ให้ระบบได้รู้จักที่จะคิดแก้ปัญหาที่กว้างขึ้นได้ ในโครงสร้างของนิวรอลเน็ตจะประกอบด้วยโหนดสำหรับอินพุท-เอาท์พุท (Input – Output) และการประมวลผลจะกระจายอยู่ในโครงสร้างเป็นชั้น ๆ ได้แก่ input layer, output layer และ hidden layers การประมวลผลของนิวรอลเน็ตเวิร์กจะอาศัยการส่งการทำงานผ่านโหนดต่าง ๆ ใน layer เหล่านี้




**ภาพที่ 2.1** รูปแบบ Neural Network

3.) Nearest neighbor วิธีจัดกลุ่มสมาชิกที่มีความใกล้เคียงกัน เป็นเทคนิคที่ใช้ในการจำแนกข้อมูลในชุดข้อมูล โดยการรวมข้อมูลที่มีลักษณะที่คล้ายคลึงกันมากที่สุดเข้าเป็นกลุ่มเดียวกัน บางครั้งเรียกเทคนิคนี้ว่า k-nearest neighbor (K-NN) เทคนิคนี้จะเหมาะสำหรับข้อมูลที่มีลักษณะเป็นตัวเลข การนำไปใช้ คือ หาวิธีการวัดระยะห่าง (Distance) ระหว่างแต่ละแอตทริบิวต์ในข้อมูลให้ได้ แล้วทำการคำนวณค่าออกมา จากนั้นรวมค่าระยะห่างของแอตทริบิวต์ทุกค่าที่วัดมาได้ เมื่อสามารถคำนวณระยะห่างระหว่างเงื่อนไขหรือกรณีต่างๆได้แล้ว จะทำการเลือกชุดของเงื่อนไขที่ใช้จัดคลาสมาเป็นฐานสำหรับการจัดคลาสในเงื่อนไขใหม่ๆ และทำให้สามารถตัดสินใจ

ได้ว่าขอบเขตของจุดข้างเคียงที่ควรเป็นนั้นควรมีขนาดเท่าไร และอาจตัดสินใจได้ด้วยว่าจะนับจำนวนจุดข้างเคียงตัวมันได้อย่างไร โดยอาจจะให้น้ำหนักกับจุดข้างเคียงที่ใกล้ตัวมันมากที่สุดมากกว่าจุดที่ไกลห่างออกไป

4.) Genetic Algorithms (GA) เป็นทฤษฎีที่จำลองกระบวนการวิวัฒนาการทางธรรมชาติ คือ การคัดเลือกทางธรรมชาติโดยอาศัยพื้นฐานความคิดทางพันธุกรรมในการถ่ายทอดลักษณะต่างๆไปยังรุ่นถัดไปที่สามารถนำมาพัฒนาใช้ในการหาคำตอบที่เหมาะสมที่สุดของแต่ละปัญหา จีเนติกอัลกอริทึ่มเป็นวิธีการหาคำตอบ โดยการพิจารณาและดำเนินการจากกลุ่มของคำตอบของปัญหาที่ถูกสร้างขึ้นมาโดยการเข้ารหัส คือการแปลงค่าตัวแปรหรือพารามิเตอร์ (Parameters) ของปัญหาให้อยู่ในรูปโครงสร้างของโครโมโซม (Chromosomes) ที่กำหนด เพื่อคัดเลือกโครโมโซมคำตอบที่เหมาะสมสำหรับสร้างวิวัฒนาการของคำตอบให้ดีขึ้นตามกระบวนการทางพันธุศาสตร์ โดยการแลกเปลี่ยนค่าพารามิเตอร์ต่างๆระหว่างโครโมโซมที่ถูกคัดเลือก ซึ่งจะทำให้คำตอบของปัญหาถูกปรับปรุงให้ดียิ่งขึ้น

เทคนิคดาต้าไมนิ่งส่วนใหญ่ ต้องการเทรนนิ่งข้อมูลจำนวนมากที่ประกอบด้วยหลายๆ ตัวอย่างเพื่อจะสร้างกฎที่ใช้ในการจัดหมวดหมู่ กฎของความสัมพันธ์ คลัสเตอร์ การทำนายล่วงหน้า ดังนั้นชุดของข้อมูลขนาดเล็กจะนำไปสู่ความไม่น่าไว้วางใจของผลสรุปที่ได้ ไม่มีเทคนิคใดเลยที่จะสามารถแก้ปัญหาของการทำดาต้าไมนิ่งได้ทุกปัญหา ดังนั้นความหลากหลายของเทคนิคจึงเป็นสิ่งที่จำเป็นในการไปสู่วิธีการแก้ปัญหาของดาต้าไมนิ่งได้ดีที่สุด

## 2.4 ชนิดของข้อมูล และรูปแบบของข้อมูล

การทำดาต้าไมนิ่ง แบ่งตัวแปรข้อมูลออกเป็น 2 ลักษณะ คือ ตัวแปรแบบ Categorical และ ตัวแปรแบบ Quantitative

1. ตัวแปรแบบ Categorical แบ่งเป็น
   - Norminal Variable เป็นตัวแปรที่ลำดับของข้อมูลไม่มีผลกับค่า คือ ไม่มีลำดับในค่าที่เป็นไปได้ (Possible Value) เช่น เพศ(ชาย, หญิง), ระดับการศึกษา(ปริญญาโท , ปริญญาตรี, ม.ปลาย, ปวช), สถานะการแต่งงาน (โสด, แต่งงาน, หย่า, ไม่ทราบ)
   - Ordinal Variable เป็นตัวแปรที่ลำดับของข้อมูลมีผลกับค่า คือ มีลำดับสำหรับค่าที่เป็นไปได้ เช่น เกรด (A, B, C, D, F), ลำดับของลูกค้า (ดี, ปานกลาง, ไม่ดี) โดยถ้าแปลงให้อยู่ในรูปของตัวเลขจะต้องคงไว้ให้ได้ความหมายเดิม
2. ตัวแปรแบบ Quantitative แบ่งเป็น
   - Continuous ค่าที่เก็บเป็นเลขจำนวนจริง (Real number) หรือเป็นค่าที่ต่อเนื่อง เช่น รายได้

- Discrete ค่าที่เก็บเป็นเลขจำนวนเต็ม (Integer) เช่น ข้อมูลจำนวนพนักงาน

## 2.5 กระบวนการทำงานของ Data Mining

กระบวนการของดาต้าไมนิ่งเป็นกระบวนการสร้างแบบจำลอง (Model) ของกลุ่มข้อมูล เพื่อสร้างความเข้าใจในแนวโน้ม รูปแบบ และความเกี่ยวข้องกันของกลุ่มข้อมูลเพื่อใช้ในการอธิบายหรือทำนายผลข้อมูล กระบวนการของการทำดาต้าไมนิ่งประกอบด้วย 4 ขั้นตอนหลัก ดังนี้

1. กำหนดจุดประสงค์ทางธุรกิจ (Business Objective Determination)
2. การเตรียมข้อมูล (Data Preparation)
3. การทำดาต้าไมนิ่ง (Data Mining)
4. Analysis of Results and Knowledge Presentation

### 2.5.1 กำหนดจุดประสงค์ทางธุรกิจ (Business Objective Determination)

การกำหนดวัตถุประสงค์ทางธุรกิจเป็นพื้นฐานหลักในการทำดาต้าไมนิ่ง ซึ่งส่วนนี้เป็นส่วนที่บอกถึงวัตถุประสงค์ ขอบเขต เป้าหมาย และสิ่งที่ต้องการจากการทำดาต้าไมนิ่ง ดังนั้นจึงต้องทำความเข้าใจปัญหา กำหนดปัญหา และเป้าหมายในการแก้ปัญหาให้ครอบคลุม ชัดเจน ซึ่งในส่วนนี้จะประกอบด้วยการวิเคราะห์ทางธุรกิจ และการวิเคราะห์ข้อมูลเบื้องต้นว่าเรามีข้อมูลใดอยู่บ้างและต้องการอะไรจากข้อมูล ซึ่งเป้าหมายทางธุรกิจนี้จะนำไปสู่การสร้างแบบจำลองที่เหมาะสม ซึ่งแบบจำลองที่สร้างขึ้นจะแตกต่างกัน โดยขึ้นอยู่กับเป้าหมายทางธุรกิจ

### 2.5.2 การเตรียมข้อมูล (Data Preparation)

เป็นขั้นตอนที่ใช้เวลานานที่สุด และใช้ความพยายามมากกว่าขั้นตอนอื่นๆทั้งหมด เนื่องจากโมเดลที่ได้จากการทำดาต้าไมน์นิ่งจะให้ผลลัพทธ์ที่ถูกต้องหรือไม่นั้น ขึ้นอยู่กับคุณภาพของข้อมูลที่ใช้ กล่าวคือถ้าข้อมูลที่ใช้นั้นไม่ถูกต้อง มีความผิดพลาด ย่อมสะท้อนถึงผลลัพธ์ที่ได้ ซึ่งอาจทำให้ตีความผลลัพธ์ได้คลาดเคลื่อนเช่นกัน หน้าที่ของขั้นตอนนี้ คือ การจัดการข้อมูลให้สามารถนำเข้าสู่อัลกอริทึมส์ของดาต้าไมนิ่งได้ Data Preparation สามารถแบ่งออกเป็น 3 ส่วน ได้แก่ Data Selection, Data Preprocessing และ Data Transformation

- ทำการคัดเลือกข้อมูล (Data Selection) เราควรกำหนดเป้าหมายก่อนว่าเราจะทำการวิเคราะห์อะไร แล้วจึงเลือกใช้เฉพาะข้อมูลที่เกี่ยวข้องกับสิ่งที่เราต้องการจะทำการวิเคราะห์ และนำข้อมูลที่ไม่ต้องการออกไป การเลือกข้อมูลนั้นจะแตกต่างไปตามจุดประสงค์ของแต่ละธุรกิจที่ได้กำหนดไว้ตั้งแต่ต้น

Data Selection เป็นขั้นตอนการระบุถึงแหล่งข้อมูลที่จะนำมาใช้ในการทำดาต้าไมนิ่ง รวมถึง การนำข้อมูลที่ต้องการออกมาจากฐานข้อมูล เพื่อทำการพิจารณาในเบื้องต้นต่อไป ในการ

เลือกข้อมูลนั้นจะต้องพิจารณาถึงอายุของข้อมูลที่จะนำมาใช้ด้วย เนื่องจากสถานการณ์ภายนอกมีการเปลี่ยนแปลงอยู่ตลอดเวลา ซึ่งจะทำให้ประสิทธิภาพของการทำดาต้าไมนิ่งลดลง เช่น ข้อมูลอาชีพของลูกค้า ซึ่งจะมีการเปลี่ยนแปลงบ่อยเมื่อเวลาผ่านไป เป็นต้น เพราะฉะนั้นการนำเอาข้อมูลมาใช้นั้นจะต้องทำการตรวจสอบให้แน่ชัดว่าข้อมูลนั้นถูกต้องหรือไม่

- การกลั่นกรองข้อมูล (Data Preprocessing) ในบางกรณีอาจพบข้อมูลที่ไม่ถูกต้อง อันเนื่องมาจากปัญหาในระหว่างการจัดเก็บข้อมูล กล่าวคือ อาจเกิด Noisy Data หรือ Missing Value ขึ้น เช่น กรอกข้อมูลไม่ครบ กรอกข้อมูลซ้ำซ้อน หรือกรอกข้อมูลผิดพลาด เป็นต้น ในขั้นตอนนี้จะทำการคัดหรือลบข้อมูลที่ไม่เกี่ยวข้องออกไป (Data Cleaning) หรืออาจทำการซ่อมข้อมูลที่ขาดหายไปด้วยวิธีการบางอย่าง เช่น การพิจารณาจากค่าเฉลี่ยของข้อมูลส่วนใหญ่ เป็นต้น เพื่อทำให้มั่นใจว่าคุณภาพของข้อมูลที่ถูกเลือกนั้นถูกต้อง และเหมาะสมที่จะนำไปทำดาต้าไมนิ่ง

  - Noisy Data คือ ค่าของข้อมูลที่ผิดไปจากค่าที่ควรจะเป็น ซึ่งอาจเกิดจากความประมาทเลินเล่อในการบันทึกข้อมูลของผู้ใช้ เช่น บันทึกอายุเป็น 650 ปี หรือบันทึกรายได้ติดลบ เป็นต้น ซึ่งข้อมูลที่ผิดนี้อาจไปรบกวนการวิเคราะห์ได้ ดังนั้นจึงต้องกำจัดข้อมูลที่ผิดนี้ออกไป หรือทำการแก้ไขข้อมูลใหม่ให้ถูกต้อง โดยมีค่าข้อมูลที่อยู่ในช่วงที่ควรจะเป็น
  - Missing Value คือ ค่าของข้อมูลที่ขาดหายไป การจัดการกับค่าที่หายไป นั้นสามารถจัดการได้ด้วยเทคนิคที่ต่าง ๆ กัน เช่น การตัดข้อมูลนั้นทิ้งทั้งรายการ หรือบันทึกส่วนที่ขาดหายไปด้วยค่าเฉลี่ย (Mean) หรือค่าที่ปรากฏบ่อย (Mode) สำหรับข้อมูลประเภท Quantitative ส่วนข้อมูลประเภท Categorical อาจบันทึกด้วยค่าที่ปรากฏบ่อย (Mode)

- การแปลงข้อมูล (Data Transformation) เป็นขั้นตอนการเตรียมข้อมูลให้อยู่ในรูปแบบที่พร้อมนำไปใช้ในการวิเคราะห์ตามอัลกอริทึมของดาต้าไมน์นิ่งที่เลือกใช้ เป็นการแปลงข้อมูลที่เลือกมาให้อยู่ในรูปแบบของข้อมูลที่ไม่มีความขัดแย้ง ถูกจัดระเบียบ และกลั่นกรองมาอย่างเรียบร้อยเหมาะสม เช่น การแปลงตัวแปรแบบ Quantitative ให้เป็นแบบ Categorical โดยแบ่งค่าของตัวแปรให้เป็นช่วงๆ เช่น การแปลงข้อมูลเงินเดือน เป็นต้น นอกจากนี้ยังมีเทคนิคของการแปลงตัวแปรแบบ Categorical ให้เป็น Numeric เช่น การแปลงยี่ห้อรถ HONDA, TOYOTA และ NISSAN ให้เป็น 001, 010 และ 011 เป็นต้น

### 2.5.3 การทำดาต้าไมนิ่ง (Data Mining)

เป็นขั้นตอนการค้นหารูปแบบที่เป็นประโยชน์จากข้อมูลที่มีอยู่ เป็นการนำข้อมูลที่จัดเตรียมไว้มาทำดาต้าไมนิ่ง เพื่อแปลงสภาพของข้อมูลดิบให้เป็นความรู้ในลักษณะของรูปแบบและกฎเกณฑ์ (Pattern And Rule Finder) ขั้นตอนการทำดาต้าไมนิ่งนี้จะเกี่ยวข้องกับการใช้

อัลกอริทึมหลายๆแบบ แต่ละ Data Mining Operation จะมีอัลกอริทึมส์ให้เลือกใช้ เช่น ถ้าเป็นการทำ Predictive Modeling อาจใช้ CART (Classification And Regression Tree) หรืออาจใช้ Supervised Learning Neural Network เช่น Back propagation Neural Net เป็นต้น

**2.5.4 Analysis of Results and Knowledge Presentation**

ในขั้นตอนนี้เป็นการวิเคราะห์และประเมินประสิทธิภาพของผลลัพธ์ที่ได้จากการทำดาต้าไมนิ่ง (โมเดลวิเคราะห์ข้อมูล) ว่ามีความครอบคลุม, เหมาะสม, ตรงตามวัตถุประสงค์ และสามารถตอบโจทย์ทางธุรกิจที่ตั้งไว้ในขั้นตอนแรกหรือไม่ เป็นการวิเคราะห์ การตีความผลลัพธ์ และการสรุปผลที่ได้จากการทำดาต้าไมนิ่ง ซึ่งเป็นสาระความรู้ (Knowledge) ที่จะนำไปเป็นสารสนเทศที่ช่วยในการตัดสินใจ การทำงานในส่วนนี้ต้องอาศัยทักษะในการวิเคราะห์ข้อมูล และการวิเคราะห์ทางธุรกิจ ซึ่งทำโดยการนำเอาแบบจำลองที่ได้ไปทดสอบกับข้อมูลชุดอื่น ที่ไม่ใช่ข้อมูลที่ใช้ในการสร้างแบบจำลอง เพื่อนำเอาผลลัพธ์ที่ได้มาเปรียบเทียบกับผลตามแบบจำลองว่ามีความแม่นยำและยอมรับได้หรือไม่ ซึ่งถ้าไม่สามารถยอมรับได้ก็จะทำการแก้ไขใหม่ เช่น การเพิ่มจำนวนของข้อมูลให้มากขึ้น หรือปรับเปลี่ยนค่าพารามิเตอร์ หรือเปลี่ยนไปใช้อัลกอริทึมอื่นแทน เป็นต้น เพื่อช่วยให้การวิเคราะห์ทำได้สะดวกและรวดเร็วขึ้น จึงได้มีการนำเสนอความรู้ที่ค้นพบ โดยการแสดงผลการวิเคราะห์ในรูปแบบที่สามารถเข้าใจได้ง่าย คือ มีการใช้เครื่องมือทางด้านกราฟฟิกในการแสดงผล ซึ่งอาจจะอยู่ในรูปแบบของตาราง กราฟ หรือรายงานรูปแบบต่างๆ เป็นต้น ในกรณีที่มีการสร้างโมเดลวิเคราะห์ข้อมูลหลายโมเดล ในขั้นตอนนี้จะทำการประเมินแต่ละโมเดลด้วยว่ามีส่วนดีส่วนด้อยอย่างไร และควรเลือกใช้โมเดลใด

กระบวนการทำดาต้าไมนิ่งนั้นเป็นกระบวนการของการกลั่นกรองสารสนเทศที่ซ่อนอยู่ในฐานข้อมูลใหญ่ เพื่อวิเคราะห์, ทำนายแนวโน้มและพฤติกรรมต่างๆ โดยอาศัยข้อมูลในอดีตเพื่อใช้สารสนเทศเหล่านี้ในการสนับสนุนการตัดสินใจทางธรุกิจ โดยที่รูปแบบที่ใช้ในการทำดาต้าไมนิ่งนั้น มีอยู่หลายรูปแบบด้วยกัน ซึ่งในแต่ละรูปแบบก็จะมีวิธีการที่แตกต่างกันออกไป ประกอบด้วยหลายขั้นตอน รูปแบบต่างๆที่นำมาใช้ในการทำดาต้าไมนิ่งต่างก็มีวัตถุประสงค์ต่างๆกันขึ้นอยู่กับผลลัพธ์ที่ต้องการ จุดที่ต้องให้ความสำคัญ คือ การทำ Data Preparation ในการปรับปรุงแบบข้อมูล และการจัดการเกี่ยวกับความขัดแย้งต่างๆของข้อมูล โดยทั่วไปหน้าที่หรือประเภทของงานตามลักษณะของแบบจำลองที่ใช้ในการทำดาต้าไมนิ่งนั้นสามารถแบ่งกลุ่มได้เป็น 2 กลุ่มใหญ่ๆ คือ Predictive Data Mining และ Descriptive Data Mining และกระบวนการของการทำดาต้าไมนิ่งก็ประกอบด้วย 4 ขั้นตอนหลัก คือ กำหนดจุดประสงค์ทางธุรกิจ (Business Objective Determination), การเตรียมข้อมูล (Data Preparation), การทำดาต้าไมนิ่ง (Data Mining) และAnalysis of Results and Knowledge Presentation

## 2.6 The Predictive Model Markup Language (PMML)

PMML คือ ภาษามาตรฐาน (de facto standard language) ที่ใช้แสดงแบบจำลองการทำ Data Mining ช่วยให้รวมเหมืองข้อมูลที่แตกต่างกันเข้าเป็นภาษาเดียวกัน ซึ่งแนวทางนี้จะสามารถย้ายแบบจำลองไปยังส่วนอื่นๆได้อย่างง่ายดาย คือ เอกสาร PMML หนึ่งอาจถูกพัฒนาโดยระบบหนึ่ง แต่สามารถนำไปใช้งานในอีกระบบหนึ่งได้

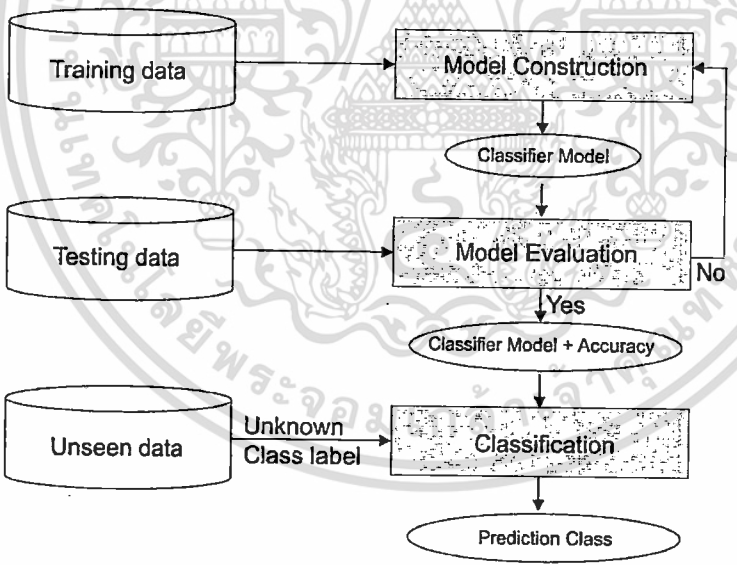
PMML ได้รับการพัฒนาโดยนักพัฒนาการทำเหมืองข้อมูล (Data Mining Group: DMG) เพื่อรองรับการทำเหมืองข้อมูลให้เป็นมาตรฐาน PMML มีหลายรุ่น หรือหลาย version ด้วยกัน โดย version ล่าสุด คือ version 4.0 ซึ่งถูกปล่อยออกมาเมื่อเดือน มิถุนายน 2009

การสร้างเอกสาร PMML จะมีกฎ หรือการกำหนดรายละเอียดที่ใช้ในการอธิบายแบบจำลองของเทคนิคต่างๆ แตกต่างกัน โดยรายละเอียดทั้งหมดสามารถอ่านเพิ่มเติมได้ที่ http://www.dmg.org/ หรือรายละเอียดเกี่ยวกับกฎในการอธิบายแบบจำลองของเทคนิค TreeModel ที่โครงการพัฒนาระบบฉบับนี้ใช้ สามารถอ่านต่อได้ที่ภาคผนวก ข

# บทที่ 3
# การจัดกลุ่ม(Classification)

การจัดกลุ่มเป็นเทคนิคหนึ่งของการทำเหมืองข้อมูล ที่ใช้ในการ Predictive Modeling ซึ่งมีการทำงานแบบ Supervised Learning

เป็นกระบวนการสร้างโมเดลจัดการข้อมูลให้อยู่ในกลุ่มที่กำหนดมาให้ โดยจะมีการนำข้อมูลส่วนหนึ่งมาสอนให้ระบบเรียนรู้ (Training Set) เพื่อจำแนกข้อมูลออกเป็นกลุ่มตามที่ได้กำหนดไว้ ผลลัพธ์ที่ได้จากการเรียนรู้คือ โมเดลจำแนกประเภทข้อมูล (Classifier Model) และจะนำข้อมูลส่วนที่เหลือจากข้อมูลสอนระบบ เป็นข้อมูลที่ใช้ทดสอบระบบ (Test Set) ซึ่งกลุ่มที่แท้จริงของข้อมูลที่ใช้ทดสอบนี้จะถูกนำมาเปรียบเทียบกับกลุ่มที่หามาได้จากโมเดล เพื่อทดสอบความถูกต้อง และปรับปรุงโมเดลจนกว่าจะได้ค่าความถูกต้องในระดับที่น่าพอใจ หลังจากนั้นเมื่อมีข้อมูลใหม่ที่ไม่เคยเห็นมาก่อนเข้ามา เราจะนำข้อมูลใหม่นั้นมา ผ่านโมเดล โดยโมเดลจะสามารถทำนายกลุ่ม หรือ class ของข้อมูลนี้ได้อย่างถูกต้อง




**ภาพที่ 3.1** แสดงกระบวนการของ Classification

## 3.1 Decision Tree

เป็นวิธีการหนึ่งในการทำ data mining แบบ Classification ที่มีการทำงานแบบ Supervised Learning เพื่อใช้ในการทำนายค่า (Predictive) เป็นแผนผังต้นไม้ตัดสินใจในรูปแบบ node แสดงผลลัพธ์จากการตัดสินใจในเงื่อนไขต่างๆเชื่อมต่อกันเป็นกิ่ง แตกแขนงออกไป เทคนิคนี้ช่วยให้ผู้ใช้เข้าใจถึงความสัมพันธ์และคุณสมบัติของข้อมูลได้ง่าย

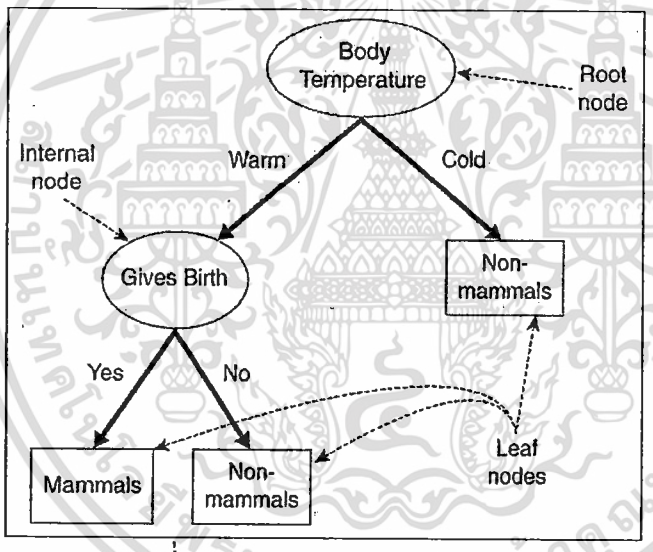
### 3.1.1 โครงสร้างของ Decision Tree ประกอบด้วย

1.) Decision Node : เป็นส่วนของเงื่อนไขการตัดสินใจ โดยที่แต่ละโหนดแสดง attribute

2.) Branch : เป็นการเชื่อมต่อระหว่าง Node โดยแต่ละกิ่งจะแสดงผลในการทดสอบ , เป็นค่า attribute ของ attribute ภายใน node ที่แตกกิ่งนี้ออกมาซึ่ง node จะแตกกิ่งเป็นจำนวนเท่ากับจำนวนค่า attribute ของ node นั้น

3.) Leaf Node : แสดงค่าที่เป็นไปได้จากเงื่อนไขการตัดสินใจ คือแสดง class ที่กำหนดไว้แล้วล่วงหน้าภายใต้เงื่อนไขการตัดสินใจ

รูปแบบของ Tree จะประกอบด้วย Node แรกสุดที่เรียกว่า Root Node จาก Root Node ก็จะแตกออกเป็น Node ลูก และที่ Node ลูกก็จะมีลูกของตัวเอง ซึ่ง Node ที่ระดับสุดท้ายจะเรียกว่า Leaf Node




**ภาพที่ 3.2** แสดงตัวอย่างของ Decision Tree

จะเห็นว่า จาก Root Node จนถึง Leaf Node จะมีเพียงเส้นทางเดียวเท่านั้น ซึ่งเส้นทางนี้จะอธิบาย ถึงกฎที่ใช้สำหรับการจัดหมวดหมู่ของแต่ละกลุ่ม ซึ่งในแต่ละ Leaf Node นั้นอาจเป็นกลุ่มเดียวกัน ซึ่งเกิดจากเหตุผล ที่แตกต่างกันได้

### 3.1.2 ลักษณะการเรียนรู้ของต้นไม้ตัดสินใจ

- ผลการเรียนรู้แสดงอยู่ในรูปที่เข้าใจง่าย ทำให้ง่ายต่อการวิเคราะห์ attribute ที่มีผลต่อการแยกแยะกลุ่มต่างๆ
- แต่ละเส้นทางจาก root node ถึง leaf node สามารถแสดงให้อยู่ในรูปกฎ if-then ได้

- มีความทนทานต่อข้อมูลข้อมูลที่มีสัญญาณรบกวน (noisy data) เช่น คุณสมบัติที่ไม่เกี่ยวข้อง และค่าคุณสมบัติที่ผิดพลาดหรือขาดหาย

**3.1.3 การสร้างต้นไม้ตัดสินใจ** จะเป็นแบบการค้นหาจากบนลงล่างแบบตะกรม (Top-Down greedy search) โดยเริ่มจาก

1. เลือก Attribute ที่สำคัญที่สุดมาแบ่งข้อมูล โดย Attribute นี้จะถูกนำมาสร้างเป็น Root Node โดยจะมี Target Attribute เป็นผลลัพธ์ซึ่งเป็น Leaf Node ถูกกำหนดไว้ก่อน

2. เมื่อข้อมูลผ่านการแบ่งแยกที่ root node ตาม attribute ของ root node แล้ว ก็หา attribute ที่ดีที่สุดของข้อมูลที่ผ่านการแบ่งแยกนั้นมาสร้างเป็น child node ของ root node นั้นต่อไป คือ นำค่าที่เป็นไปได้ใน Attribute ที่ถูกเลือกมาแตกออกเป็นกลุ่มของตัวเอง

3. วนกลับไปทำที่ขั้นตอนแรก คือ เลือกAttribute ที่สำคัญที่สุดจากข้อมูลที่เข้ามาเพื่อเป็นตัวแบ่งต่อไป กล่าวคือ จะวนสร้าง child node และ sub tree ของแต่ละกิ่งไปเรื่อยๆจนกว่าข้อมูลที่ผ่านการแบ่งแยกนั้นจะจัดอยู่ในกลุ่มเดียวกัน หรือจำนวนข้อมูลที่ผ่านการแบ่งแยกในกิ่งหนึ่งๆมีค่าน้อยกว่าค่าที่กำหนดไว้

**3.1.4 ข้อดีของการทำ Classification แบบ Decision Tree**

- เป็นแบบจำลองที่ง่ายในการทำความเข้าใจ ทำการสร้างกฎ (Rule) แสดงความสัมพันธ์ของข้อมูลจากต้นไม้ได้ง่าย
- ในการสร้างแบบจำลองจะมีการคัดตัวแปรที่ไม่มีผลในการสร้างต้นไม้ออกได้ โดยไม่กระทบการสร้างต้นไม้
- แต่ละกิ่งของต้นไม้แสดงความสัมพันธ์ที่เกี่ยวข้องกันของข้อมูล
- จัดการกับข้อมูลที่ไม่สมบรูณ์ที่ไม่มีผลต่อการสร้างต้นไม้และเพื่อลดขนาดของต้นไม้

**3.1.5 ข้อเสีย การทำ Classification แบบ Decision Tree**

- การแบ่งกลุ่มแบบ Decision Tree กรณีเป็นข้อมูลที่มีค่าต่อเนื่อง เช่น ข้อมูลรายได้ ข้อมูลราคา ต้องทำการแปลงให้อยู่ในช่วงหรือตัดเป็นกลุ่มก่อน
- เมื่อ Algorithm เลือกว่าจะใช้ค่าไหนเป็นตัวแบ่งกลุ่มแล้วก็จะไม่สนใจค่าอื่นที่อาจมีความสำคัญเช่นเดียวกัน
- การจัดการกับข้อมูลที่ไม่ทราบค่า อาจมีผลกระทบกับผลลัพธ์ของ Decision Tree
- ปัญหาเรื่อง Over-Fitting คือ การที่ต้นไม้สร้าง node และ branch ที่มีความลึก และซับซ้อนมากเกินความจำเป็น หรือการที่จำนวนของตัวอย่างข้อมูลที่ใช้เรียนรู้ (Training Data) มีจำนวนน้อยเกินกว่าจะสร้างต้นไม้ที่จะจัดกลุ่มได้ตามที่ต้องการ

เป็นปัญหาการเจาะจง model กับข้อมูลมากเกินไป โดยปัญหานี้ทำให้ได้โครงสร้างต้นไม้ที่สามารถจำแนกข้อมูลได้ดีกับชุดข้อมูลที่ใช้สร้างต้นไม้ตัดสินใจเท่านั้น แต่เมื่อนำไปใช้กับข้อมูลใหม่ ประสิทธิภาพในการจำแนกกลุ่มข้อมูลจะลดลง

วิธีการแก้ปัญหา Over-Fitting คือ การใช้เทคนิค Pruning Tree ซึ่งเป็นวิธีการตัดเล็มต้นไม้เพื่อให้ต้นไม้มีขนาดเล็กลง และลดความซับซ้อนของต้นไม้ และยังสามารถนำไปใช้งานได้มีประสิทธิภาพมากกว่าต้นไม้ที่มีขนาดใหญ่ ซึ่งก็คือ ต้นไม้ที่ได้ก่อนทำการ prune นั่นเอง

## 3.2 ID3 Algorithm

ID3 คือวิธีการสร้างต้นไม้ตัดสินใจ โดยมีพื้นฐานจากเทคนิค Divide-and-Conquer (วิธีการแบ่งปัญหาใหญ่เป็นปัญหาย่อย) ที่ใช้ในการสร้างต้นไม้ หรือที่เรียกว่า Top-Down Induction พัฒนาโดย J. Ross Quinlan (1975) เป็นอัลกอริธึมพื้นฐานที่ใช้ในการสร้างการตัดสินใจแบบโครงสร้างต้นไม้ที่ใช้หลักการของ Information Gain ค่าที่วัดได้จะนำมาใช้ตัดสินใจว่าจะเลือกใช้ตัวแปรใดในการทำนาย หรือแบ่งประเภทของข้อมูล โดยการจำแนกที่ดีที่สุด คือ ให้ leaf node ที่เป็นข้อมูลเดียวกันทั้งหมด และค่า gain ที่สูงที่สุด หมายถึง การจำแนก class ที่ดีที่สุด

ค่ามาตรฐานเกน (Gain criterion) คำนวณได้จาก ค่าสารสนเทศทั้งหมดของชุดข้อมูลนั้นลบด้วยค่าสารสนเทศหลังจากเลือก attribute ใด attribute หนึ่งเป็น root หรือ node

การวัดหาค่าของ entropy หรือค่าสารสนเทศ ค่า entropy ที่น้อย จะบ่งบอกว่าข้อมูลชุดนั้นแตกต่างกันน้อยหรือเกือบจะเป็นพวกเดียวกันทั้งหมด แต่ถ้าค่า entropy สูง จะบ่งบอกว่าข้อมูลชุดนั้นมีความแตกต่างกันมาก หรือประกอบด้วยตัวอย่างหลายพวกที่มีจำนวนใกล้เคียงกัน ซึ่งหมายความว่า ยิ่ง entropy มีค่าน้อยเท่าไรจะยิ่งดีเท่านั้น กล่าวคือความบริสุทธิ์ของข้อมูลจะยิ่งสูงขึ้น

การวัดหาค่าของ entropy เป็นวิธีการวัดความแตกต่างของกลุ่มข้อมูลที่ใช้กันอย่างแพร่หลาย ซึ่งค่า entropy สามารถมีค่าได้ตั้งแต่ 0 ถึง 1 โดย

ค่า entropy จะมีค่าเท่ากับ 0 ถ้ากลุ่มข้อมูลมีค่าข้อมูลที่เหมือนกันทั้งหมด

ค่า entropy จะมีค่าสูงขึ้นเรื่อยๆ ถ้าจำนวนข้อมูลที่มีค่าข้อมูลที่แตกต่างกันมีจำนวนไม่เท่ากัน คือ ยิ่งจำนวนข้อมูลของแต่ละค่าที่แตกต่างกันมีจำนวนใกล้เคียงกันมากเท่าไหร่ ค่า entropy ก็จะยิ่งมีค่าสูงเพิ่มขึ้นเท่านั้น

ค่า entropy จะมีค่าเท่ากับ 1 ถ้าจำนวนข้อมูลที่มีค่าข้อมูลที่แตกต่างกันมีจำนวนเท่าๆกัน

ขั้นตอนการทำ เริ่มจาก

1.) หาค่าสารสนเทศของ T หรือ ค่า entropy ของ T ที่ต้องการสำหรับจำแนกข้อมูลออกเป็นแต่ละกลุ่ม ตามสมการ

$$\text{info}(T) = -\sum_{j=1}^{K} \frac{freq(C_j, T)}{|T|} \times \log_2 \left[ \frac{freq(C_j, T)}{|T|} \right] \quad (3.1)$$

โดยที่

*freq*($C_j$, T) แทน จำนวนของข้อมูลใน T ซึ่งอยู่ใน class $C_j$

T แทน set ของข้อมูลใด ๆ

|T| แทน จำนวนของข้อมูลใน T

2.) หาค่า entropy ของ attribute X ซึ่งมีค่าของ attribute เป็น ($x_1$, $x_2$, … , $x_n$) ตามสมการ

$$\text{info}_X(T) = \sum_{i=1}^{n} \frac{|T_i|}{|T|} \times \text{info}(T_i) \quad (3.2)$$

โดยที่

*T* แทน Training Set

3.) หาค่า Gain criterion ของทุก ๆ attribute ที่ไม่ใช่ attribute เป้าหมาย (ไม่ใช่ attribute ที่เป็น class) ซึ่ง attribute ใดมีค่า Gain สูงสุดจะถูกกำหนดเป็น node บนสุด (Root Node) ตามสมการ

$$gain(X) = \text{info}(T) - \text{info}_X(T) \quad (3.3)$$

โดยที่

*gain(X)* เป็นค่าที่บอกระดับความสามารถของการจำแนก class ของ attribute X เพื่อจัดกลุ่มของข้อมูล

4.) ทำซ้ำ (Loop) ค่าต่างๆทั้งหมดที่เป็นไปได้ของ X ตามขั้นตอนต่อไปเรื่อยๆจนกว่าจะหมด แล้วเลือกตัวอย่าง (Sample) ของแอททริบิวต์ X ที่มีค่ามากที่สุด ทำซ้ำตั้งแต่หัวข้อ 1.) ซ้ำจนกว่าจะสร้างโครงสร้างต้นไม้ตัดสินใจเสร็จ โดยการส่งค่า parameter ต่าง ๆ คือ ค่า training data ทั้งหมดที่มีค่าตรงตามเป้าหมาย และค่าของ attribute ที่ลบเอา attribute X ออก

## 3.3 C4.5 Algorithm

C4.5 เป็นอัลกอริทึมในการสร้างต้นไม้การตัดสินใจ ซึ่งพัฒนาโดย J. Ross Quinlan (1993) เป็นการนำเอา ID3 มาปรับปรุงให้มีความสามารถมากขึ้น โดย

- ใช้วิธีการ gain ratio criterion เพิ่มเติม มีการแก้ปัญหาการเกิดความโน้มเอียง (Bias) หรือความอคติของข้อมูล โดยการใช้ค่า split information ของแต่ละ attribute
- สามารถจัดการกับข้อมูลที่ขาดหายไป (Missing Values) หรือไม่ทราบค่าได้ (Unknown attribute values)
- สามารถจัดการกับข้อมูลที่เป็นค่าต่อเนื่อง หรือข้อมูลตัวเลขได้ (Continuous attribute values)

ใน ID3 จะใช้ค่ามาตรฐานเกน (Gain criterion) เป็นหลักในการเลือก attribute ที่จะใช้เป็น root หรือ node แต่ใน C4.5 ได้เพิ่มการใช้ค่ามาตรฐานอัตราส่วนเกน (Gain Ratio criterion) ในการตัดสินใจเลือก attribute ที่จะใช้เป็น root หรือ node อีกอย่างหนึ่ง เนื่องจากค่ามาตรฐานเกนจะมีอคติ (Bias) อย่างมากกับข้อมูลที่ประกอบด้วย attribute ที่มีค่าที่เป็นไปได้จำนวนมากๆ เช่น ข้อมูลที่ประกอบด้วย attribute หมายเลขประจำตัว ซึ่งปกติจะมีค่าที่ไม่ซ้ำกัน ถ้าแบ่งข้อมูลตาม attribute นี้จะทำให้ได้จำนวนตัวอย่างเพียง 1 ตัวอย่าง ต่อ 1 กิ่งของ Decision tree และเมื่อคำนวณค่า entropy จากการแบ่งตัวอย่างบน attribute นี้จะได้เท่ากับ 0 เนื่องจากค่า $\log_2(1) = 0$ ทำให้ค่า gain ที่ได้ใน attribute นี้จะมีค่าสูงที่สุดเสมอ เช่น หากมีจำนวน 14 record จะต้องทำการสร้าง 14 กิ่ง โดยที่แต่ละกิ่งจะมีค่าเพียงอย่างเดียว

การแก้ไขความอคติ หรือ bias ของค่ามาตรฐานเกนสามารถทำได้ โดยการปรับค่ามาตรฐานเกนให้ถูกต้องโดยใช้ค่าสารสนเทศของการแบ่งแยก (split information)

ตามสมการ

$$\text{SplitInfo}(x) = -\sum_{i=1}^{n} \frac{|T_i|}{|T|} \times \log_2 \frac{|T_i|}{|T|} \qquad (3.4)$$

โดยที่

$T$ แทน record ใน training data ทั้งหมด

$T_i$ แทน จำนวนของ records ในแต่ละ subset ของ training data, หลังจากการ split attribute X

ค่า Split Information นี้จะแสดงถึงระดับการกระจายของข้อมูล เมื่อแบ่งข้อมูลตัวอย่าง T เป็น n ชุดย่อยตาม attribute X โดยค่านี้จะสูงสุดเมื่อ $|t_i|$ เป็น 1 เท่ากันในทุกกิ่ง และลดลงเมื่อค่า $|t_i|$ เพิ่มขึ้น เมื่อนำค่านี้ไปหารค่า gain criterion จะได้ค่า gain ratio criterion ซึ่งช่วยแก้ไขความอคติของค่า gain criterion ได้ โดยทำให้ค่า gain ratio criterion ที่ใช้ในการแบ่งด้วย attribute ที่มี

การกระจายสูงถูกปรับลดลง ดังนั้นค่า gain ratio criterion ใน attribute ตัวอย่างที่มีการกระจายตัวของข้อมูลสูงดังที่กล่าวมาแล้วจึงไม่มีค่าสูงที่สุดเสมอ โดย

$$\text{Gain Ratio (x)} = \frac{\text{InfoGain (x)}}{\text{SplitInfo (x)}} \tag{3.5}$$

### 3.3.1 การจัดการกับ Unknown attribute values

ใน training data ที่นำมาใช้สร้าง decision tree อาจมีกรณีที่ข้อมูลบางตัวในบาง attribute เป็นค่าที่ไม่ทราบค่าข้อมูลบรรจุอยู่ จะมีทางเลือกในการจัดการ 2 ทางเลือก คือ

1.) จะนำเฉพาะ training data ที่ทราบค่าเท่านั้น มาใช้เป็นข้อมูลสอน นั่นคือ ตัด record ของ training data ที่มีค่าข้อมูลที่ไม่ทราบค่าออกไปทั้งหมด ซึ่งจะส่งผลให้จำนวน training data ที่ใช้สร้าง decision tree ลดน้อยลง และอาจจะสูญเสียความรู้บางอย่างที่สำคัญ และสมควรจะได้จาก training data ชุดนี้

2.) ไม่มีการตัด record ของ training data ออก โดยจะรวมเอา training data ที่มีค่าข้อมูลที่ไม่ทราบค่ารวมเข้าไปด้วย วิธีการของ C4.5 จะใช้วิธีการคำนวณค่า gain criterion จากชุดข้อมูลที่ทราบค่าของ attribute นั้น แล้วปรับลดค่าให้ถูกต้องด้วยความน่าจะเป็นของตัวอย่างที่ทราบค่าต่อตัวอย่างทั้งหมด โดย

2.1) หา attribute เพื่อใช้แบ่งข้อมูล ทำโดย

- หาค่า info(T) และ $info_x(T)$ โดยพิจารณาเฉพาะข้อมูลที่รู้ค่าของ A
- หาค่า gain(X) ตามสมการ

$$\begin{aligned} \text{Gain}(x) &= \textit{probability A is known} \\ &\quad \times \textit{(info(T) – info}_x\textit{(T))} \\ &\quad + \textit{probability A is not known} \times 0 \\ &= F_x\textit{(info(T)) – info}_x\textit{(T))} \end{aligned} \tag{3.6}$$

โดยที่

Gain(X) แทน gain criterion

Probability A is known คือ ความน่าจะเป็นที่ทราบค่าของ A

(Info(T) – $info_x$(T)) คือ gain criterion ของชุดข้อมูลที่ทราบค่าเท่านั้น

T แทน training set

X แทน attribute ที่ใช้ทดสอบบนตัวอย่าง A

Probability A is not known คือ ความน่าจะเป็นที่ไม่ทราบค่าของ A

และ กำหนดให้ค่า gain criterion ของตัวอย่างที่ไม่ทราบค่าเป็น 0

- หาค่า split info(X) ซึ่งจะต้องมีการเพิ่มกลุ่มของข้อมูลที่ไม่รู้ค่าของ A เป็นอีก 1 subset คือ ปรับเพิ่มชุดตัวอย่างอีก 1 กลุ่ม เช่น ถ้า attribute ที่จะนำมาทดสอบมีค่าที่เป็นไปได้ n ค่า split info(X) จะถูกคำนวณโดยแบ่งข้อมูลออกเป็น n+1 subsets

2.2) ทำการแบ่ง training set ตาม attribute X เป็น subset $t_1, t_2, \ldots, t_n$ ชุด ตามค่าที่เป็นไปได้ คือ $O_1, O_2, \ldots, O_n$ ค่า

ข้อมูลตัวอย่าง (training data) จาก training set T ซึ่งทราบค่า $O_i$ จะถูกแบ่งกลุ่มอยู่ในชุดย่อย $t_i$ โดยมีค่าความน่าจะเป็นที่ training data นี้จะถูกแบ่งกลุ่มอยู่ในกลุ่ม $t_i$ เป็น 1 และค่าความน่าจะเป็นที่ training data นี้ จะถูกแบ่งอยู่ในกลุ่มอื่นมีค่าเป็น 0 แต่สำหรับ training data T ที่ไม่ทราบค่า เป็นไปได้ว่า training data นี้อาจจะมีค่าข้อมูลเป็นค่าใดค่าหนึ่งใน $O_i$ ซึ่งค่าความน่าจะเป็นจะมีค่าน้อยลง

ดังนั้นถ้าให้ w (weight) เป็นความน่าจะเป็นที่ training data นี้จะถูกแบ่งอยู่ในแต่ละ subset สำหรับ training data ที่ทราบค่า ค่าของ w จะมีค่าเป็น 1 และ training data ที่ไม่ทราบค่า ค่าของ w จะมีค่าเป็นความน่าจะเป็นที่จะเกิด $O_i$ ในแต่ละ subset $t_i$ ทำให้เมื่อต้องการค่า $|t_i|$ จะคำนวณได้จากผลรวมของค่า w ในแต่ละ subset $t_i$ แทนที่จะเป็นผลรวมของจำนวนตัวอย่างใน subset $t_i$

record ใน Tree จะมีค่า weight w ซึ่งค่าใน attribute ไม่ทราบค่าจะถูกกำหนดให้แต่ละ subset $T_i$ ด้วย weight

$$W \times \textit{Probability of outcome } O_i \qquad (3.7)$$

โดยความน่าจะเป็น คือ ผลรวมของ weight ของข้อมูลทั้งหมดใน T ซึ่งมีค่า $O_i$ หารด้วยผลรวมของ weight ของข้อมูลทั้งหมดใน T ซึ่งค่าใน attribute เป็นค่าที่ทราบค่า

### 3.3.2 การจัดการกับ Continuous attribute values

สำหรับ attribute ที่มีค่าข้อมูลเป็นค่าต่อเนื่อง จะทำการแบ่งตัวอย่างตามจุดแบ่ง (threshold) ที่เป็นไปได้ในระดับต่างๆของ attribute ที่เป็นค่าต่อเนื่อง แล้วทำการคำนวณ gain ratio criterion ในแต่ละจุดแบ่ง จากนั้นจะทำการเลือกจุดแบ่งที่มีค่า gain ratio criterion สูงที่สุดเป็นระดับที่จะใช้ในการแบ่งข้อมูลตัวอย่าง และใช้ค่า gain ratio criterion ที่สูงที่สุดนี้เป็นตัวแทนในการพิจารณาเลือก attribute ที่จะใช้แบ่งข้อมูลตัวอย่าง

สมมติว่า A เป็น attribute ชนิด continuous attribute values วิธีการหาค่า Threshold ที่เหมาะสม จะมีขั้นตอนดังนี้

1.) เรียงลำดับ training set ด้วยค่าใน attribute A จากน้อยไปมาก และเลือกเฉพาะค่าที่ไม่ซ้ำกัน m ค่า มาพิจารณา จะได้ $\{v_1, v_2, \ldots, v_n\}$

2.) หาค่า threshold (จุดที่เป็นระดับที่ใช้แบ่งข้อมูล) ซึ่งค่า threshold ใดๆจะอยู่ระหว่างค่าของ $v_i$ และ $v_{i+1}$ ดังนั้น จึงมีจุดที่ใช้ในการแบ่งข้อมูลเป็นจำนวน m-1 จุดที่เป็นไปได้ โดยคำนวณจุดกึ่งกลาง (midpoint) ของแต่ละช่วงได้จาก $V_i+V_{i+1}/2$

โดย C4.5 จะเลือกค่าที่มากที่สุดใน attribute A แต่ต้องไม่เกินค่า midpoint ในแต่ละช่วงนั้นๆของ training set แทนที่จะใช้จุด midpoint เป็นตัวแบ่ง เพื่อรับประกันว่าค่า threshold ทั้งหมดที่ปรากฏอยู่ใน tree หรือ rule เป็นค่าที่เกิดขึ้นจริงในข้อมูลตัวอย่าง

3.) หาค่า threshold ที่เหมาะสม โดยพิจารณาจากค่า threshold ที่มี gain ratio criterion สูงสุด

หลังจากได้ค่า threshold ที่เหมาะสมที่จะนำมาใช้ในการแบ่งข้อมูลแล้ว จะนำค่า threshold นั้นไปใช้ในการทดสอบค่าที่ attribute A ในการแบ่งข้อมูลตามเงื่อนไขเป็น $A \leq Z$ และ $A > Z$ (ทำการเปรียบเทียบค่าของ A กับค่า Threshold value Z)

### 3.3.3 การตัดกิ่ง (Pruning) แบบ Error-Based Pruning (EBP)

การตัดกิ่ง decision tree จะใช้ค่าทางสถิติในการตัดกิ่งที่มีความน่าเชื่อถือน้อยที่สุดออกไป เพื่อทำให้ต้นไม้ใหม่ที่ได้ สามารถทำงานได้รวดเร็วขึ้น และยังเป็นการปรับปรุงขีดความสามารถของต้นไม้ในการทำนายข้อมูลใหม่ๆได้แม่นยำมากขึ้นอีกด้วย

การ pruning นั้น จะทำให้แต่ละ leaf node ของ tree ใหม่ที่ได้ ไม่จำเป็นที่จะต้องประกอบด้วยข้อมูลที่อยู่ใน class เดียวกันทั้งหมด โดยในแต่ละ leaf node จะมีการระบุการกระจายของข้อมูลแต่ละ class ไว้ ซึ่งจะบอกถึงความน่าจะเป็นที่ข้อมูลจะอยู่ใน class นั้นๆ

วิธีการทำ pruning มีอยู่ด้วยกันหลายวิธี แต่โครงการพัฒนาระบบงานฉบับนี้ ได้เลือกทำการศึกษาวิธีการ prune มาเพียง 1 วิธี คือ การ pruning แบบ Error-Based Pruning (EBP) ซึ่งขั้นตอนการตัดกิ่งจะเริ่มขึ้นหลังจาก decision tree ได้ถูกสร้างขึ้นสมบูรณ์แล้ว วิธีนี้ใช้ชุดข้อมูลฝึกสำหรับสร้าง และตัดกิ่งต้นไม้ตัดสินใจ โดยไม่ต้องใช้ชุดข้อมูลที่แยกออกไปต่างหากสำหรับการตัดกิ่งโดยเฉพาะ

ขั้นตอนการทำงาน จะตรวจสอบ node จากล่างสุดขึ้นไปยัง root ของต้นไม้ โดยใช้วิธีการเข้าถึง node แบบ post-order traversal

การ pruning แบบ Error-Based pruning มีขั้นตอนการทำงานในการตัดแต่งหรือตัดเล็มต้นไม้ โดยที่ไม่ทำให้ค่าความผิดพลาดภายหลังจากการตัดแต่งต้นไม้เรียบร้อยแล้ว มีค่าเพิ่มขึ้น คือ จะพิจารณาตัด node ภายใน แล้วแทนด้วย leaf node โดยเปรียบเทียบอัตราความผิดพลาดของต้นไม้ตัดสินใจว่า อัตราความผิดพลาดของต้นไม้ก่อนการแทนที่ด้วย leaf node มีค่ามากกว่าหลัง

การแทนที่ด้วย leaf node ก็จะแทนที่ node นั้นด้วย leaf node ทันที กล่าวคือ จะยังคง node ภายใน node นั้นไว้ กับอัตราความผิดพลาดของต้นไม้ที่ได้ ด้วย class ที่เป็นไปได้ ถ้าหลังการแทนที่ด้วย leaf node มีอัตราความผิดพลาดที่มีค่ามากกว่า

การคำนวณความผิดพลาดที่เกิดจากการทำนายของแต่ละ leaf node และ subtree จะทำโดยข้อสมมติที่ว่า การแบ่งกลุ่ม set ของข้อมูลที่ไม่เคยพบมาก่อน มีขนาดเท่ากับ training set โดยการคำนวณจะใช้ function ทางสถิติ ซึ่งอยู่บนพื้นฐานการกระจายแบบ binomial ที่ระดับความเชื่อมั่นเท่ากับ CF (confidence factor)

จำนวน error ที่เกิดขึ้นเมื่อข้อมูลมีขนาดเท่ากับ N = N x $U_{CF}(E,N)$ (3.8)

โดยที่

N แทน ขนาดของข้อมูลที่ leaf node ใดๆ

E แทน จำนวนตัวอย่างที่แบ่งกลุ่มไม่ถูกต้องจาก N ตัวอย่าง

$U_{CF}(E,N)$ แทน ความน่าจะเป็นสูงสุดที่จะเกิด error

Algorithm C4.5 ใช้ค่าระดับความเชื่อมั่น CF = 25% โดยปริยาย

การ pruning แบบ Error-Based pruning ใช้การคำนวณช่วงความเชื่อมั่น (confidence level) เพื่อลดความลำเอียงที่เกิดจากการใช้ training set คำนวณค่าความผิดพลาดเพียงชุดข้อมูลเดียว

สมการคำนวณค่าความผิดพลาดของแต่ละ node คือ

$$e = \left[ f + \frac{z^2}{2N} + z\sqrt{\frac{f}{N} - \frac{f^2}{N} + \frac{z^2}{4N^2}} \right] \Big/ \left[ 1 + \frac{z^2}{N} \right] \quad (3.9)$$

โดย

CF = 25% คือ z = 0.69

f คือ error บน training data

N คือ ขนาดของข้อมูลที่ leaf node

## 3.4 ตัวอย่าง การสร้าง C4.5 Tree Model

ตารางที่ 3.1 แสดง Training Set

| Outlook | Temp (°F) | Humidity (%) | Windy? | Class |
|---|---|---|---|---|
| sunny | 75 | 70 | true | Play |
| sunny | 80 | 90 | true | Don't Play |
| sunny | 85 | 85 | false | Don't Play |
| sunny | 72 | 95 | false | Don't Play |
| sunny | 69 | 70 | false | Play |
| overcast | 72 | 90 | true | Play |
| overcast | 83 | 78 | false | Play |
| overcast | 64 | 65 | true | Play |
| overcast | 81 | 75 | false | Play |
| rain | 71 | 80 | true | Don't Play |
| rain | 65 | 70 | true | Don't Play |
| rain | 75 | 80 | false | Play |
| rain | 68 | 80 | false | Play |
| rain | 70 | 96 | false | Play |

$$\text{Info}(T) = -9/14 * \log_2(9/14) - 5/14 * \log_2(5/14)$$

$$= 0.940$$

$$\text{Info}_{\text{Outlook}}(T) = 5/14 * (-2/5 * \log_2(2/5) - 3/5 * \log_2(3/5)) + 4/14 * (-4/4 * \log_2(4/4) - 0/4 * \log_2(0/4)) + 5/14 * (-3/5 * \log_2(3/5) - 2/5 * \log2(2/5))$$

$$= 0.694$$

$$\text{Gain(Outlook)} = 0.940 - 0.694$$

$$= 0.246$$

$$\text{Split info(Outlook)} : \text{info}([5, 4, 5]) = -5/14 * \log_2(5/14) - 4/14 * \log_2(4/14) - 5/14 * \log_2(5/14)$$

$$= 1.577$$

$$\text{Gain ratio(Outlook)} = 0.246/1.577$$

$$= 0.156$$

$$\text{Info}_{\text{Windy}}(T) = 6/14 * (-3/6 * \log_2(3/6) - 3/6 * \log_2(3/6)) + 8/14 * (-6/8 * \log_2(6/8) - 2/8 * \log_2(2/8))$$

$$= 0.892$$

$$\text{Gain(Windy)} = 0.940 - 0.892 = 0.048$$

$$\text{Split info(Windy)} : \text{info}([8, 6]) = 0.985$$

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

| 64 | 65 | 68 | 69 | 70 | 71 | 72 | 72 | 75 | 75 | 80 | 81 | 83 | 85 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Yes | No | Yes | Yes | Yes | No | No | Yes | Yes | Yes | No | Yes | Yes | No |

$Info_{Temp}(T) = 6/14 * (-2/6 * \log_2(2/6) - 4/6 * \log_2(4/6))$

$+ 8/14 * (-3/8 * \log_2(3/8) - 5/8 * \log_2(5/8))$

$= 0.939$

Gain(Temp) = 0.940 – 0.939 = 0.001

Split info(Temp) : info([6,8])

$= -6/14 * \log_2(6/14) - 8/14 * \log_2(8/14)$

$= 0.985$

Gain ratio(Temp) = 0.001/0.985 = 0.001

| 65 | 70 | 70 | 70 | 75 | 78 | 80 | 80 | 80 | 85 | 90 | 90 | 95 | 96 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Yes | Yes | Yes | No | Yes | Yes | No | Yes | Yes | No | No | Yes | No | Yes |

$Info_{Humidity}(T) = 5/14 * (-1/5 * \log_2(1/5) - 4/5 * \log_2(4/5))$

$+ 9/14 * (-4/9 * \log_2(4/9) - 5/9 * \log_2(5/9))$

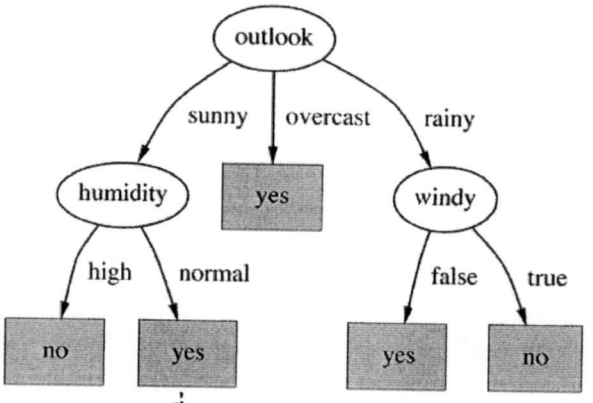
$= 0.895$

Gain(Humidity) = 0.940 – 0.895 = 0.045

Split info(Humidity) : info(5, 9) = $-5/14 * \log_2(5/14) - 9/14 * \log_2(9/14)$ = 0.940

Gain ratio(Humidity) = 0.045/0.940 = 0.048

จากตัวอย่าง เนื่องจาก attribute Outlook มีค่า Gain ratio สูงที่สุด ดังนั้นจึงเลือก attribute Outlook เป็น attribute ที่ใช้ในการแบ่งกลุ่ม ภายหลังจากการทำการแบ่งกลุ่มตัวอย่างไปเรื่อยๆ จนกระทั่งเสร็จเรียบร้อยแล้ว จะได้ต้นไม้ตามรูปด้านล่างนี้




**ภาพที่ 3.3** แสดง Decision Tree

ตัวอย่างกรณีที่ข้อมูลใน training set มีค่าว่างหรือไม่ทราบค่า

สมมติว่า ค่าใน attribute Outlook ใน record ที่ 6 เป็นค่าที่ไม่ทราบค่า ซึ่งแทนด้วย "?" เราจะพิจารณาเฉพาะ record ที่ทราบค่า (ข้อมูล 13 record) ซึ่งสามารถนับจำนวนตัวอย่างในแต่ละกลุ่มได้ดังนี้

**ตารางที่ 3.2** แสดงความถี่ของข้อมูล

| | Play | Don't Play | Total |
|---|---|---|---|
| outlook = sunny | 2 | 3 | 5 |
| overcast | 3 | 0 | 3 |
| rain | 3 | 2 | 5 |
| Total | 8 | 5 | 13 |

คำนวณค่าต่างๆของ attribute Outlook ได้ดังนี้

$$\text{Info}(T) = -8/13 * \log_2(8/13) - 5/13 * \log_2(5/13) = 0.961$$

$$\text{Info}_{\text{Outlook}}(T) = 5/13 * (-2/5 * \log_2(2/5) - 3/5 * \log_2(3/5)) + 3/13 * (-3/3 * \log_2(3/3) - 0/3 * \log_2(0/3)) + 5/13 * (-3/5 * \log_2(3/5) - 2/5 * \log_2(2/5)) = 0.747$$

$$\text{Gain(Outlook)} = 13/14 * (0.961 - 0.747) = 0.199$$

$$\text{Split info(Outlook)} = -5/14 * \log_2(5/14) \text{ (for sunny)} -3/14 * \log_2(3/14) \text{ (for overcast)} -5/14 * \log_2(5/14) \text{ (for rain)} -1/14 * \log_2(1/14) \text{ (for "?")} = 1.809$$

$$\text{Gain ratio(Outlook)} = 0.199/1.809 = 0.110$$

เมื่อ Training set ทั้ง 14 record ถูกแบ่งออกเป็น subset ตาม attribute Outlook ในตัวอย่าง 13 ตัวอย่าง ที่ทราบค่า Outlook จะถูกแบ่งตามปกติ แต่ตัวอย่างที่เหลือ 1 ตัวอย่างที่ไม่ทราบค่าของ Outlook จะถูกแบ่งให้กับทุกๆค่าที่เป็นไปได้ของ attribute Outlook คือ sunny, overcast, rain ด้วยค่า weight เป็น5/13, 3/13, 5/13 ตามลำดับ

ถ้าดู subset หลังจากแบ่งด้วย attribute Outlook เฉพาะใน Outlook ที่เป็น sunny จะประกอบด้วย record ตามตารางดังนี้

**ตารางที่ 3.3** แสดง subset ของ outlook = sunny

| Outlook | Temp (°F) | Humidity (%) | Windy? | Decision | Weight |
|---|---|---|---|---|---|
| sunny | 75 | 70 | true | Play | 1 |
| sunny | 80 | 90 | true | Don't Play | 1 |
| sunny | 85 | 85 | false | Don't Play | 1 |
| sunny | 72 | 95 | false | Don't Play | 1 |
| sunny | 69 | 70 | false | Play | 1 |
| ? | 72 | 90 | true | Play | 5/13 |

ถ้า subset ตัวอย่างนี้ถูกแบ่งต่อตาม attribute Humidity ที่ 75 % จะสามารถแบ่งตัวอย่างออกเป็น 2 ชุด คือ

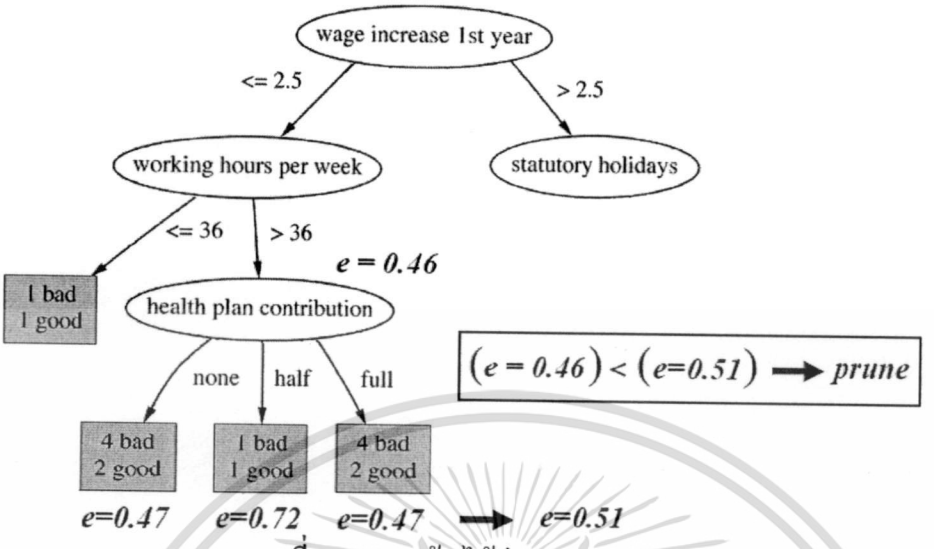
1. Humidity ≤ 75% จะประกอบด้วย ตัวอย่างที่ตัดสินใจ play 2 ตัวอย่าง และ Don't Play 0 ตัวอย่าง

2. Humidity > 75% จะประกอบด้วย ตัวอย่างที่ตัดสินใจ play 5/13 ตัวอย่าง และ Don't Play 3 ตัวอย่าง

เมื่อสร้างเป็นต้นไม้ตัดสินใจ จะได้ต้นไม้ที่มีลักษณะเหมือนเดิม ดังนี้

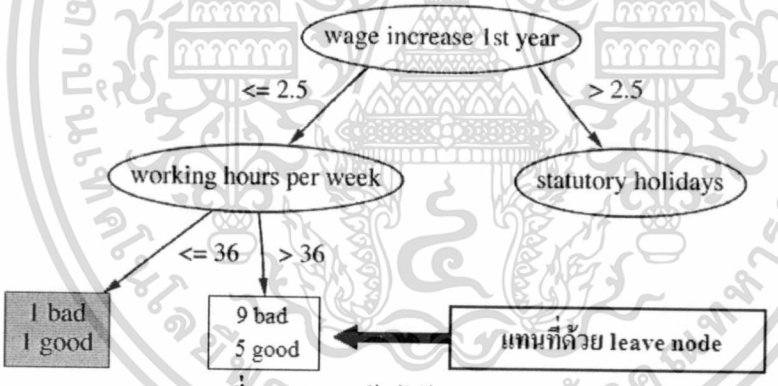
```
outlook = sunny:
|   humidity ≤ 75:  Play (2.0)
|   humidity > 75:  Don't Play (3.4 / 0.4)
outlook = overcast:  Play (3.2)
outlook = rain:
|   windy = true:  Don't Play (2.4 / 0.4)
|   windy = false:  Play (3.0)
```

## 3.5 ตัวอย่างการตัดกิ่ง




ภาพที่ 3.4 แสดงต้นไม้ก่อนการ prune

ดังนั้น Combined using ratio 6:2:6 gives 0.51 จะได้



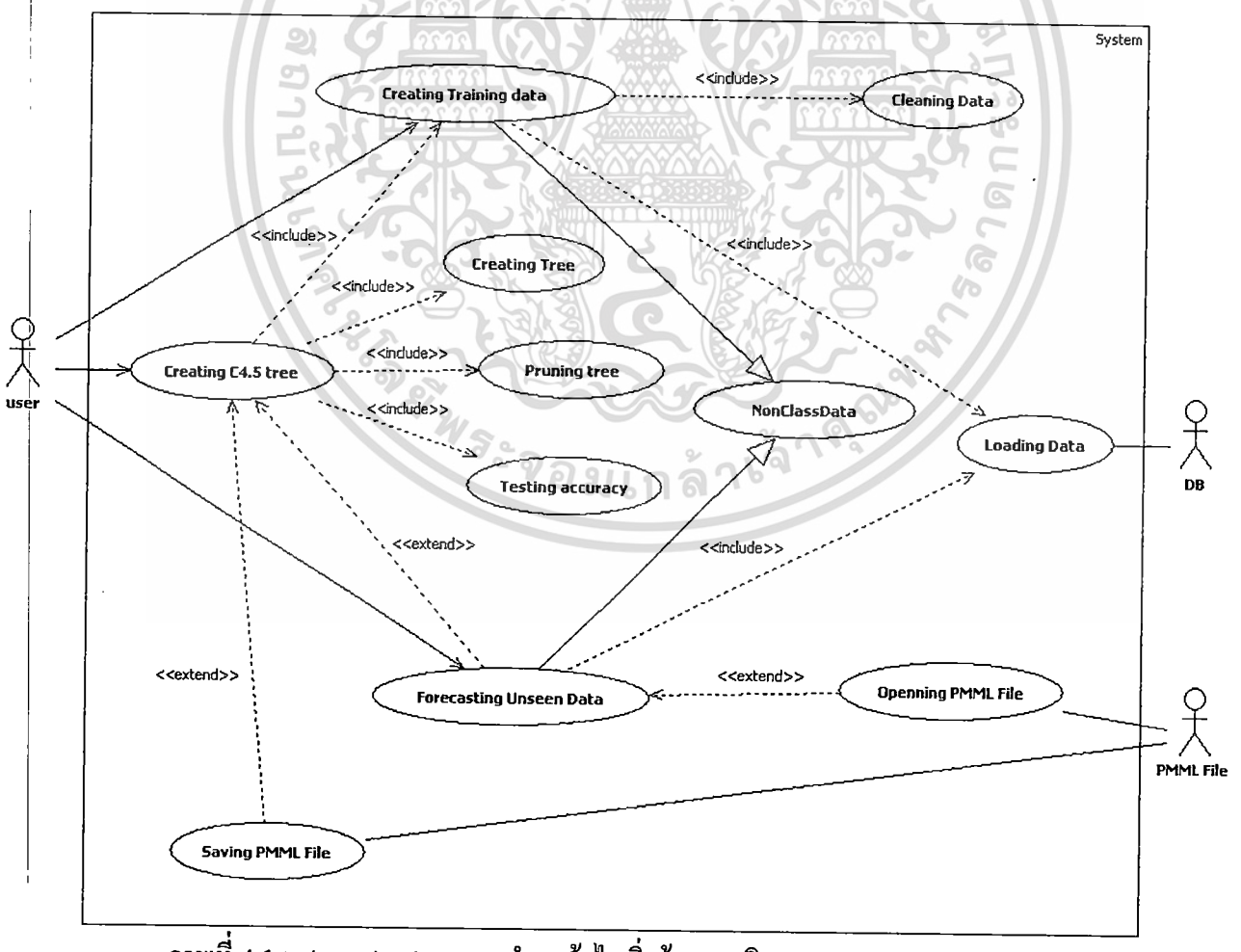

ภาพที่ 3.5 แสดงต้นไม้หลังการ prune

# บทที่ 4
# การวิเคราะห์ และออกแบบระบบ

เนื้อหาในบทนี้จะกล่าวถึงการวิเคราะห์ และออกแบบกระบวนการสร้างระบบดาต้าไมนิ่งด้วยวิธี Classification โดยอาศัยเทคนิค C4.5 Decision Tree โดยใช้ยูสเคสไดอะแกรม (Use Case Diagram), คลาสไดอะแกรม (Class Diagram) และซีเควนไดอะแกรม (Sequence Diagram) ในการอธิบายการทำงานของระบบ ดังนี้

## 4.1 ยูสเคสไดอะแกรม

ยูสเคสไดอะแกรมของการพัฒนาโปรแกรมดาต้าไมนิ่งแบบ Classification โดยใช้อัลกอริทึม C4.5 Decision Tree ของโครงการพัฒนาระบบฉบับนี้ แสดงได้ดังภาพที่ 4.1




**ภาพที่ 4.1** แสดงยูสเคสของการทำดาต้าไมนิ่งด้วยเทคนิค C4.5 Decision Tree

จากภาพที่ 4.1 เป็นยูสเคสไดอะแกรมที่ใช้แสดงถึงขอบเขตการพัฒนาระบบดาต้าไมนิ่งแบบ Classification โดยใช้อัลกอริทึม C4.5 Decision Tree ซึ่งประกอบไปด้วยรายละเอียดดังนี้ คือ

แอกเตอร์ที่เกี่ยวข้องกับระบบ มี 2 แอกเตอร์ ดังนี้ คือ

1) Use คือ ผู้ใช้งานระบบ
2) DB คือ ระบบฐานข้อมูลที่ทำหน้าที่ในการจัดเก็บข้อมูลที่จะนำมาใช้ในการสร้างโมเดลต้นไม้ (Training Data) หรือจัดเก็บข้อมูลที่ไม่เคยพบเห็นมาก่อน (Unseen Data) เพื่อให้โมเดลต้นไม้ทำการทำนายผลข้อมูล
3) PMML File คือ เอกสาร PMML ที่ทำหน้าที่ในการจัดเก็บโมเดลต้นไม้ C4.5 Decision Tree

ยูสเคสที่เกี่ยวข้องกับระบบมีดังนี้ คือ

1) Creating Training Data เป็นงานที่ทำหน้าที่ในการนำข้อมูลเข้าสู่ระบบ ตั้งแต่เชื่อมต่อฐานข้อมูล ไปจนกระทั่งจัดเก็บข้อมูล และ metadata ของ training data ที่จำเป็นต้องใช้ในการสร้างแบบจำลองต้นไม้
2) Cleaning Data เป็นงานที่ทำหน้าที่ในการทำความสะอาดข้อมูลที่ถูกนำมาใช้เป็น training data เช่น จัดการกับข้อมูลที่เป็นค่าว่าง และข้อมูลที่ไม่ทราบค่า
3) Creating C4.5 Tree ทำหน้าที่ในการสร้างแบบจำลองต้นไม้ C4.5 Decision Tree โดยมีขั้นตอน คือ
    3.1 Creating Tree ทำหน้าที่ในการสร้างต้นไม้ ด้วยเทคนิค C4.5 Decision Tree โดยระบบจะทำการแบ่งข้อมูลออกเป็น 2 ส่วน คือ training data และ testing data ก่อนทำการสร้างโมเดล ในส่วนของ testing data นั้น เกิดจากการสุ่มข้อมูลจาก training data ที่ได้หลังจากทำความสะอาดข้อมูลขึ้นมาเป็นจำนวน 30% ของจำนวนทั้งหมดใน training data ซึ่ง testing data นี้จะถูกนำไปใช้ในการคำนวณหาค่าความถูกต้องแม่นยำในการทำนายผลข้อมูลของแบบจำลอง ภายหลังจากสร้างแบบจำลองเรียบร้อยแล้ว และในส่วนของ training data ที่เหลือจากการสุ่ม จะถูกนำไปใช้ในการสร้างแบบจำลองต้นไม้ต่อไป
    3.2 Pruning Tree ทำหน้าที่ในการตัดกิ่งต้นไม้ที่ได้ ด้วยวิธีการของ EBP
    3.3 Testing accuracy ทำหน้าที่ทดสอบความแม่นยำในการทำนายผลข้อมูลของแบบจำลองต้นไม้ที่ได้ โดยนำ testing data ที่ได้ทำการสุ่มไว้แล้วก่อนหน้ามาทดสอบการทำนายผล และคำนวณหาค่าความถูกต้องในการทำนาย โดยคิดออกมาเป็นเปอร์เซ็นต์ เทียบกับจำนวนของ testing data ทั้งหมด
    3.4 Saving PMML File ทำหน้าที่ในการบันทึกแบบจำลองต้นไม้ ในลักษณะของเอกสาร PMML เพื่อนำไปใช้ให้เกิดประโยชน์ต่อไปในภายหลัง

4) Forecasting Unseen Data ทำหน้าที่ในการทำนายผลข้อมูลที่ไม่เคยพบเห็นมาก่อน โดยผู้ใช้อาจจะให้ระบบทำการทำนายผลต่อภายหลังจากการสร้างแบบจำลองต้นไม้เรียบร้อยแล้ว หรืออาจจะให้ทำนายผลหลังจากทำการเปิดอ่านแบบจำลองต้นไม้จากเอกสาร PMML ในส่วนของการทำงานของ use case Opening PMML File เข้าสู่ระบบแล้ว ซึ่งผู้ใช้จะทำการนำเข้าข้อมูลที่ต้องการนำมาใช้ในการทำนายผล (Unseen data) เข้าสู่ระบบ ด้วยการทำงานของ use case Loading Data และทำนายผลข้อมูลออกมา

## 4.2 คลาสไดอะแกรม

จากภาพที่ 4.2 เป็นภาพที่แสดงคลาสไดอะแกรม ซึ่งสามารถอธิบายได้ว่า

1. คลาส LoadingData เป็นคลาสที่ใช้ในการนำเข้าข้อมูลจากฐานข้อมูลเข้าสู่ระบบ ไม่ว่าจะเป็น training data หรือ unseen data ก็ตาม โดยผู้ใช้จะต้องระบุ driver, url, username, password ของฐานข้อมูลที่จัดเก็บข้อมูลที่ต้องการนำเข้าสู่ระบบด้วย

2. คลาส NonClassData เป็นคลาสที่ใช้ในการจัดเก็บข้อมูลที่ถูกนำเข้ามาในระบบ โดยจะเก็บข้อมูลทุกอย่างที่เกี่ยวข้องกับ non class data ได้แก่ ตัวข้อมูลที่ถูกโหลดเข้ามา, เก็บ metadata

3. คลาส TrainingData เป็นคลาสที่มีการสืบทอดต่อมาจากคลาส NonClassData โดยจะมีส่วนของการจัดเก็บข้อมูลที่เป็น class data เพิ่มขึ้นมา และมีการจัดเก็บข้อมูลในลักษณะเช่นเดียวกันกับ non class data

4. คลาส UnseenData เป็นคลาสที่มีการสืบทอดมาจากคลาส NonClassData เช่นเดียวกันกับคลาส TrainingData แต่ในส่วนของการจัดเก็บข้อมูลที่เพิ่มขึ้นมานั้นจะแตกต่างกัน โดยจะเป็นการเก็บข้อมูลผลลัพธ์การทำนายผลแบบจำลองต้นไม้แทน

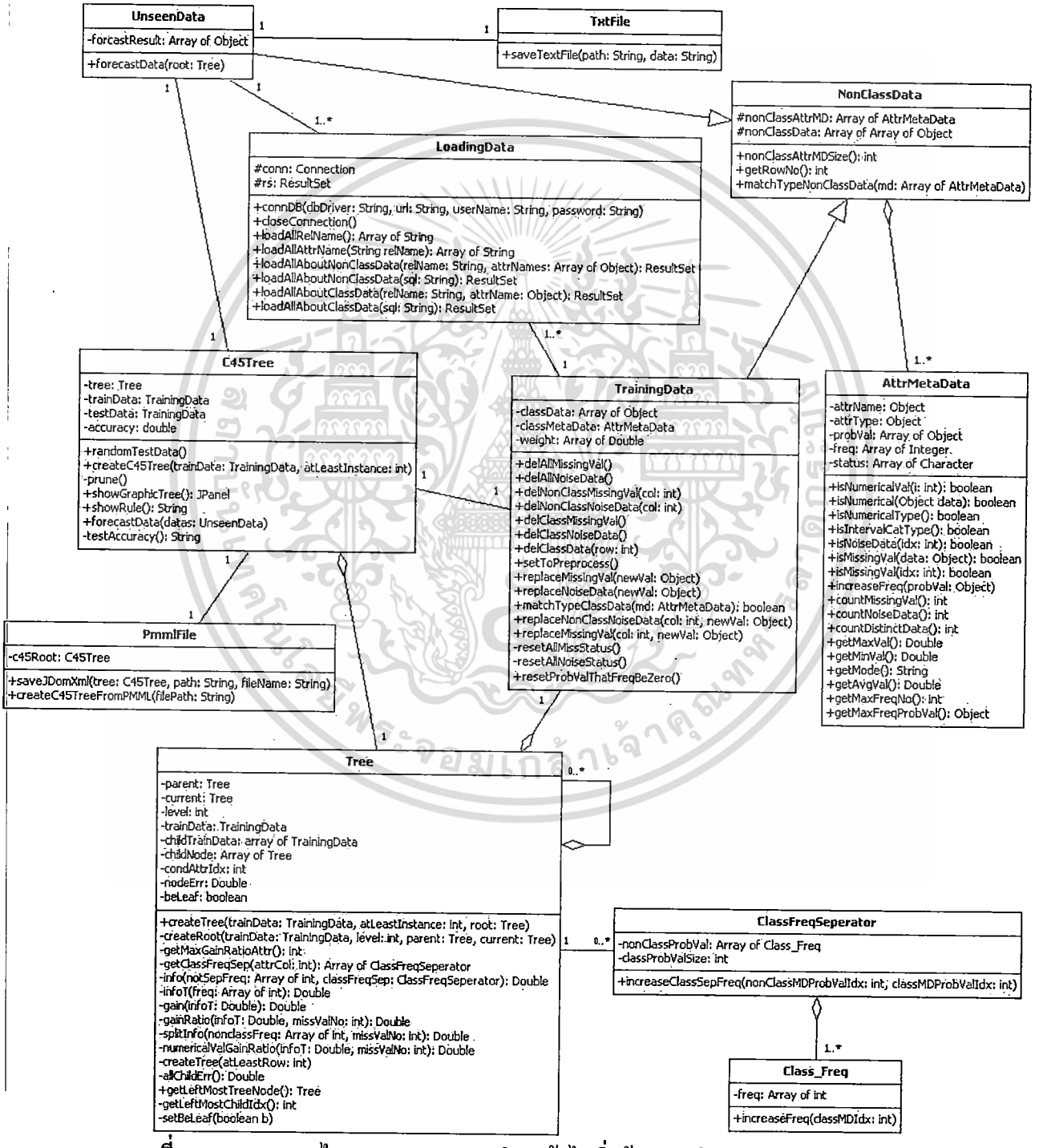
5. คลาส AttrMetaData เป็นคลาสที่ใช้จัดเก็บข้อมูลของข้อมูลที่เกี่ยวข้องกับการสร้างแบบจำลองต้นไม้ หรือการทำนายผลของแบบจำลองต้นไม้ โดยที่ metadata จะประกอบไปด้วยชื่อ attribute, ชนิดข้อมูล, ค่าที่เป็นไปได้, ความถี่ของจำนวนค่าที่เป็นไปได้ในแต่ละค่า โดยความถี่เหล่านั้นจะถูกนำมาใช้ในการคำนวณหาค่า Gain Ratio ในขั้นตอนของการสร้างแบบจำลองต้นไม้

6. คลาส C45Tree เป็นคลาสที่ใช้ในการสร้างแบบจำลองต้นไม้ โดยนำข้อมูลจาก training data มาใช้ในการสร้างแบบจำลอง โดยก่อนสร้างแบบจำลอง ระบบจะทำการแบ่งข้อมูลออกเป็น 2 กลุ่มก่อน คือ กลุ่มของ training data และ testing data จากนั้นระบบจะทำการคำนวณหาค่าความถูกต้องแม่นยำของแบบจำลองต้นไม้ที่สร้างขึ้น โดยคิดเป็นเปอร์เซนต์

7. คลาส Tree เป็นคลาสที่ใช้ในการจัดเก็บข้อมูลต่างๆ ในแต่ละโหนดของต้นไม้ ทั้งก่อนการแตกกิ่ง และหลังจากการแตกกิ่งของโหนดนั้นๆ เช่น เก็บ training data ที่เข้าไปอยู่ในแต่ละ

โหนดภายหลังจากการแตกกิ่งออกมาจากโหนดพ่อม, เก็บ attribute ที่มีค่า Gain Ratio สูงที่สุด เป็นต้น

8. คลาส Class_Freq และ คลาส ClassFreqSeperator เป็นคลาสที่นำมาใช้ในการจัดเก็บข้อมูลความถี่ของค่าที่เป็นไปได้ใน class attribute ของแต่ละกลุ่มค่าที่เป็นไปได้ใน non class attribute นั้นๆ เพื่อใช้ในการคำนวณในส่วนของขั้นตอนการคำนวณหาค่า Gain Ratio




**ภาพที่ 4.2** แสดงคลาสไดอะแกรมของการทำดาต้าไมนิ่งด้วยเทคนิค C4.5 Tree

**ตารางที่ 4.1** แสดงการอธิบาย Class LoadingData ด้วย CRC (Class Responsibility Collaboration)

Front:

| Class Name: LoadingData | |
|---|---|
| Superclasses: - | |
| Subclasses: - | |
| Responsibilities (หน้าที่ของคลาส):<br>เชื่อมต่อฐานข้อมูล และนำข้อมูลเข้าสู่ระบบ | Collaborators (ทำงานร่วมกับ):<br>TrainingData, UnseenData |

Back:

| Attributes:<br>conn (เก็บการเชื่อมต่อฐานข้อมูล) (Connection)<br>rs (เก็บข้อมูลจากฐานข้อมูล) (ResultSet) |
|---|
| Relationships:<br>Aggregation (has-parts): -<br>Other Associations: - |

**ตารางที่ 4.2** แสดงการอธิบาย Class NonClassData ด้วย CRC

Front:

| Class Name: NonClassData | |
|---|---|
| Superclasses: - | |
| Subclasses: TrainingData, UnseenData | |
| Responsibilities (หน้าที่ของคลาส):<br>จัดเก็บข้อมูล และ metadata ของข้อมูลที่ไม่ใช่กลุ่มเป้าหมาย | Collaborators (ทำงานร่วมกับ): |

Back:

| Attributes:<br>nonClassAttrMD (เก็บรายละเอียดของข้อมูล) (AttrMetaData)<br>nonClassData (เก็บข้อมูล) (Object) |
|---|
| Relationships:<br>Aggregation (has-parts): AttrMetaData<br>Other Associations: - |

**ตารางที่ 4.3** แสดงการอธิบาย Class TrainingData ด้วย CRC

Front:

| Class Name: TrainingData | |
|---|---|
| Superclasses: NonClassData | |
| Subclasses: - | |
| Responsibilities (หน้าที่ของคลาส):<br>เก็บข้อมูล และ metadata ของข้อมูลฝึก ทั้งข้อมูลที่เป็นกลุ่มเป้าหมาย และไม่ใช่กลุ่มเป้าหมาย | Collaborators (ทำงานร่วมกับ):<br>LoadingData, C45Tree |

Back:

| Attributes:<br>classData (เก็บข้อมูลที่เป็นกลุ่มเป้าหมาย) (Array of Object)<br>classMetaData (เก็บ metadata ข้อมูลที่เป็นกลุ่มเป้าหมาย) (AttrMetaData)<br>weight (เก็บค่าน้ำหนักของข้อมูล) (Array of Double) |
|---|
| Relationships:<br>Aggregation (has-parts): -<br>Other Associations: - |

**ตารางที่ 4.4** แสดงการอธิบาย Class UnseenData ด้วย CRC

Front:

| Class Name: UnseenData | |
|---|---|
| Superclasses: NonClassData | |
| Subclasses: - | |
| Responsibilities (หน้าที่ของคลาส):<br>เก็บข้อมูล และ metadata ของข้อมูลที่ไม่เคยเห็นมาก่อน และผลลัพธ์ในการทำนาย | Collaborators (ทำงานร่วมกับ):<br>LoadingData, C45Tree, TxtFile |

Back:

| Attributes:<br>forecastResult (เก็บผลลัพธ์การทำนายของโมเดลต้นไม้) (Connection) |
|---|
| Relationships:<br>Aggregation (has-parts): -<br>Other Associations: - |

**ตารางที่ 4.5** แสดงการอธิบาย Class AttrMetaData ด้วย CRC

Front:

| Class Name: AttrMetaData | |
|---|---|
| Superclasses: - | |
| Subclasses: - | |
| Responsibilities (หน้าที่ของคลาส):<br>เก็บ metadata ของข้อมูล | Collaborators (ทำงานร่วมกับ): |

Back:

| Attributes:<br>attrName (ชื่อ attribute) (Object)<br>attrType (ชนิดข้อมูล) (Object: "Categorical", "Numerical")<br>probVal (ค่าที่เป็นไปได้ของข้อมูล) (Array of Object)<br>freq (ความถี่ของข้อมูล) (Array of Integer)<br>status (สถานะของข้อมูล) (Array of Character: "M", "N", "I") |
|---|
| Relationships:<br>Aggregation (has-parts): -<br>Other Associations: - |

**ตารางที่ 4.6** แสดงการอธิบาย Class TxtFile ด้วย CRC

Front:

| Class Name: TxtFile | |
|---|---|
| Superclasses: - | |
| Subclasses: - | |
| Responsibilities (หน้าที่ของคลาส):<br>บันทึก Unseen Data และผลลัพธ์การทำนาย | Collaborators (ทำงานร่วมกับ):<br>UnseenData |

Back:

| Attributes: |
|---|
| Relationships:<br>Aggregation (has-parts): -<br>Other Associations: - |

**ตารางที่ 4.7** แสดงการอธิบาย Class PmmlFile ด้วย CRC

Front:

| Class Name: PmmlFile | |
|---|---|
| Superclasses: - | |
| Subclasses: - | |
| Responsibilities (หน้าที่ของคลาส):<br>อ่าน หรือบันทึกโมเดลต้นไม้ในรูปเอกสาร PMML | Collaborators (ทำงานร่วมกับ):<br>C45Tree |

Back:

| Attributes:<br>c45Root (เก็บโมเดลต้นไม้ C4.5) (C45Tree) |
|---|
| Relationships:<br>Aggregation (has-parts): -<br>Other Associations: - |

**ตารางที่ 4.8** แสดงการอธิบาย Class C45Tree ด้วย CRC

Front:

| Class Name: C45Tree | |
|---|---|
| Superclasses: - | |
| Subclasses: - | |
| Responsibilities (หน้าที่ของคลาส):<br>สร้างโมเดลต้นไม้ C4.5 , prune, ทดสอบความแม่นยำ, สร้างกฎ | Collaborators (ทำงานร่วมกับ):<br>TrainingData, UnseenData, PmmlFile |

Back:

| Attributes:<br>tree (เก็บโมเดลต้นไม้) (Tree)<br>trainData (เก็บชุดข้อมูลฝึก) (TrainingData)<br>testData (เก็บชุดข้อมูลที่ใช้ทดสอบความแม่นยำ) (TrainingData)<br>accuracy (ความแม่นยำของโมเดล) (double) |
|---|
| Relationships:<br>Aggregation (has-parts): Tree<br>Other Associations: - |

**ตารางที่ 4.9** แสดงการอธิบาย Class Class_Freq ด้วย CRC

Front:

| Class Name: Class_Freq | |
|---|---|
| Superclasses: - | |
| Subclasses: - | |
| Responsibilities (หน้าที่ของคลาส):<br>เก็บความถี่ค่าข้อมูลของ Class Data ในแต่ละค่าข้อมูลของ Non Class Data | Collaborators (ทำงานร่วมกับ): |

Back:

| Attributes:<br>freq (เก็บความถี่ค่าข้อมูลของ Class Data ) (Array of integer) |
|---|
| Relationships:<br>Aggregation (has-parts): -<br>Other Associations: - |

**ตารางที่ 4.10** แสดงการอธิบาย Class ClassFreqSeperator ด้วย CRC

Front:

| Class Name: ClassFreqSeperator | |
|---|---|
| Superclasses: - | |
| Subclasses: - | |
| Responsibilities (หน้าที่ของคลาส):<br>เก็บความถี่ค่าข้อมูลของ Class Data ในทุกค่าข้อมูลของ Non Class Data แต่ละค่า | Collaborators (ทำงานร่วมกับ):<br>Tree |

Back:

| Attributes:<br>nonClassProbVal (เก็บความถี่ค่าข้อมูลของ Class Data ในทุกค่าข้อมูลของ Non Class Data) (Array of Class_Freq)<br>classProbValSize (เก็บจำนวนค่าข้อมูลของ Class Data) (int) |
|---|
| Relationships:<br>Aggregation (has-parts): Class_Freq<br>Other Associations: - |

**ตารางที่ 4.11** แสดงการอธิบาย Class Tree ด้วย CRC

Front:

| Class Name: Tree | |
|---|---|
| Superclasses: - | |
| Subclasses: - | |
| Responsibilities (หน้าที่ของคลาส):<br>สร้างโมเดลต้นไม้ก่อนตัดกิ่ง (prune) | Collaborators (ทำงานร่วมกับ):<br>ClassFreqSeperator |

Back:

Attributes:

parent (เก็บตำแหน่งของ parent node) (Tree)

current (เก็บตำแหน่งของ node ปัจจุบัน) (Tree)

level (เก็บ level ของ node ปัจจุบัน) (Integer)

trainData (เก็บ Training Data ที่ node ปัจจุบันใช้ในการแตกกิ่ง) (TrainingData)

childTrainData (เก็บ Training Data ของ node ลูก) (Array of TrainingData)

childNode (เก็บตำแหน่งของ node ลูก) (Array of Tree)

condAttrIdx (เก็บ idex ของ attribute ที่มีค่า Gain Ratio สูงสุด) (Integer)

nodeErr (เก็บค่า Error ของ node ใช้เปรียบเทียบตอนตัดกิ่งต้นไม้) (Double)

beLeaf (เก็บสถานะของ node ว่าเป็นใบหรือไม่) (boolean)

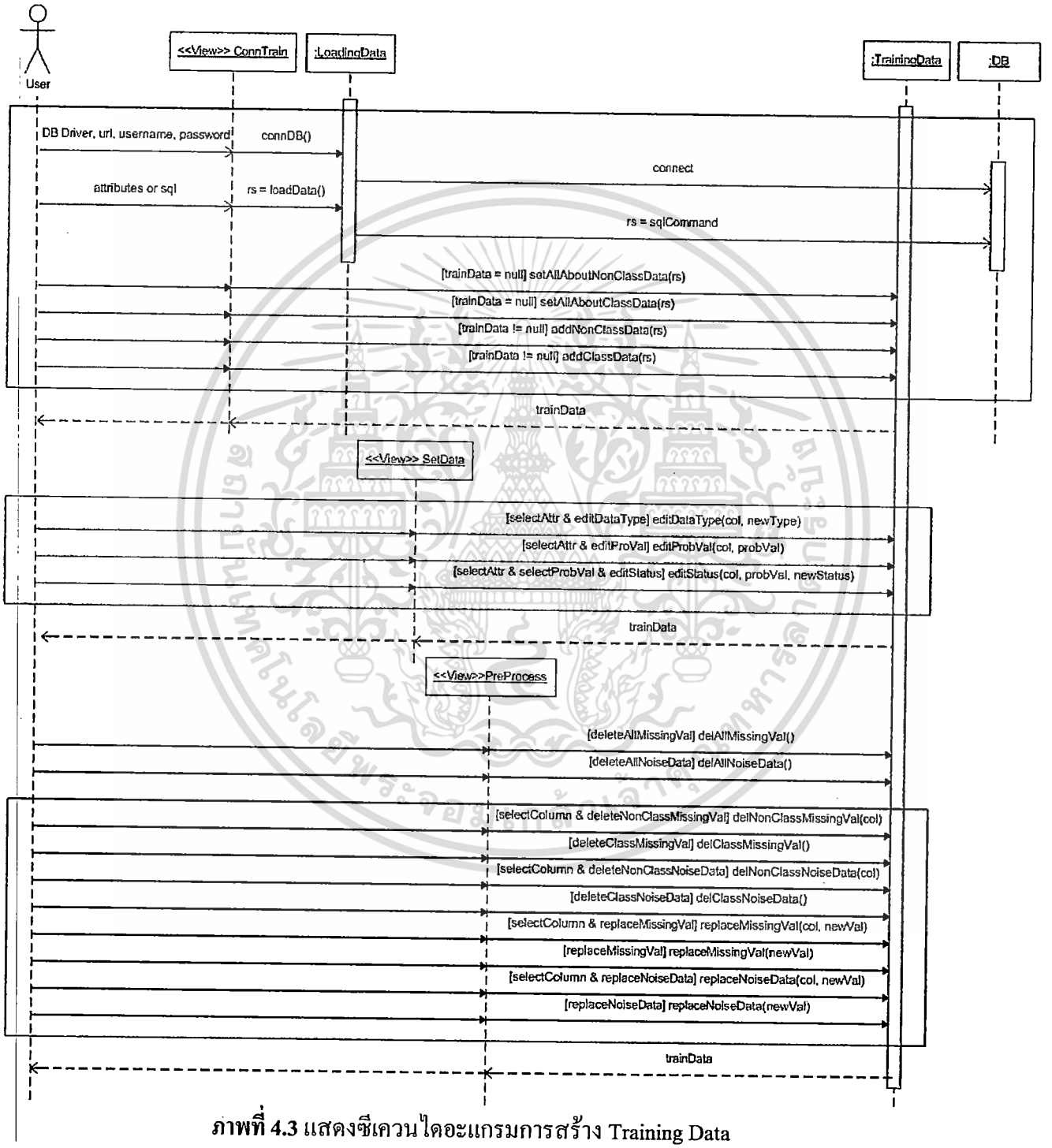
Relationships:

Aggregation (has-parts): Tree

Other Associations: -

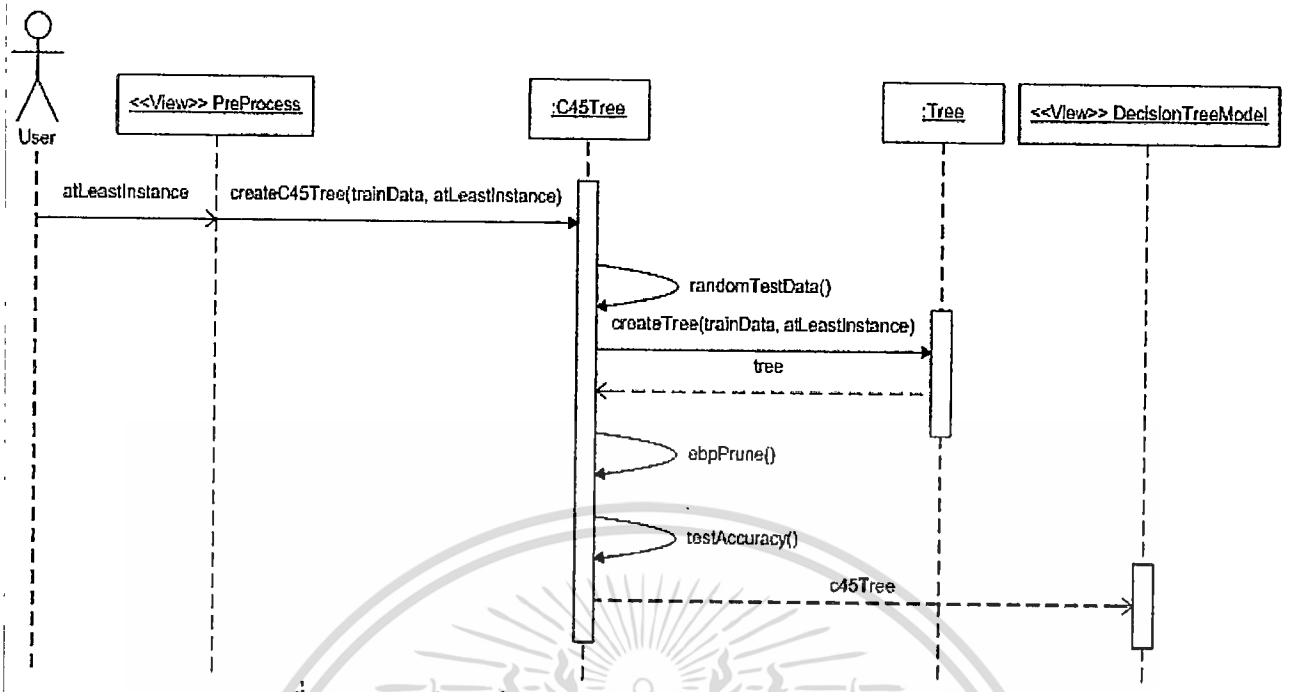
## 4.3 ซีเควนไดอะแกรม

จากภาพที่ 4.3 แสดงซีเควนไดอะแกรม อธิบายการสร้าง Training Data โดยเริ่มตั้งแต่การเชื่อมต่อฐานข้อมูล, การนำเข้าข้อมูล และ metadata ไปจนถึง การทำความสะอาดข้อมูล



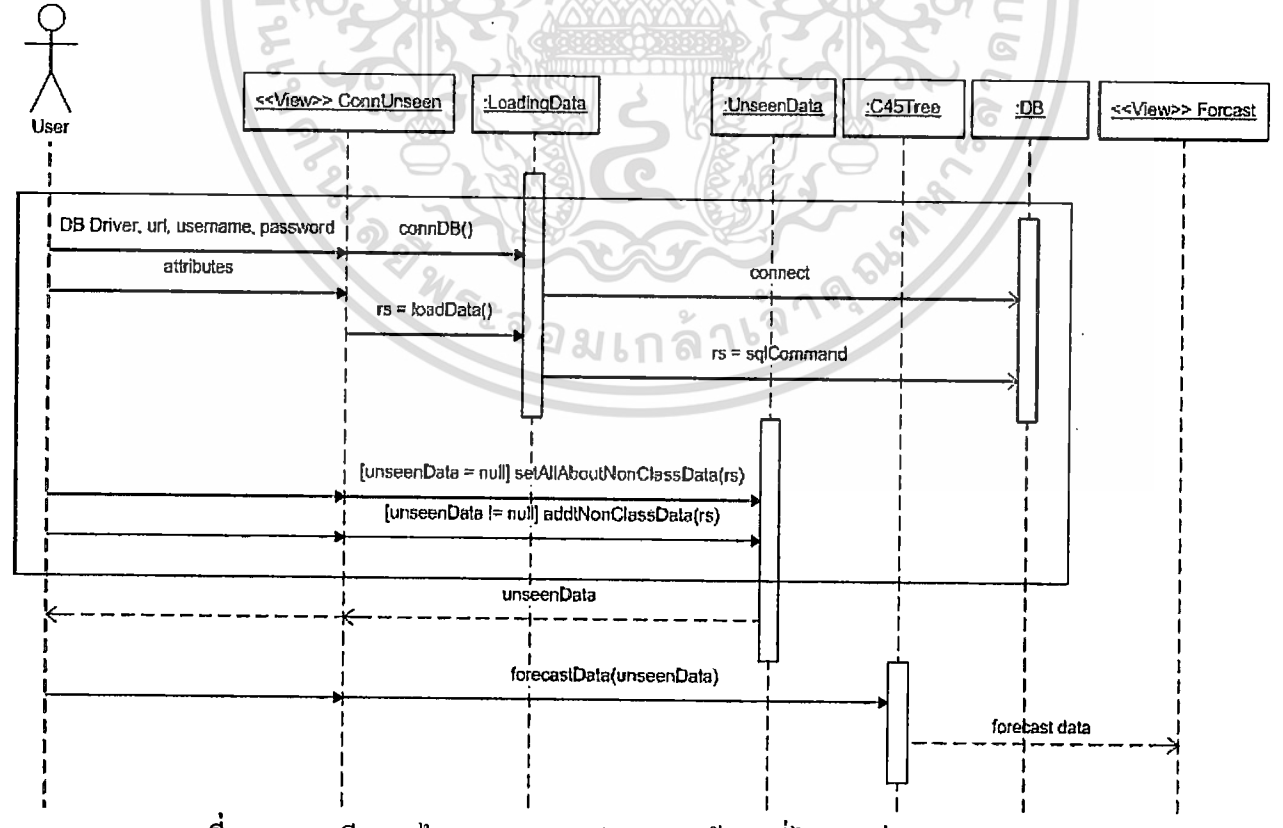

**ภาพที่ 4.3** แสดงซีเควนไดอะแกรมการสร้าง Training Data

จากภาพที่ 4.4 แสดงซีเควนไดอะแกรม อธิบายการสร้างโมเดลต้นไม้ C4.5 ไปจนถึงการทดสอบความแม่นยำในการทำนายผลข้อมูลของโมเดลต้นไม้ที่สร้างขึ้น



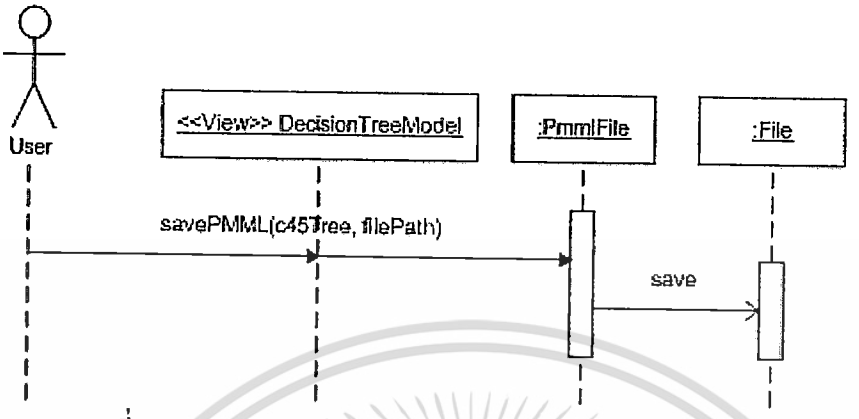

**ภาพที่ 4.4** แสดงซีเควนไดอะแกรมการสร้างโมเดลต้นไม้ C4.5

จากภาพที่ 4.5 แสดงซีเควนไดอะแกรม อธิบายการทำนายผลข้อมูลที่ไม่เคยเห็นมาก่อน โดยเริ่มตั้งแต่การเชื่อมต่อกับฐานข้อมูล เพื่อนำเข้า Unseen Data จนกระทั่งทำนายผล



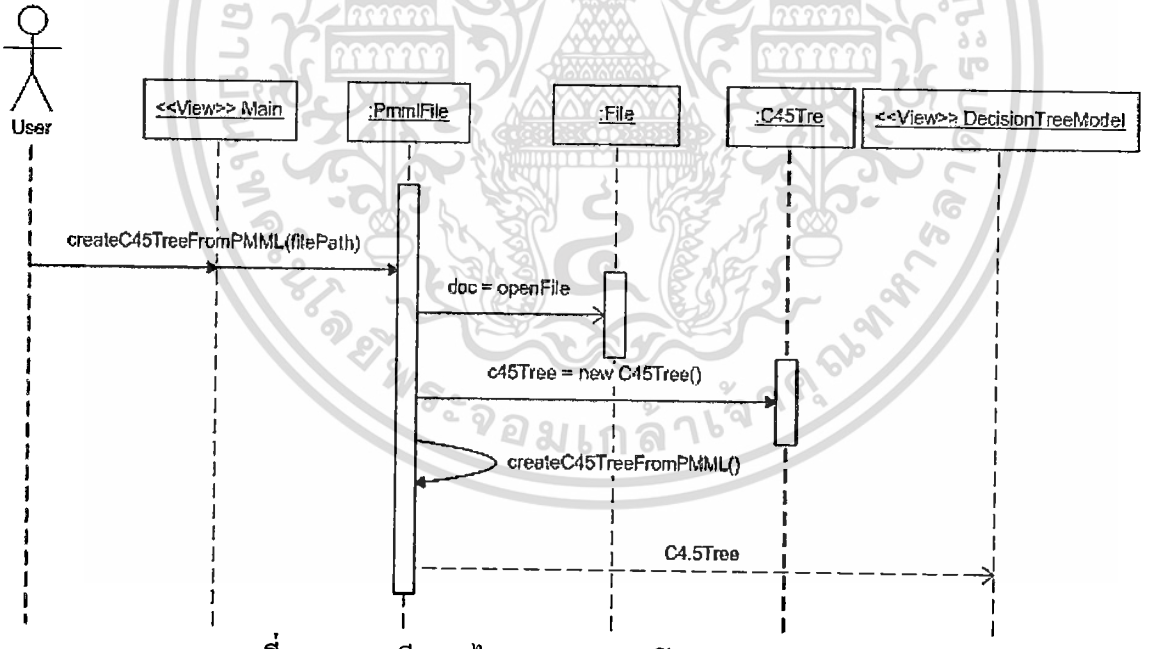

**ภาพที่ 4.5** แสดงซีเควนไดอะแกรมการทำนายผลข้อมูลที่ไม่เคยเห็นมาก่อน

จากภาพที่ 4.6 แสดงซีเควนไดอะแกรม อธิบายการบันทึกโมเดลต้นไม้ที่ได้จากการทำดาต้าไปนิ่งให้อยู่ในรูปของเอกสาร PMML



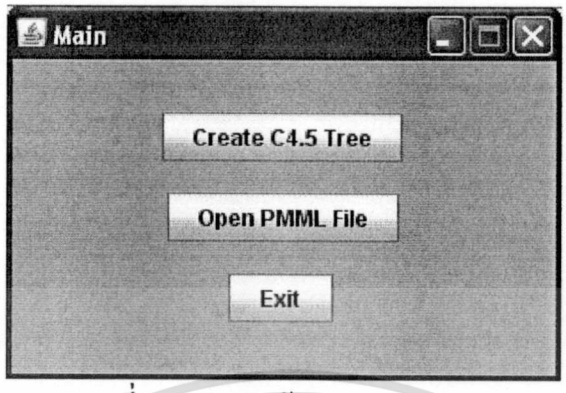

**ภาพที่ 4.6** แสดงซีเควนไดอะแกรมการบันทึกแบบจำลองต้นไม้

จากภาพที่ 4.7 แสดงซีเควนไดอะแกรม อธิบายการเปิดอ่านเอกสาร PMML เข้ามาในระบบ เพื่อใช้ในการทำนายผลข้อมูลที่ไม่เคยพบเห็นมาก่อน



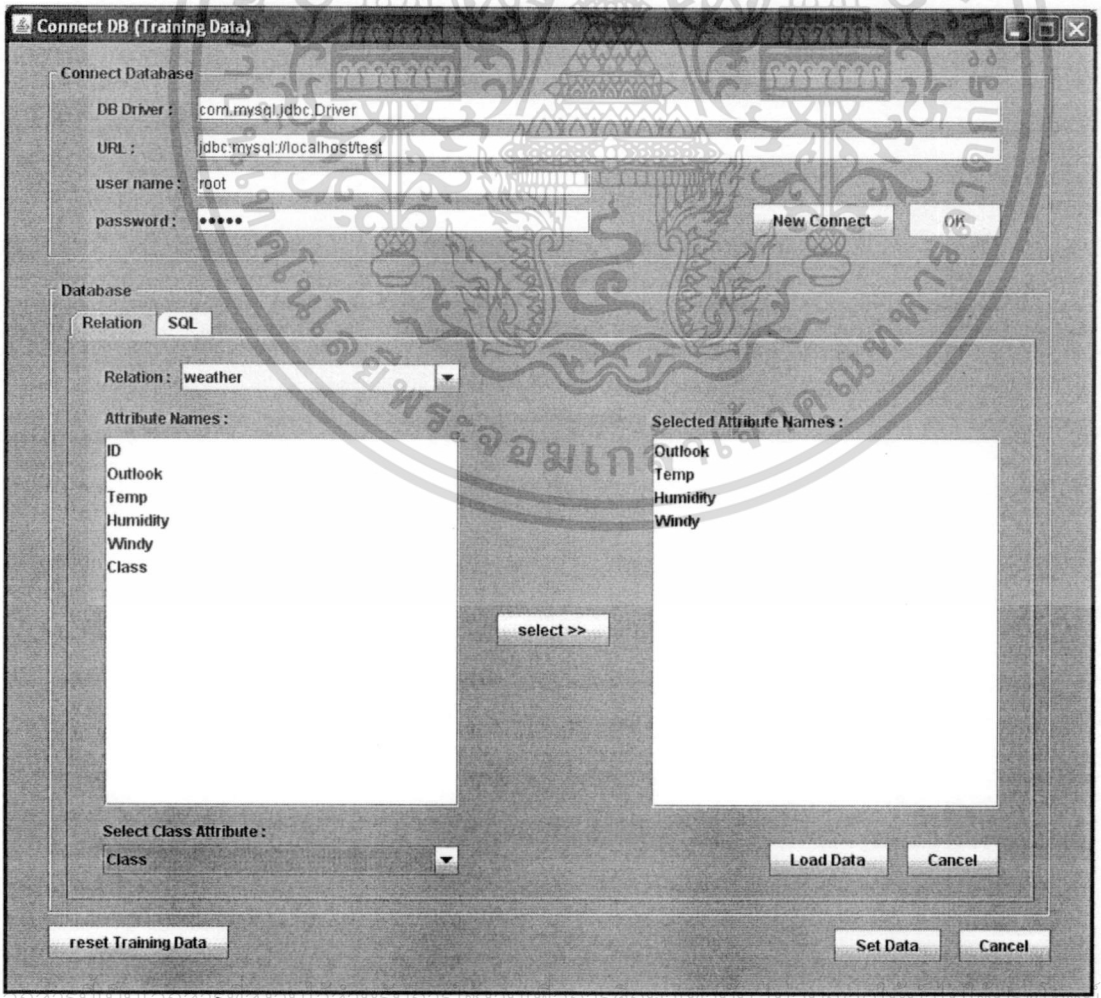

**ภาพที่ 4.7** แสดงซีเควนไดอะแกรมการเปิดเอกสาร PMML

## 4.4 ส่วนติดต่อกับผู้ใช้




**ภาพที่ 4.8** แสดงหน้าจอแรกของระบบ

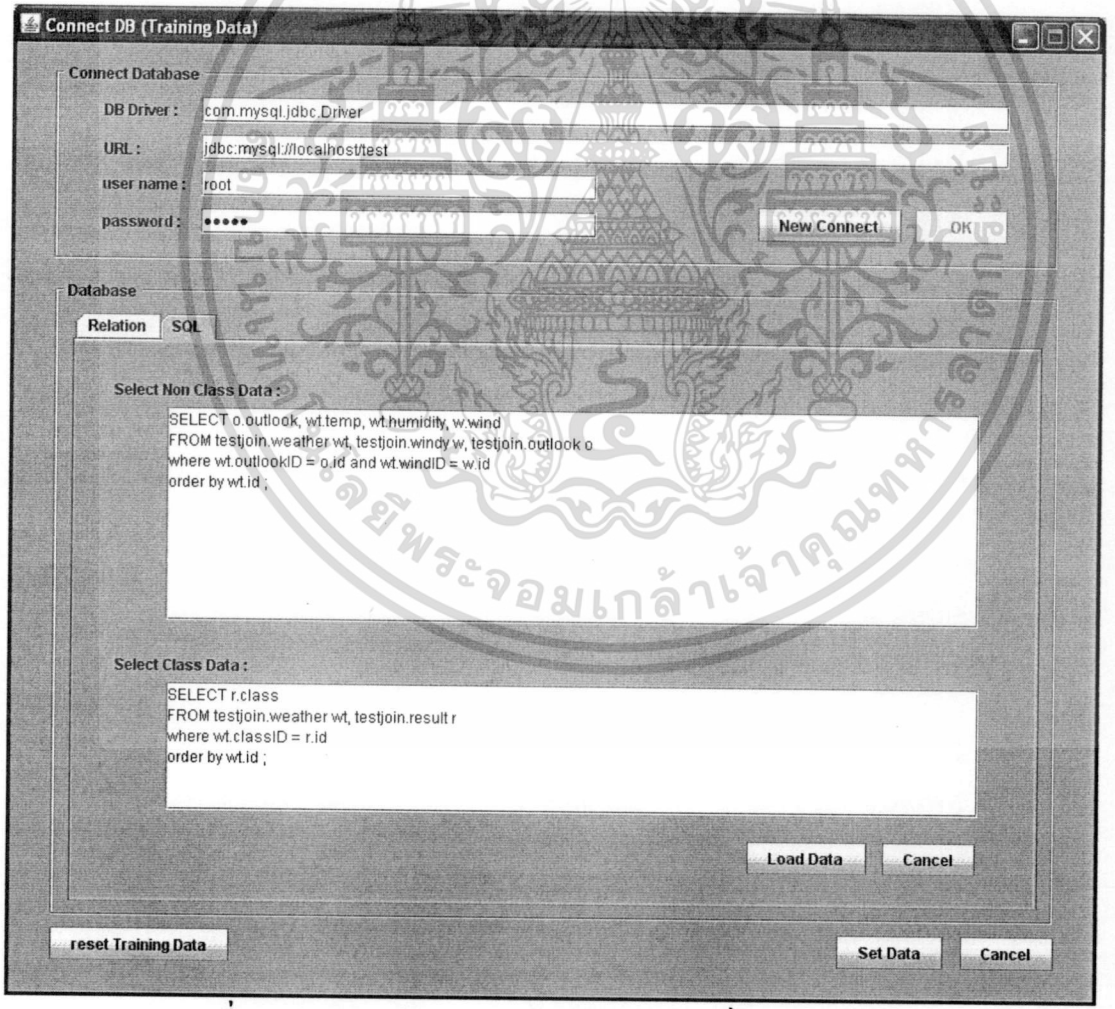
จากภาพที่ 4.8 แสดงหน้าจอแรกของระบบ สามารถอธิบายได้ว่า ในส่วนของปุ่ม Create C4.5 Tree เป็นส่วนของการสร้างแบบจำลองต้นไม้ C4.5 Tree และปุ่ม Open PMML File จะเป็นส่วนของการเปิดแบบจำลองต้นไม้ที่ถูกจัดเก็บอยู่ในรูปเอกสาร PMML ขึ้นมาเพื่อทำนายผลข้อมูล




**ภาพที่ 4.9** แสดงหน้าจอการนำข้อมูลเข้าสู่ระบบซึ่งแสดงแทบ Relation

จากภาพที่ 4.9 แสดงหน้าจอการนำข้อมูลเข้าสู่ระบบซึ่งแสดงแทบ Relation สามารถอธิบายได้ว่า ผู้ใช้จะต้องทำการติดต่อกับฐานข้อมูล เพื่อดึงข้อมูลที่ต้องการนำมาใช้เป็นชุดข้อมูลฝึก (Training Data Set) โดยผู้ใช้จะต้องป้อน Driver, URL, User Name (ถ้ามี) และ Password (ถ้ามี) เมื่อระบบทำการเชื่อมต่อกับฐานข้อมูลสำเร็จแล้ว ผู้ใช้จะสามารถทำการดึงข้อมูลในตารางฐานข้อมูลออกมาได้ 2 วิธีด้วยกัน คือ

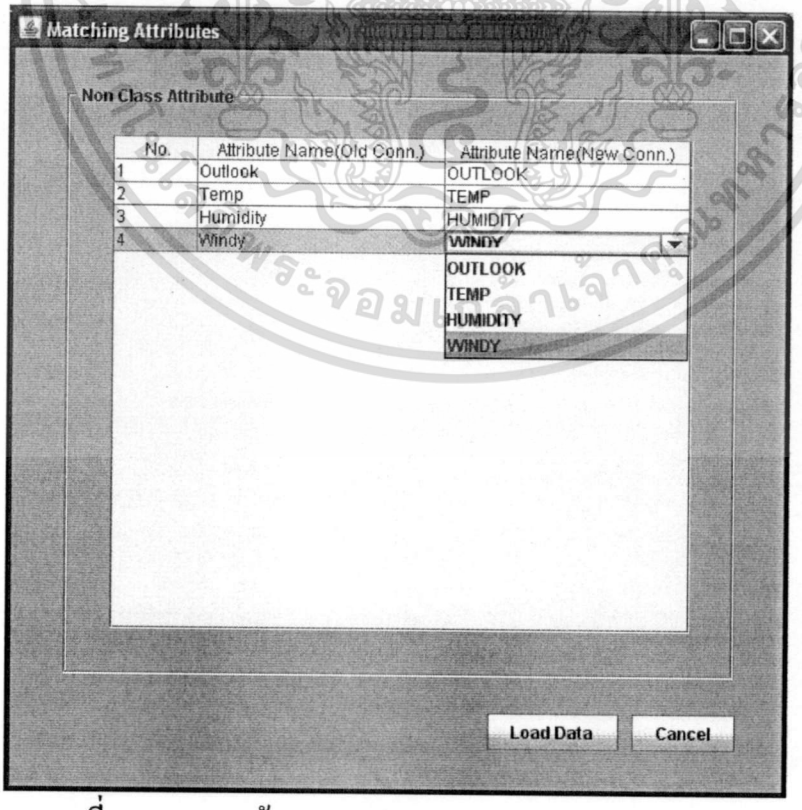
1. ระบบจะทำการแสดงชื่อตารางฐานข้อมูล (Relation) และชื่อ attribute ในตาราง (Attribute Names) ไว้ในส่วนของแทบ Relation เพื่อให้ผู้ใช้สามารถคลิกเลือกตารางและ attribute ที่ต้องการได้โดยสะดวก (ภาพที่ 4.9)
2. ในส่วนของแทบ SQL เป็นส่วนที่ให้ผู้ใช้ใช้ในการสร้างภาษา SQL เพื่อดึงข้อมูลจากฐานข้อมูล เช่น ในกรณีที่ข้อมูลที่ต้องการใช้นั้นต้องมาจากการ join กันของหลายๆ ตาราง (ภาพที่ 4.10)



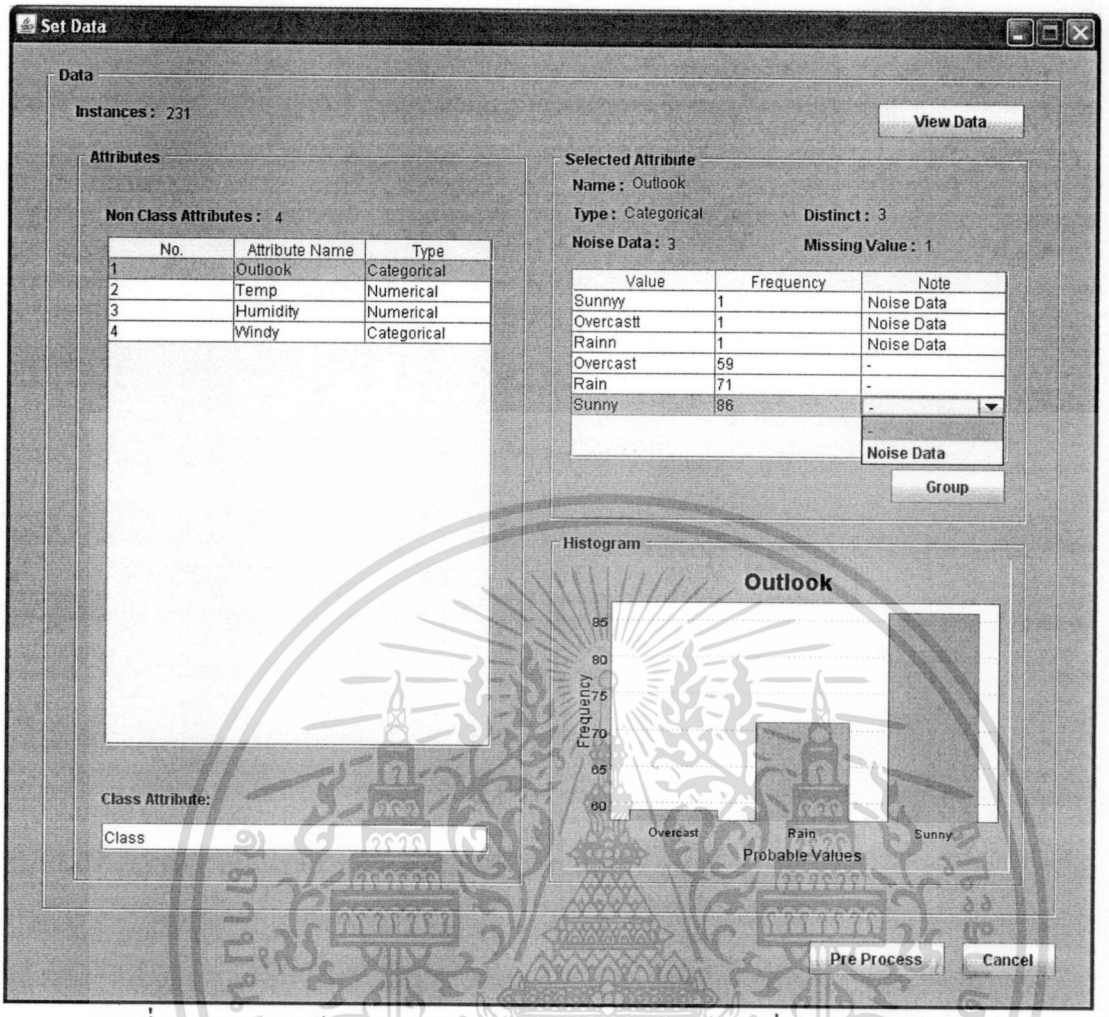

**ภาพที่ 4.10** แสดงหน้าจอการนำข้อมูลเข้าสู่ระบบซึ่งแสดงแทบ SQL

ขั้นตอนการนำเข้าชุดข้อมูลฝึก ผู้ใช้สามารถทำการเชื่อมต่อกับฐานข้อมูลใหม่ (ฐานข้อมูลอื่น) และดึงข้อมูลในฐานข้อมูลใหม่มารวมกับข้อมูลที่ดึงมาแล้วก่อนหน้าจากอีกฐานข้อมูลได้

เพื่อเพิ่มจำนวนชุดข้อมูลฝึกไปเรื่อยๆตามต้องการ โดยที่จำนวนของ Non Class Attributes ที่ใช้จะต้องมีจำนวนเท่ากัน ในการทำดาต้าไมนิ่งยิ่งชุดข้อมูลฝึกมีจำนวนมาก จะยิ่งส่งผลต่อความแม่นยำที่สูงขึ้นในการทำนายผลข้อมูลที่ไม่เคยเห็นมาก่อน ดังนั้นในกรณีที่ข้อมูลของผู้ใช้ถูกจัดเก็บไว้มากกว่า 1 ฐานข้อมูล ผู้ใช้ก็จะสามารถนำข้อมูลของแต่ละฐานข้อมูลมาทำงานร่วมกันได้อย่างสะดวกมากขึ้น และโมเดลต้นไม้ที่ได้ก็จะมีประสิทธิภาพมากยิ่งขึ้นในแง่ของความแม่นยำด้วย โดยวิธีการคือ กดปุ่ม New Connect เพื่อเริ่มทำการเชื่อมต่อกับอีกฐานข้อมูล และดึงข้อมูลเข้าสู่ระบบตามลำดับ โดยหลังจากกดปุ่ม New Connect แล้ว ระบบจะทำการเคลียร์หน้าจอ จากนั้นก็ดำเนินการในแบบเดียวกันกับการนำข้อมูลชุดแรกเข้าสู่ระบบ แต่จำนวน attribute ของข้อมูลชุดใหม่ที่จะโหลดเข้ามาจะต้องมีจำนวนเท่ากัน เมื่อผู้ใช้กดปุ่ม Load Data จากหน้าจอ Connect DB (Training Data) ระบบจะแสดงหน้าจอ Matching Attributes โดยจะแสดงชื่อ attribute ทั้งหมดของชุดข้อมูลแรกไว้ทางซ้ายมือ และattribute ของชุดข้อมูลใหม่ไว้ทางขวามือของตาราง โดยระบบจะทำการ match attribute ระหว่างชุดข้อมูลแรก กับชุดข้อมูลใหม่ โดยพิจารณาจากจำนวนของค่าที่เป็นไปได้ในแต่ละ attribute ของชุดข้อมูลใหม่ ที่ตรงกับค่าที่เป็นไปได้ใน attribute ของชุดข้อมูลแรกมากที่สุด แต่ถ้าระบบทำการ match attribute ผิด ผู้ใช้ก็สามารถทำการแก้ไขให้ถูกต้องได้ โดยเลือกจากรายการ drop down list ดังที่แสดงในภาพที่ 4.11 สุดท้ายข้อมูลจะถูกโหลดเข้าไปรวมกับข้อมูลที่มีอยู่แล้วก่อนหน้า ซึ่งเกิดจากการนำเข้าจากฐานข้อมูลแรก



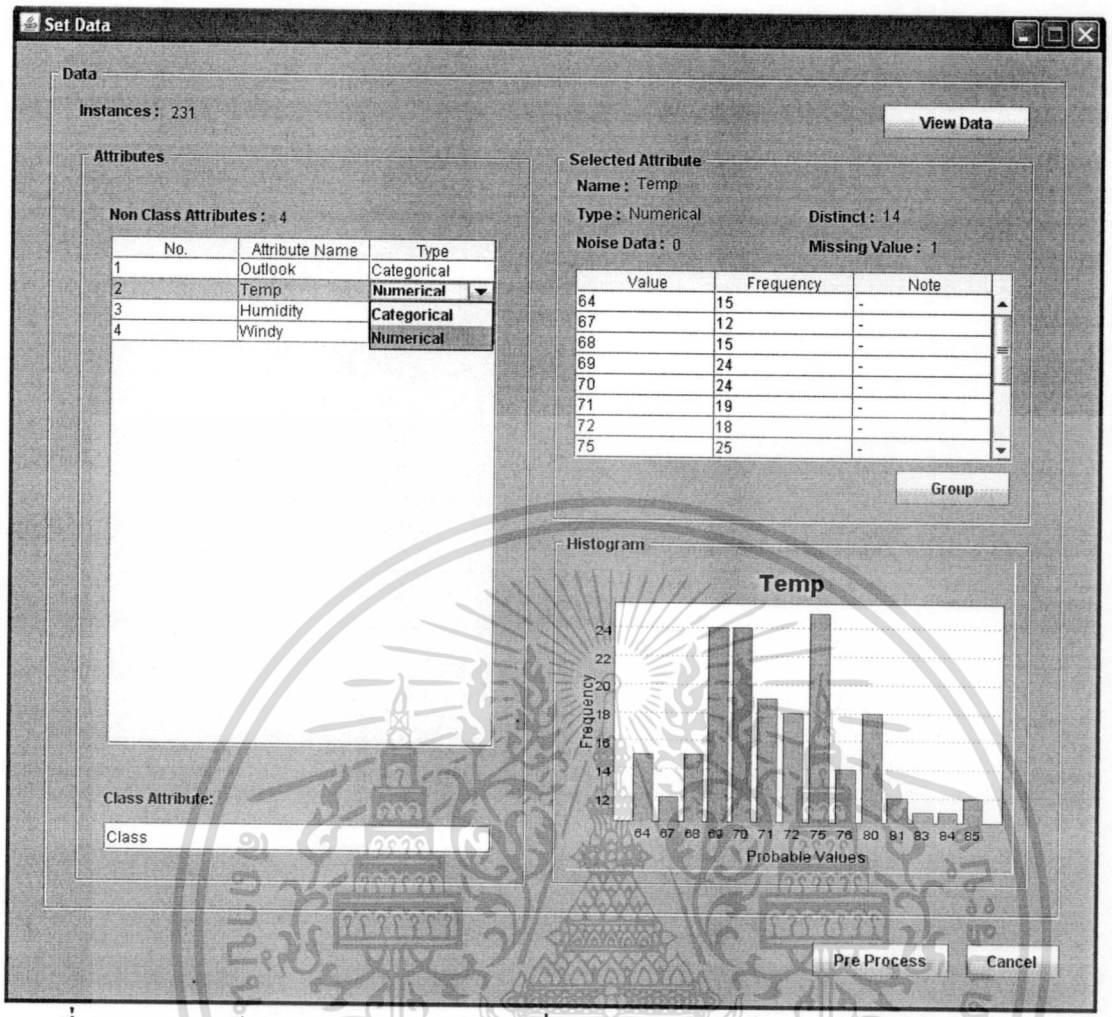

**ภาพที่ 4.11** แสดงหน้าจอ Matching Attribute ของ Training Data




**ภาพที่ 4.12** แสดงหน้าจอ Set Data แสดงการแก้ไขสถานะค่าที่เป็นไปได้ของข้อมูล

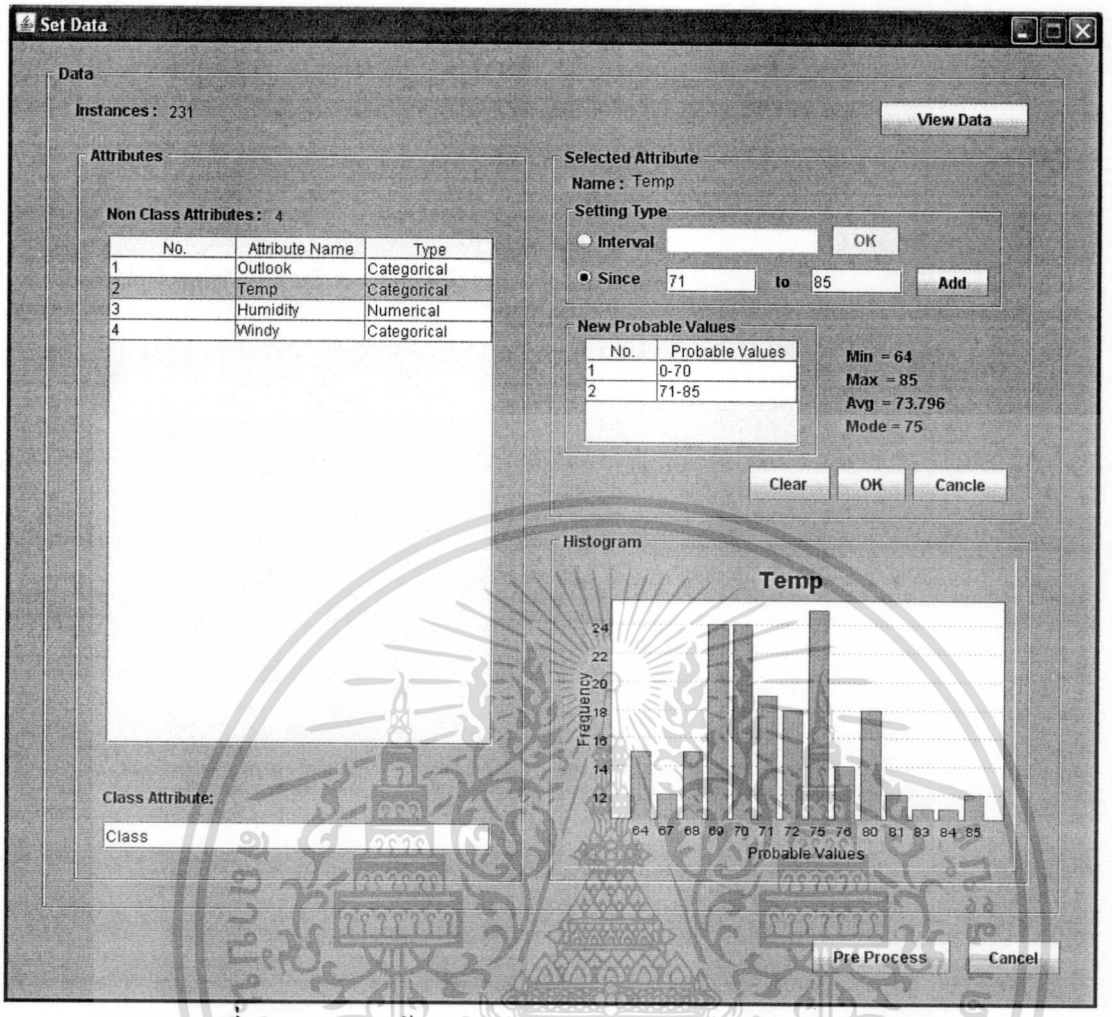
จากภาพที่ 4.12 แสดงหน้าจอ Set Data แสดงการแก้ไขสถานะค่าที่เป็นไปได้ของข้อมูล สามารถอธิบายได้ว่า หลังจากที่ผู้ใช้ดึงข้อมูลจากฐานข้อมูลมาแล้ว หน้าจอต่อไปจะเป็นส่วนของการเซตค่าข้อมูล (Set Data) ในส่วนของหน้าจอนี้เป็นการแสดงค่าข้อมูลของ Training Data ที่ถูกนำเข้ามาในระบบ โดยระบบจะทำการแสดงผลข้อมูลแต่ละ Attribute ว่า ชื่ออะไร, มีชนิดข้อมูลเป็นอะไร (Categorical หรือ Numerical), มีค่าที่เป็นไปได้ของข้อมูลอะไรบ้าง และมีความถี่ของข้อมูลจำนวนเท่าไหร่, มีจำนวน Noise Data เท่าไหร่, จำนวน Missing Value เท่าไหร่ นอกจากนั้นในส่วนของหน้าจอนี้ยังมีส่วนของการจัดการกับการเปลี่ยนชนิดข้อมูลจาก numerical เป็น categorical ซึ่งจะอธิบายต่อไป

จากภาพที่ 4.12 ระบบจะดูที่ค่าความถี่ของค่าข้อมูลแต่ละค่า ถ้าค่าข้อมูลใดมีความถี่ต่ำกว่า 2% ของจำนวน Training Data ทั้งหมด ระบบจะมองค่าข้อมูลนั้นเป็น Noise Data แต่ในบางครั้งข้อมูลที่ระบบมองว่าเป็น Noise Data อาจจะเป็นข้อมูลปกติที่ใช้ หรือบางค่าข้อมูลที่ระบบมองว่าเป็นข้อมูลปกติ อาจจะเป็น Noise Data ก็เป็นได้ ในส่วนของหน้าจอนี้จะเป็นส่วนที่ให้ผู้ใช้สามารถแก้ไขปรับเปลี่ยนค่าข้อมูลต่างๆได้ว่า เป็น Noise Data หรือ ข้อมูลปกติ




**ภาพที่ 4.13** แสดงหน้าจอ Set Data แสดงการเปลี่ยนชนิดข้อมูลจาก Numerical เป็น Categorical

จากภาพที่ 4.13 แสดงหน้าจอ Set Data แสดงการเปลี่ยนชนิดข้อมูลจาก Numerical เป็น Categorical สามารถอธิบายได้ว่า ผู้ใช้สามารถเปลี่ยนชนิดข้อมูลของแต่ละ Attribute ได้ ในกรณีที่เปลี่ยนชนิดข้อมูลจาก Numerical เป็น Categorical โดยระบบจะให้ผู้ใช้ทำการกำหนดค่าที่เป็นไปได้ของข้อมูลใหม่ โดยกำหนดเป็นช่วงข้อมูล ซึ่งข้อมูลในแต่ละช่วง ระบบจะมองเป็นค่าข้อมูลที่เป็นไปได้ 1 ค่า ดังที่แสดงในภาพที่ 4.14 และหากผู้ใช้ต้องการรวมกลุ่มของค่าข้อมูลที่เป็นไปได้ก็สามารถทำได้ โดยการคลิกเลือกค่าที่เป็นไปได้ของข้อมูลที่ต้องการรวมกลุ่ม แล้วกดปุ่ม Group จากนั้นระบบจะแสดงหน้าจอให้ผู้ใช้พิมพ์ค่าข้อมูลใหม่ที่ต้องการเพื่อรวมกลุ่มข้อมูล

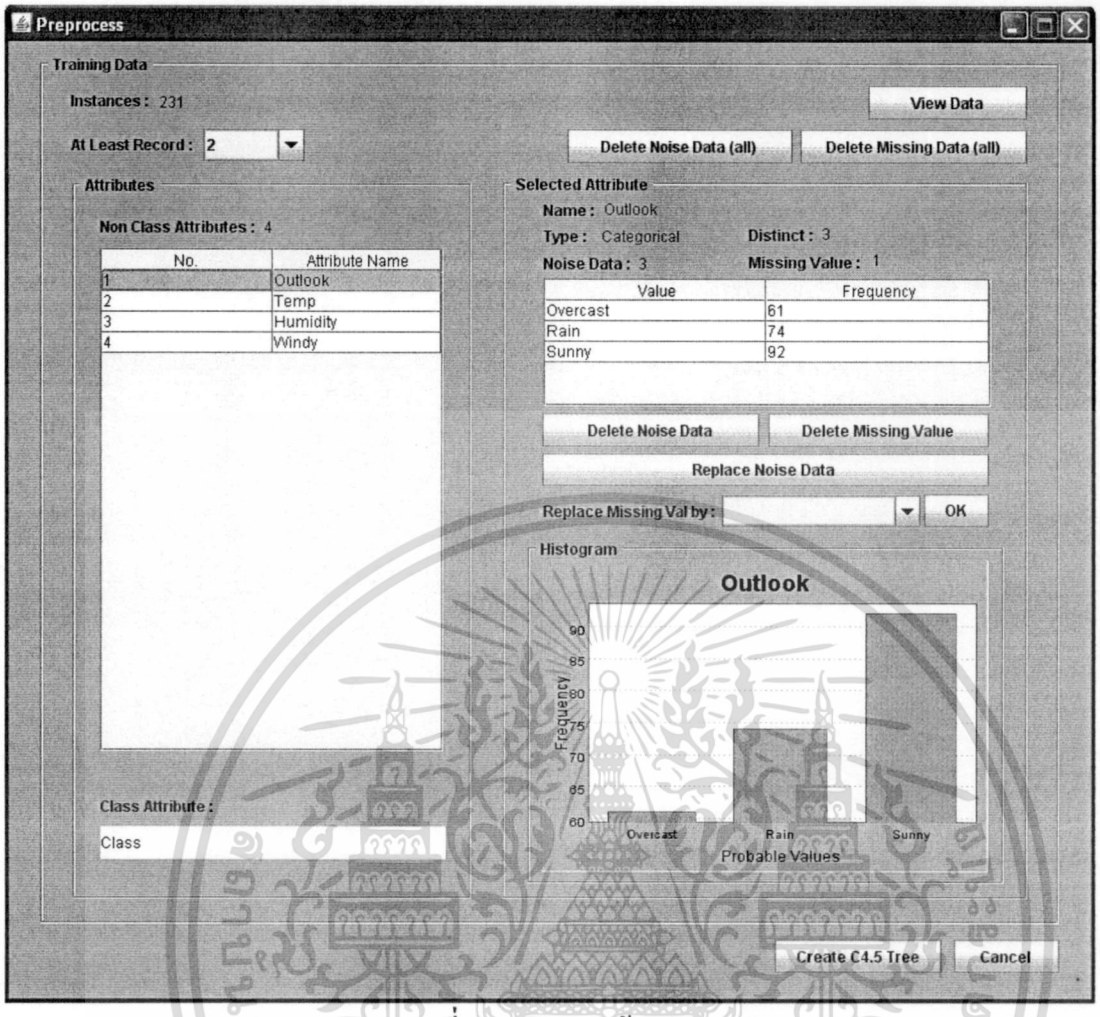



**ภาพที่ 4.14** แสดงหน้าจอ Set Data แสดงการกำหนดช่วงข้อมูลใหม่

จากภาพที่ 4.14 แสดงหน้าจอ Set Data แสดงการกำหนดช่วงข้อมูลใหม่ สามารถอธิบายได้ว่า ผู้ใช้สามารถกำหนดช่วงข้อมูลใหม่ได้ 2 วิธีด้วยกัน คือ

- กำหนดจำนวนช่วงที่ต้องการ โดยใส่จำนวนในช่อง Interval เช่น 3 จากนั้นระบบจะทำการแบ่งช่วงข้อมูลออกมาเป็น 3 ช่วงตามที่ผู้ใช้กำหนด
- กำหนดช่วงที่ต้องการเอง โดยพิมพ์ในส่วนของ Since และ to แล้วกดปุ่ม Add เพื่อกำหนดช่วงในแต่ละช่วง

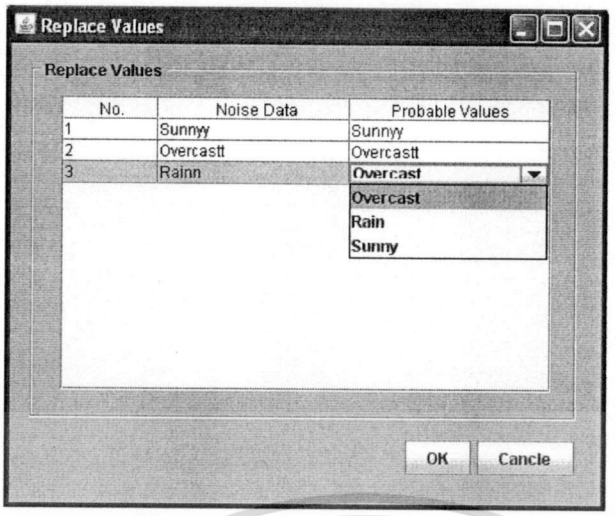
ค่าข้อมูลใหม่ (ช่วงข้อมูล) ที่ถูกกำหนดจะแสดงในตาราง New Probable Values ว่ามีจำนวนเท่าใด และมีค่าข้อมูลช่วงใดบ้าง หลังจากกดปุ่ม OK ระบบจะทำการบันทึกค่าช่วงข้อมูลใหม่ตามที่ผู้ใช้กำหนด




**ภาพที่ 4.15** แสดงหน้าจอ Preprocess

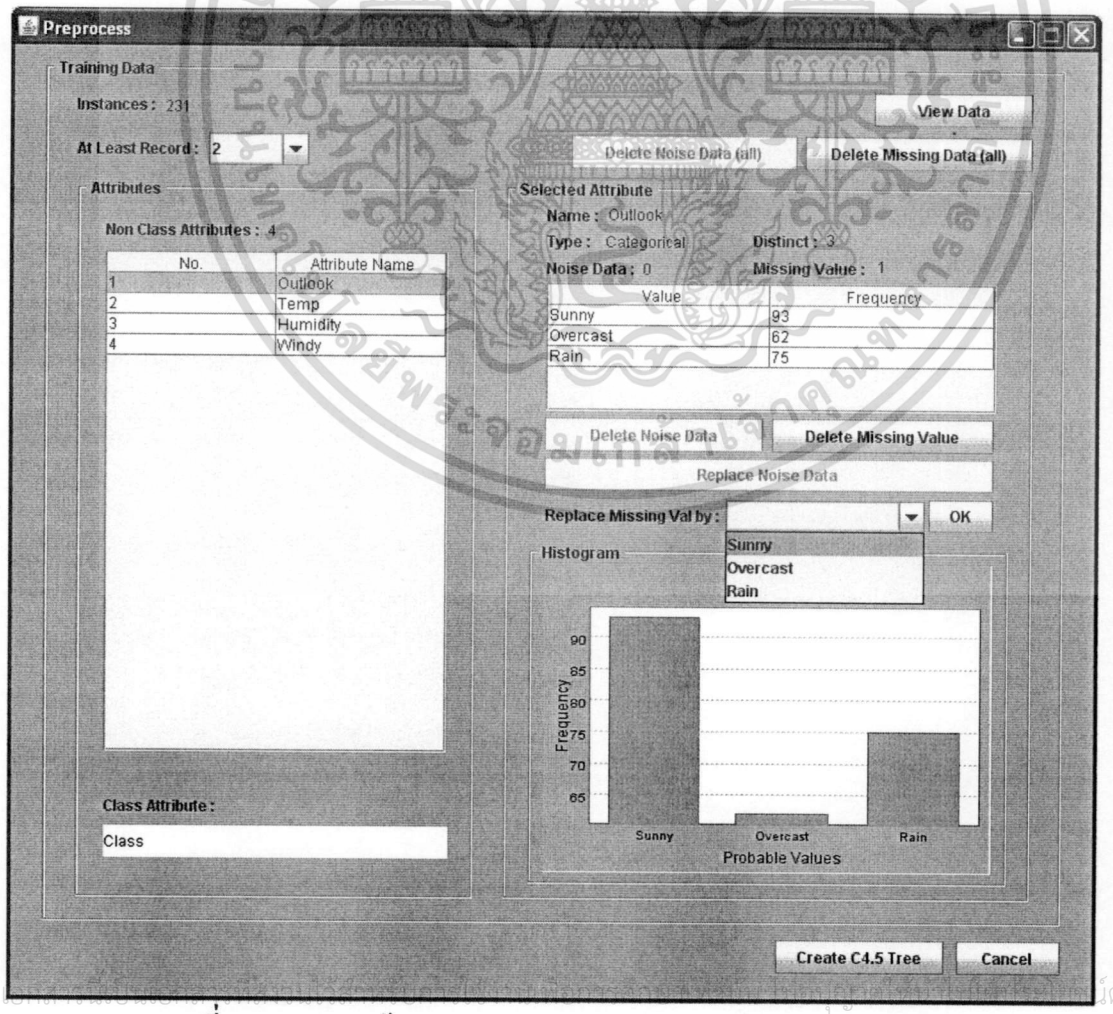
จากภาพที่ 4.15 หน้าจอ Preprocess สามารถอธิบายได้ว่า หน้าจอนี้เป็นส่วนที่ใช้ในการจัดการกับข้อมูลที่เป็น Missing Value และ Noise Data ก่อนจะนำไปสร้าง C4.5 Tree Model ว่าจะ

- ทำการลบทุกแถวข้อมูลที่มีการเก็บค่า Missing Value
- ทำการแทนค่า Missing Value ด้วยค่าที่เป็นไปได้ของข้อมูล (ในกรณีที่ข้อมูลมีชนิดข้อมูลเป็น Numerical หรือ Categorical) หรือค่า Max, Min, Mode หรือ Average (ในกรณีที่ข้อมูลมีชนิดข้อมูลเป็น Numerical)
- ไม่ทำอะไรกับแถวข้อมูลที่เก็บค่า Missing Value (นำไปประมวลผลด้วย)
- ทำการลบทุกแถวข้อมูลที่มีการเก็บค่า Noise Data
- ทำการแทนค่า Noise Data ด้วยค่าที่เป็นไปได้ของข้อมูล (ในกรณีที่ข้อมูลมีชนิดข้อมูลเป็น Numerical หรือ Categorical) หรือค่า Max, Min, Mode หรือ Average (ในกรณีที่ข้อมูลมีชนิดข้อมูลเป็น Numerical) ดังแสดงในภาพที่ 4.16
- ไม่ทำอะไรกับแถวข้อมูลที่เก็บค่า Noise Data (ในกรณีนี้ ระบบจะทำการกำหนดให้ Noise Data ทุกค่า เป็น Missing Value ก่อนจะนำไปสร้างโมเดลต้นไม้ โดยอัตโนมัติ)

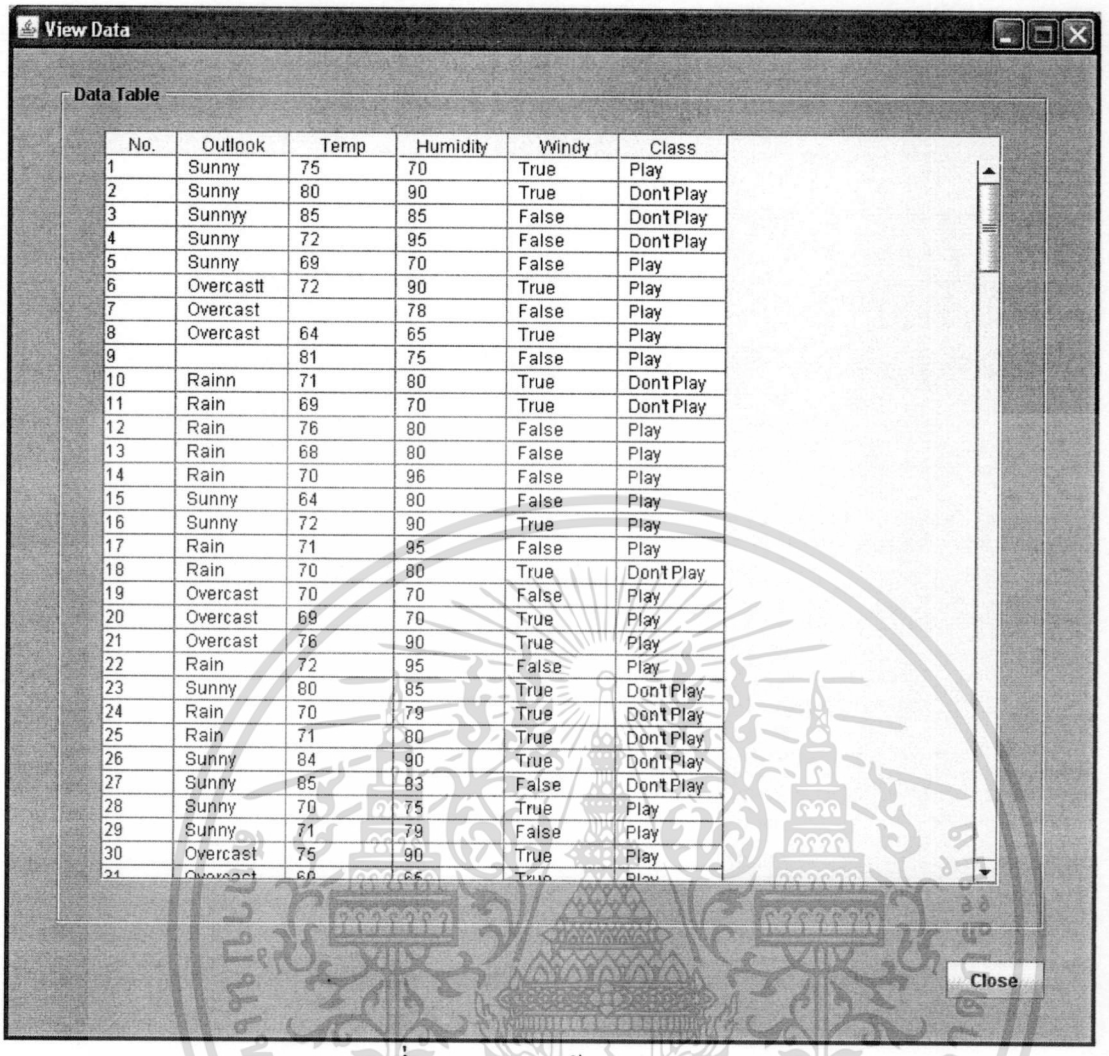



**ภาพที่ 4.16** แสดงหน้าจอ Replace Values แสดงการแทนค่า Noise Data

จากภาพที่ 4.16 สามารถอธิบายได้ว่า หลังจากกดปุ่ม Replace Noise Data จากหน้าจอ Preprocess ระบบจะทำการแสดงหน้าจอ Replace Values ขึ้นมา เพื่อให้ผู้ใช้ทำการกำหนดค่า Noise Data แต่ละค่า ว่าจะให้มีค่าข้อมูลชนิดใหม่ เป็นค่าข้อมูลใดบ้าง



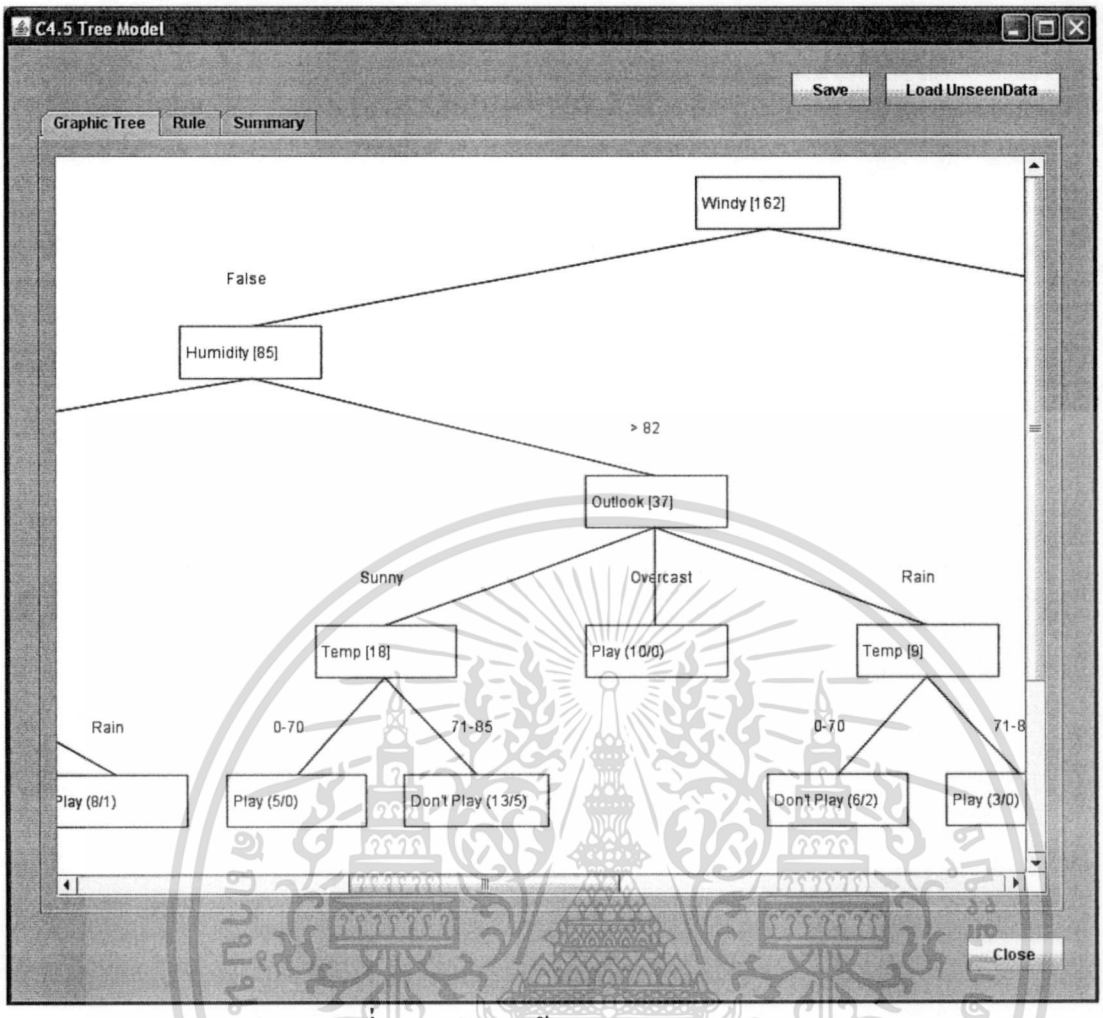

**ภาพที่ 4.17** แสดงหน้าจอ Preprocess แสดงการแทนค่า Missing Values




| No. | Outlook | Temp | Humidity | Windy | Class |
|---|---|---|---|---|---|
| 1 | Sunny | 75 | 70 | True | Play |
| 2 | Sunny | 80 | 90 | True | Don't Play |
| 3 | Sunnyy | 85 | 85 | False | Don't Play |
| 4 | Sunny | 72 | 95 | False | Don't Play |
| 5 | Sunny | 69 | 70 | False | Play |
| 6 | Overcastt | 72 | 90 | True | Play |
| 7 | Overcast | | 78 | False | Play |
| 8 | Overcast | 64 | 65 | True | Play |
| 9 | | 81 | 75 | False | Play |
| 10 | Rainn | 71 | 80 | True | Don't Play |
| 11 | Rain | 69 | 70 | True | Don't Play |
| 12 | Rain | 76 | 80 | False | Play |
| 13 | Rain | 68 | 80 | False | Play |
| 14 | Rain | 70 | 96 | False | Play |
| 15 | Sunny | 64 | 80 | False | Play |
| 16 | Sunny | 72 | 90 | True | Play |
| 17 | Rain | 71 | 95 | False | Play |
| 18 | Rain | 70 | 80 | True | Don't Play |
| 19 | Overcast | 70 | 70 | False | Play |
| 20 | Overcast | 69 | 70 | True | Play |
| 21 | Overcast | 76 | 90 | True | Play |
| 22 | Rain | 72 | 95 | False | Play |
| 23 | Sunny | 80 | 85 | True | Don't Play |
| 24 | Rain | 70 | 79 | True | Don't Play |
| 25 | Rain | 71 | 80 | True | Don't Play |
| 26 | Sunny | 84 | 90 | True | Don't Play |
| 27 | Sunny | 85 | 83 | False | Don't Play |
| 28 | Sunny | 70 | 75 | True | Play |
| 29 | Sunny | 71 | 79 | False | Play |
| 30 | Overcast | 75 | 90 | True | Play |

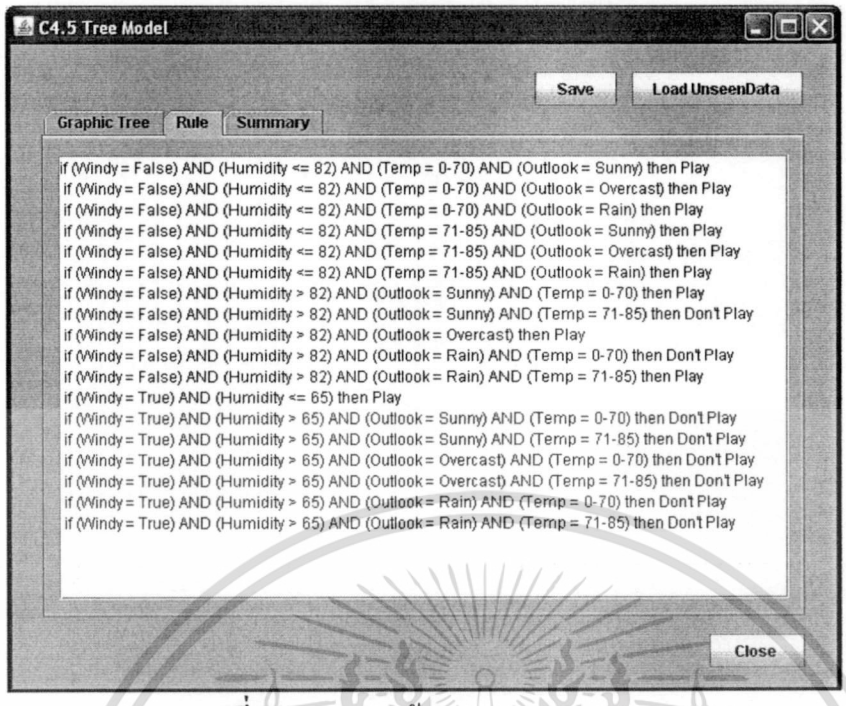
**ภาพที่ 4.18** แสดงหน้าจอ View Data

จากภาพที่ 4.18 แสดงหน้าจอ View Data สามารถอธิบายได้ว่า หน้าจอนี้เป็นหน้าจอที่ใช้สำหรับแสดงข้อมูลที่ถูกนำเข้ามาในระบบในลักษณะของตารางข้อมูล ไม่ว่าจะเป็น Training Data หรือ Unseen Data ก็ตาม โดยถ้าข้อมูลที่ถูกนำเข้าเป็น Training Data ระบบจะแสดงทั้งข้อมูลที่เป็น Non Class Data และ Class Data โดยที่ Class Data จะถูกแสดงในคอลัมน์ทางขวาสุดของตาราง และคอลัมน์ทางซ้ายก็จะเป็นข้อมูลที่เป็น Non Class Data แต่ถ้าข้อมูลที่ถูกนำเข้าเป็น Unseen Data ระบบจะทำการแสดงผลเฉพาะข้อมูลที่เป็น Non Class Data เท่านั้น



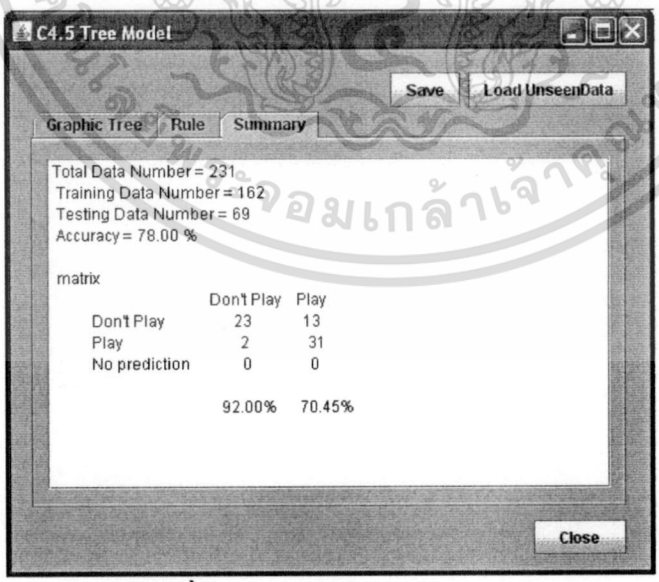

**ภาพที่ 4.19** แสดงหน้าจอ C4.5 Tree Model

ภาพที่ 4.19 แสดงหน้าจอ C4.5 Tree Mode สามารถอธิบายได้ว่า หน้าจอนี้เป็นส่วนของการแสดงผลโมเดลต้นไม้ที่ได้จากการทำดาต้าไมนิ่งในลักษณะของรูปภาพ จากภาพ node ที่อยู่บนสุด คือ root ของต้นไม้ โดยแต่ละ node ที่ไม่ใช่ใบจะมีการเก็บรายละเอียดเกี่ยวกับ node คือ ชื่อ attribute ที่ใช้ในการแตกกิ่ง (attribute ที่มีค่า Gain Ratio สูงสุดใน node นั้นๆ), จำนวนข้อมูลหรือจำนวน record ทั้งหมดที่อยู่ใน node นั้นๆ ซึ่งแสดงไว้ในวงเล็บ ส่วนกิ่งแต่ละที่แตกออกมาคือ ค่าที่เป็นไปได้แต่ละค่าของ attribute หรือ node นั้นๆ และสุดท้าย ในส่วนของ Leaf Node จะบอกผลลัพธ์การทำนายของโมเดลต้นไม้, จำนวนข้อมูลทั้งหมดที่อยู่ใน Leaf Node และจำนวน Error (N/E)




**ภาพที่ 4.20** แสดงหน้าจอการแสดงผล Rule

จากภาพที่ 4.20 แสดงหน้าจอการแสดงผล Rule สามารถอธิบายได้ว่า หน้าจอนี้เป็นหน้าจอที่ใช้สำหรับแสดงกฎของแบบจำลองต้นไม้ที่ได้จากการทำเหมืองข้อมูล โดยระบบจะทำการแสดงผลของกฎในทุกๆเส้นทางที่สามารถเดินทางจาก root node ไปจนถึง node ที่เป็นใบทุกๆ node ใบ ของโมเดลต้นไม้

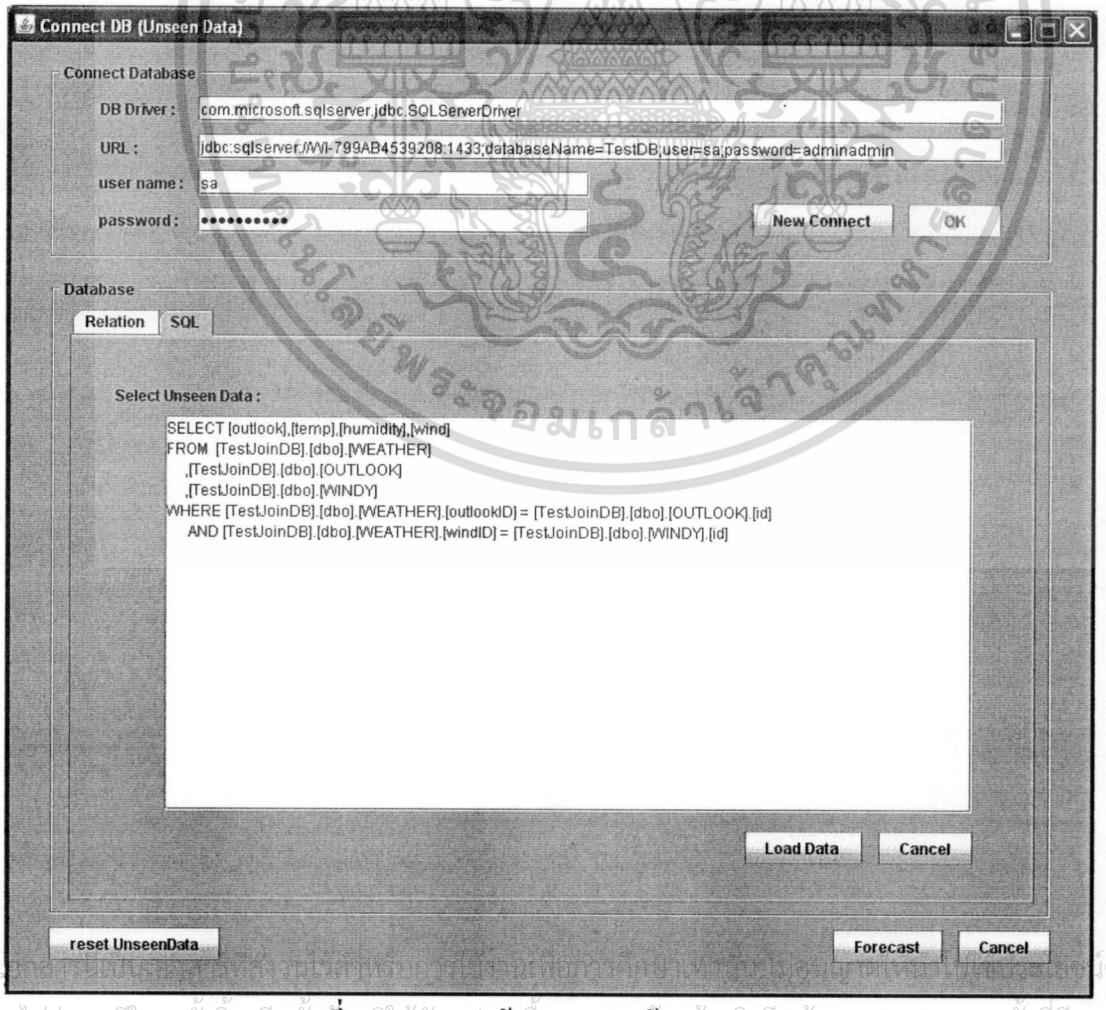



**ภาพที่ 4.21** แสดงหน้าจอการสรุปผล

จากภาพที่ 4.21 แสดงหน้าจอการสรุปผล สามารถอธิบายได้ว่า หน้าจอนี้เป็นการสรุปผลต่างๆ ได้แก่

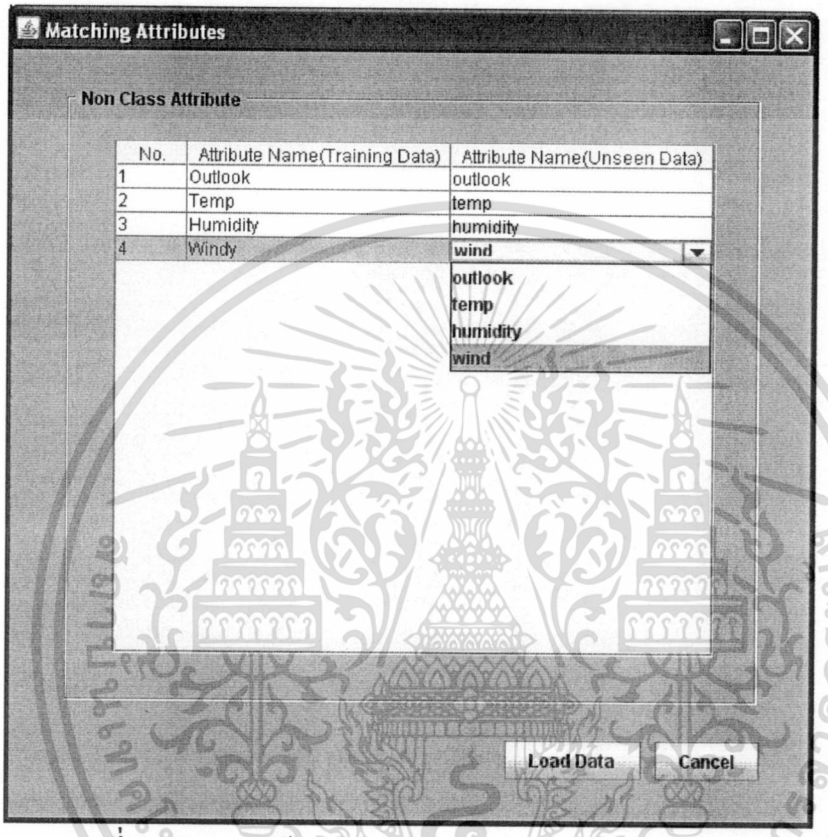
- จำนวนข้อมูลทั้งหมดที่ใช้
- จำนวนข้อมูลที่ใช้เป็น Training Data
- จำนวนข้อมูลที่ใช้เป็น Testing Data
- ความแม่นยำในการทำนายผลข้อมูล คิดเป็นเปอร์เซ็นต์
- Matrix โดยค่าข้อมูลที่เป็นไปได้ของข้อมูล classในแนวนอน คือ ค่าของ Testing Data และค่าข้อมูลที่เป็นไปได้ของข้อมูล class ในแนวตั้งเป็นค่าที่ระบบทำนายผลลัพธ์ของ Testing Data ส่วน No prediction เป็นผลลัพธ์ที่ระบบส่งกลับในกรณีที่การทำนายผลข้อมูลไม่สามารถท่องไปถึง node ใบได้ เช่น ในกรณีที่ข้อมูลเป็นข้อมูลที่ไม่ทราบค่า เป็นต้น จากภาพที่ 4.21 สามารถอ่านได้ว่า class ของ Testing Data มีผลลัพธ์เป็น Don't Play 25 ข้อมูล แต่ระบบทำนายผลเป็น Don't Play 23 ข้อมูล Play 2 ข้อมูล และ class ของ Testing Data มีผลลัพธ์เป็น Play 44 ข้อมูล แต่ระบบทำนายผลเป็น Don't Play 13 ข้อมูล Play 31 ข้อมูล

และเมื่อผู้ใช้กดปุ่ม Save ระบบจะทำการบันทึกโมเดลต้นไม้ที่ได้ในรูปของเอกสาร PMML



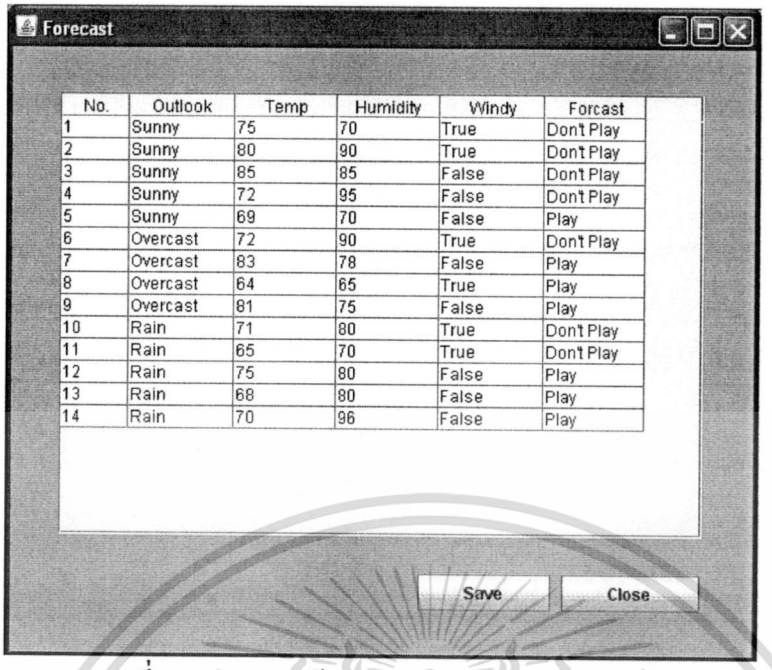

**ภาพที่ 4.22** แสดงหน้าจอการนำเข้า Unseen Data

จากภาพที่ 4.22 แสดงหน้าจอการนำเข้า Unseen Data สามารถอธิบายได้ว่า หน้าจอนี้ เป็นหน้าจอการนำ Unseen Data เข้าสู่ระบบ เพื่อให้ระบบทำการทำนายผลข้อมูล จากโมเดลต้นไม้ที่สร้างขึ้นมาก่อนหน้านี้ โดยมีลักษณะการทำงานคล้ายกับ หน้าจอการนำเข้า Training Data ซึ่งได้อธิบายไปแล้วก่อนหน้านี้ (ภาพที่ 4.9 และภาพที่ 4.10)




**ภาพที่ 4.23** แสดงหน้าจอ Matching Attribute ของ Unseen Data

จากภาพที่ 4.23 แสดงหน้าจอ Matching Attribute ของ Unseen Data สามารถอธิบายได้ว่า เมื่อผู้ใช้กดปุ่ม Load Data ที่หน้าจอการนำเข้า Unseen Data ระบบจะทำแสดงหน้าจอ Matching Attribute ขึ้นมา เพื่อทำการ match attribute ระหว่าง attribute ของ Training Data และ Unseen Data โดย attribute ของ Training Data จะอยู่ทางซ้ายมือ และทางขวามือจะเป็น attribute ของ Unseen Data โดยวิธีการ match จะใช้วิธีเดียวกันกับการนำเข้า Training Data จากฐานข้อมูลมากกว่า 1 ฐานข้อมูล เมื่อกดปุ่ม Load Data ระบบจะทำการโหลด Unseen Data เข้าสู่ระบบ และกลับสู่หน้าจอการนำเข้า Unseen Data เมื่อผู้ใช้กดปุ่ม Forecast ที่หน้าจอการนำเข้า Unseen Data ระบบจะทำการทำนายผล และแสดงผลลัพธ์ที่ได้ต่อไป




**ภาพที่ 4.24** แสดงหน้าจอการทำนายผล Unseen Data

จากภาพที่ 4.24 แสดงหน้าจอการทำนายผล Unseen Data สามารถอธิบายได้ว่า เป็นหน้าจอที่ใช้ในการแสดงผลการทำนายข้อมูลที่ไม่เคยเห็นมาก่อน โดย Attribute หรือคอลัมน์ทางขวามือสุดของตาราง คือ ผลลัพธ์ในการทำนายของโมเดลต้นไม้ และเมื่อกดปุ่ม Save ระบบจะทำการบันทึกข้อมูล Unseen Data และผลการทำนายที่ได้ ไว้ในรูปของ file เอกสาร

```
Attribute
Outlook, Temp, Humidity, Windy, Forecast

Data
Sunny, 75, 70, True, Don't Play
Sunny, 80, 90, True, Don't Play
Sunny, 85, 85, False, Don't Play
Sunny, 72, 95, False, Don't Play
Sunny, 69, 70, False, Play
Overcast, 72, 90, True, Don't Play
Overcast, 83, 78, False, Play
Overcast, 64, 65, True, Play
Overcast, 81, 75, False, Play
Rain, 71, 80, True, Don't Play
Rain, 65, 70, True, Don't Play
Rain, 75, 80, False, Play
Rain, 68, 80, False, Play
Rain, 70, 96, False, Play
```

**ภาพที่ 4.25** แสดง file เอกสารของ Unseen Data และผลลัพธ์การทำนาย

จากภาพที่ 4.25 แสดง file เอกสารของ Unseen Data และผลลัพธ์การทำนาย สามารถอธิบายได้ว่า file เอกสารจะทำการจัดเก็บ attribute และข้อมูลที่เป็น Unseen Data และผลลัพธ์การทำนายของโมเดลต้นไม้ โดยผลลัพธ์ในการทำนายจะอยู่ที่คอลัมน์ทางขวามือสุด

```
<?xml version="1.0" encoding="UTF-8" ?>
- <PMML xmlns="http://www.dmg.org/PMML-4_0"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="4.0">
   <Header copyright="www.dmg.org" />
  - <DataDictionary numberOfFields="4">
    - <DataField name="Outlook" optype="categorical" dataType="string">
        <Value value="Sunny" />
        <Value value="Overcast" />
        <Value value="Rain" />
      </DataField>
    - <DataField name="Temp" optype="categorical" dataType="string">
        <Value value="0-70" />
        <Value value="71-85" />
      </DataField>
      <DataField name="Humidity" optype="continuous" dataType="double" />
    - <DataField name="Windy" optype="categorical" dataType="string">
        <Value value="False" />
        <Value value="True" />
      </DataField>
    - <DataField name="Class" optype="categorical" dataType="string">
        <Value value="Don't Play" />
        <Value value="Play" />
      </DataField>
   </DataDictionary>
 - <TreeModel modelName="test" functionName="classification" missingValueStrategy="nullPrediction">
    - <MiningSchema>
        <MiningField name="Outlook" />
        <MiningField name="Temp" />
        <MiningField name="Humidity" />
        <MiningField name="Windy" />
        <MiningField name="Class" usageType="predicted" />
      </MiningSchema>
    - <Node id="1" score="Play" recordCount="162">
        <True />
        <ScoreDistribution value="Don't Play" recordCount="71" confidence=".44" />
        <ScoreDistribution value="Play" recordCount="91" confidence=".56" />
      - <Node id="2" score="Play" recordCount="85.0">
          <SimplePredicate field="Windy" operator="equal" value="False" />
          <ScoreDistribution value="Don't Play" recordCount="19" confidence=".22" />
          <ScoreDistribution value="Play" recordCount="66" confidence=".78" />
        - <Node id="3" score="Play" recordCount="48.0">
            <SimplePredicate field="Humidity" operator="lessOrEqual" value="82" />
            <ScoreDistribution value="Don't Play" recordCount="7" confidence=".15" />
            <ScoreDistribution value="Play" recordCount="41" confidence=".85" />
          - <Node id="4" score="Play" recordCount="20.0">
              <SimplePredicate field="Temp" operator="equal" value="0-70" />
              <ScoreDistribution value="Don't Play" recordCount="1" confidence=".05" />
              <ScoreDistribution value="Play" recordCount="19" confidence=".95" />
            - <Node id="5" score="Play" recordCount="9.0">
                <SimplePredicate field="Outlook" operator="equal" value="Sunny" />
                <ScoreDistribution value="Don't Play" recordCount="1" confidence=".11" />
                <ScoreDistribution value="Play" recordCount="8" confidence=".89" />
              </Node>
            - <Node id="6" score="Play" recordCount="4.0">
                <SimplePredicate field="Outlook" operator="equal" value="Overcast" />
                <ScoreDistribution value="Play" recordCount="4" confidence="1.00" />
              </Node>
            - <Node id="7" score="Play" recordCount="7.0">
                <SimplePredicate field="Outlook" operator="equal" value="Rain" />
                <ScoreDistribution value="Play" recordCount="7" confidence="1.00" />
              </Node>
            </Node>
          - <Node id="8" score="Play" recordCount="28.0">
              <SimplePredicate field="Temp" operator="equal" value="71-85" />
              <ScoreDistribution value="Don't Play" recordCount="6" confidence=".21" />
              <ScoreDistribution value="Play" recordCount="22" confidence=".79" />
            - <Node id="9" score="Play" recordCount="9.0">
                <SimplePredicate field="Outlook" operator="equal" value="Sunny" />
                <ScoreDistribution value="Don't Play" recordCount="3" confidence=".33" />
                <ScoreDistribution value="Play" recordCount="6" confidence=".67" />
              </Node>
            - <Node id="10" score="Play" recordCount="11.0">
                <SimplePredicate field="Outlook" operator="equal" value="Overcast" />
                <ScoreDistribution value="Don't Play" recordCount="2" confidence=".18" />
                <ScoreDistribution value="Play" recordCount="9" confidence=".82" />
              </Node>
```



**ภาพที่ 4.26** แสดงเอกสาร PMML ของโมเดลต้นไม้

```
        - <Node id="11" score="Play" recordCount="8.0">
            <SimplePredicate field="Outlook" operator="equal" value="Rain" />
            <ScoreDistribution value="Don't Play" recordCount="1" confidence=".12" />
            <ScoreDistribution value="Play" recordCount="7" confidence=".88" />
          </Node>
        </Node>
      </Node>
    - <Node id="12" score="Play" recordCount="37.0">
        <SimplePredicate field="Humidity" operator="greaterThan" value="82" />
        <ScoreDistribution value="Don't Play" recordCount="12" confidence=".32" />
        <ScoreDistribution value="Play" recordCount="25" confidence=".68" />
      - <Node id="13" score="Play" recordCount="18.0">
          <SimplePredicate field="Outlook" operator="equal" value="Sunny" />
          <ScoreDistribution value="Don't Play" recordCount="8" confidence=".44" />
          <ScoreDistribution value="Play" recordCount="10" confidence=".56" />
        - <Node id="14" score="Play" recordCount="5.0">
            <SimplePredicate field="Temp" operator="equal" value="0-70" />
            <ScoreDistribution value="Play" recordCount="5" confidence="1.00" />
          </Node>
        - <Node id="15" score="Don't Play" recordCount="13.0">
            <SimplePredicate field="Temp" operator="equal" value="71-85" />
            <ScoreDistribution value="Don't Play" recordCount="8" confidence=".62" />
            <ScoreDistribution value="Play" recordCount="5" confidence=".38" />
          </Node>
        </Node>
      - <Node id="16" score="Play" recordCount="10.0">
          <SimplePredicate field="Outlook" operator="equal" value="Overcast" />
          <ScoreDistribution value="Play" recordCount="10" confidence="1.00" />
        </Node>
      - <Node id="17" score="Play" recordCount="9.0">
          <SimplePredicate field="Outlook" operator="equal" value="Rain" />
          <ScoreDistribution value="Don't Play" recordCount="4" confidence=".44" />
          <ScoreDistribution value="Play" recordCount="5" confidence=".56" />
        - <Node id="18" score="Don't Play" recordCount="6.0">
            <SimplePredicate field="Temp" operator="equal" value="0-70" />
            <ScoreDistribution value="Don't Play" recordCount="4" confidence=".67" />
            <ScoreDistribution value="Play" recordCount="2" confidence=".33" />
          </Node>
        - <Node id="19" score="Play" recordCount="3.0">
            <SimplePredicate field="Temp" operator="equal" value="71-85" />
            <ScoreDistribution value="Play" recordCount="3" confidence="1.00" />
          </Node>
        </Node>
      </Node>
    </Node>
  - <Node id="20" score="Don't Play" recordCount="77.0">
      <SimplePredicate field="Windy" operator="equal" value="True" />
      <ScoreDistribution value="Don't Play" recordCount="52" confidence=".68" />
      <ScoreDistribution value="Play" recordCount="25" confidence=".32" />
    - <Node id="21" score="Play" recordCount="5.0">
        <SimplePredicate field="Humidity" operator="lessOrEqual" value="65" />
        <ScoreDistribution value="Don't Play" recordCount="1" confidence=".20" />
        <ScoreDistribution value="Play" recordCount="4" confidence=".80" />
      </Node>
    - <Node id="22" score="Don't Play" recordCount="72.0">
        <SimplePredicate field="Humidity" operator="greaterThan" value="65" />
        <ScoreDistribution value="Don't Play" recordCount="51" confidence=".71" />
        <ScoreDistribution value="Play" recordCount="21" confidence=".29" />
      - <Node id="23" score="Don't Play" recordCount="24.0">
          <SimplePredicate field="Outlook" operator="equal" value="Sunny" />
          <ScoreDistribution value="Don't Play" recordCount="15" confidence=".62" />
          <ScoreDistribution value="Play" recordCount="9" confidence=".38" />
        - <Node id="24" score="Don't Play" recordCount="6.0">
            <SimplePredicate field="Temp" operator="equal" value="0-70" />
            <ScoreDistribution value="Don't Play" recordCount="4" confidence=".67" />
            <ScoreDistribution value="Play" recordCount="2" confidence=".33" />
          </Node>
```

**ภาพที่ 4.27** แสดงเอกสาร PMML ของโมเดลต้นไม้ (ต่อ)

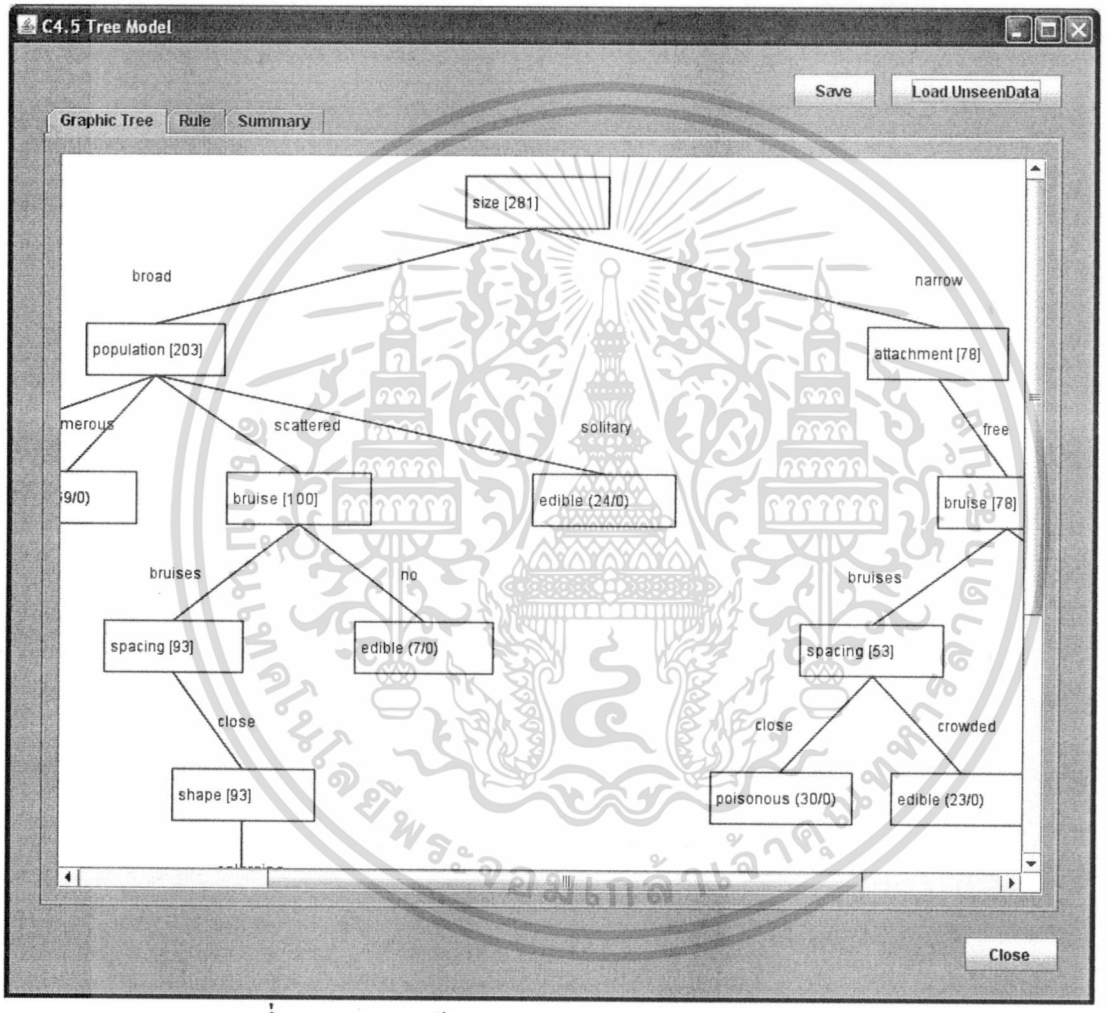
```
        - <Node id="25" score="Don't Play" recordCount="18.0">
            <SimplePredicate field="Temp" operator="equal" value="71-85" />
            <ScoreDistribution value="Don't Play" recordCount="11" confidence=".61" />
            <ScoreDistribution value="Play" recordCount="7" confidence=".39" />
          </Node>
        </Node>
      - <Node id="26" score="Don't Play" recordCount="19.0">
          <SimplePredicate field="Outlook" operator="equal" value="Overcast" />
          <ScoreDistribution value="Don't Play" recordCount="12" confidence=".63" />
          <ScoreDistribution value="Play" recordCount="7" confidence=".37" />
        - <Node id="27" score="Don't Play" recordCount="3.0">
            <SimplePredicate field="Temp" operator="equal" value="0-70" />
            <ScoreDistribution value="Don't Play" recordCount="2" confidence=".67" />
            <ScoreDistribution value="Play" recordCount="1" confidence=".33" />
          </Node>
        - <Node id="28" score="Don't Play" recordCount="16.0">
            <SimplePredicate field="Temp" operator="equal" value="71-85" />
            <ScoreDistribution value="Don't Play" recordCount="10" confidence=".62" />
            <ScoreDistribution value="Play" recordCount="6" confidence=".38" />
          </Node>
        </Node>
      - <Node id="29" score="Don't Play" recordCount="29.0">
          <SimplePredicate field="Outlook" operator="equal" value="Rain" />
          <ScoreDistribution value="Don't Play" recordCount="24" confidence=".83" />
          <ScoreDistribution value="Play" recordCount="5" confidence=".17" />
        - <Node id="30" score="Don't Play" recordCount="15.0">
            <SimplePredicate field="Temp" operator="equal" value="0-70" />
            <ScoreDistribution value="Don't Play" recordCount="13" confidence=".87" />
            <ScoreDistribution value="Play" recordCount="2" confidence=".13" />
          </Node>
        - <Node id="31" score="Don't Play" recordCount="14.0">
            <SimplePredicate field="Temp" operator="equal" value="71-85" />
            <ScoreDistribution value="Don't Play" recordCount="11" confidence=".79" />
            <ScoreDistribution value="Play" recordCount="3" confidence=".21" />
          </Node>
        </Node>
      </Node>
    </Node>
  </Node>
 </TreeModel>
</PMML>
```

**ภาพที่ 4.28** แสดงเอกสาร PMML ของโมเดลต้นไม้ (ต่อ)

จากภาพที่ 4.26-4.28 แสดงเอกสาร PMML ของโมเดลต้นไม้ สามารถอธิบายได้ว่า เอกสาร PMML จะทำการจัดเก็บข้อมูลทุกอย่างที่เกี่ยวข้องกับโมเดลต้นไม้ เช่น ชื่อ attribute, ค่าที่เป็นไปได้ในแต่ละ attribute, จำนวนข้อมูลในแต่ละ node, โครงสร้างของต้นไม้ตั้งแต่ root จนถึงใบ เป็นต้น ซึ่งรายละเอียดต่างๆสามารถดูได้ที่ภาคผนวก ข

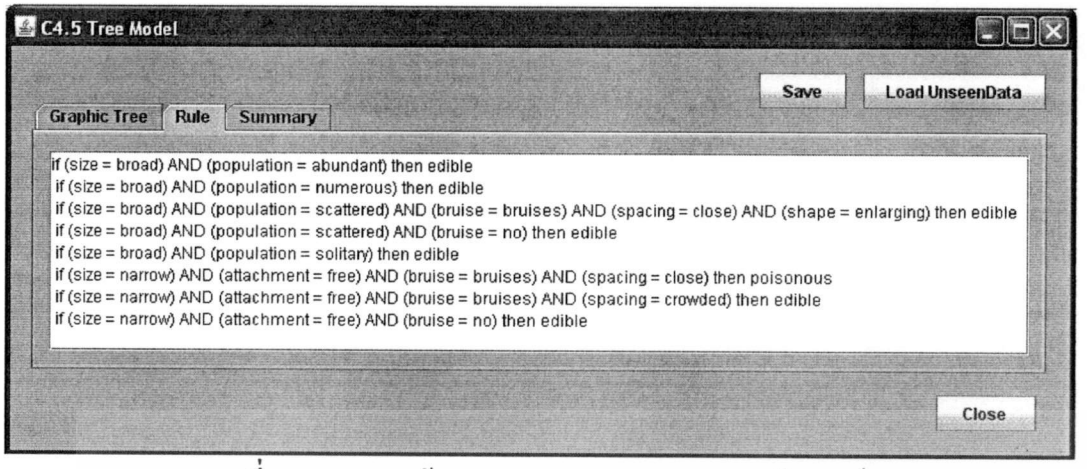
โครงการระบบฉบับนี้ได้ทำการทดลองระบบ โดยใช้ชุดข้อมูลในการทดลอง จำนวน 2 ชุด ด้วยกัน คือ ชุดข้อมูลการเล่นเทนนิส และชุดข้อมูลเห็ด ซึ่งรายละเอียดของข้อมูลที่ใช้สามารถอ่านเพิ่มเติมได้ที่ภาคผนวก ค

จากภาพการอธิบายส่วนติดต่อกับผู้ใช้ที่ผ่านมาข้างต้นนั้น ได้เลือกใช้ชุดข้อมูลการเล่นเทนนิส ในการสาธิตการใช้งานระบบ ส่วนผลลัพธ์ในการใช้งานจากชุดข้อมูลเห็ด จะแสดงในลำดับถัดไปนี้

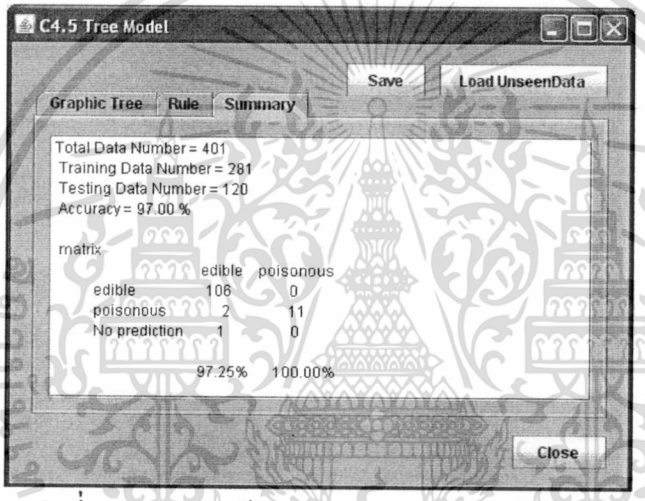



**ภาพที่ 4.29** แสดงหน้าจอ C4.5 Tree Model ของชุดข้อมูลเห็ด

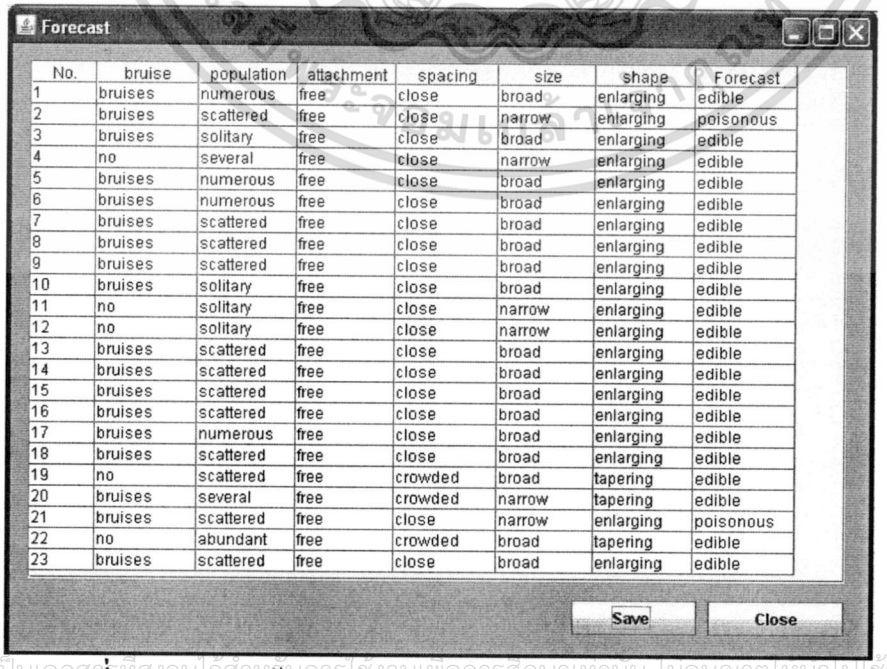
จากภาพที่ 4.29 แสดงหน้าจอ C4.5 Tree Model ของชุดข้อมูลเห็ด สามารถอธิบายได้ว่า attribute ที่เลือกใช้ในการทดสอบมีด้วยกัน 7 attribute ได้แก่ attribute bruise, population, attachment, spacing, size, shape, class จำนวน 401 record แบ่งเป็น Training Data 281 record และ Testing Data 120 record




**ภาพที่ 4.30** แสดงหน้าจอการแสดงผล Rule ของชุดข้อมูลเห็ด




**ภาพที่ 4.31** แสดงหน้าจอการสรุปผลของชุดข้อมูลเห็ด




| No. | bruise | population | attachment | spacing | size | shape | Forecast |
|---|---|---|---|---|---|---|---|
| 1 | bruises | numerous | free | close | broad | enlarging | edible |
| 2 | bruises | scattered | free | close | narrow | enlarging | poisonous |
| 3 | bruises | solitary | free | close | broad | enlarging | edible |
| 4 | no | several | free | close | narrow | enlarging | edible |
| 5 | bruises | numerous | free | close | broad | enlarging | edible |
| 6 | bruises | numerous | free | close | broad | enlarging | edible |
| 7 | bruises | scattered | free | close | broad | enlarging | edible |
| 8 | bruises | scattered | free | close | broad | enlarging | edible |
| 9 | bruises | scattered | free | close | broad | enlarging | edible |
| 10 | bruises | solitary | free | close | broad | enlarging | edible |
| 11 | no | solitary | free | close | narrow | enlarging | edible |
| 12 | no | solitary | free | close | narrow | enlarging | edible |
| 13 | bruises | scattered | free | close | broad | enlarging | edible |
| 14 | bruises | scattered | free | close | broad | enlarging | edible |
| 15 | bruises | scattered | free | close | broad | enlarging | edible |
| 16 | bruises | scattered | free | close | broad | enlarging | edible |
| 17 | bruises | numerous | free | close | broad | enlarging | edible |
| 18 | bruises | scattered | free | close | broad | enlarging | edible |
| 19 | no | scattered | free | crowded | broad | tapering | edible |
| 20 | bruises | several | free | crowded | narrow | tapering | edible |
| 21 | bruises | scattered | free | close | narrow | enlarging | poisonous |
| 22 | no | abundant | free | crowded | broad | tapering | edible |
| 23 | bruises | scattered | free | close | broad | enlarging | edible |

**ภาพที่ 4.32** แสดงหน้าจอการทำนายผล Unseen Data ของชุดข้อมูลเห็ด

# บทที่ 5
# สรุปผลการวิจัย และข้อเสนอแนะ

## 5.1 สรุปผลการวิจัย

Data Mining เป็นกระบวนการที่ใช้เพื่อค้นหาสารสนเทศที่มีประโยชน์ ที่ถูกซ่อนเร้นอยู่ภายในฐานข้อมูล เพื่อนำมาช่วยประกอบการตัดสินใจในการดำเนินงานต่างๆ วิธีการแก้ปัญหาด้วย Data Mining นั้นมีอยู่ด้วยกันหลายรูปแบบขึ้นอยู่กับวัตถุประสงค์ของการทำงาน โครงการพัฒนาระบบงานฉบับนี้ได้เสนอวิธีการสร้างแบบจำลองพยากรณ์ (Predictive Modeling) เพื่อทำนายผลข้อมูลที่ไม่เคยเห็นมาก่อน โดยใช้อัลกอริทึม C4.5 ซึ่งเป็นอัลกอริทึมของ Classification ที่มีการใช้งานกันอย่างกว้างขวาง อันเนื่องมาจากความมีประสิทธิภาพในการแก้ปัญหา และสามารถทำความเข้าใจได้ง่าย

ผลจากการศึกษาทำให้ได้ระบบที่ใช้สำหรับจัดกลุ่มของข้อมูล และสามารถทำนายผลข้อมูลที่ไม่เคยพบเห็นมาก่อนได้ โดยระบบสามารถเชื่อมต่อกับฐานข้อมูลต่างๆได้ โดยไม่ยึดติดกับฐานข้อมูลใดฐานข้อมูลหนึ่ง และสามารถนำข้อมูลจากฐานข้อมูลอื่นที่มีโครงสร้างการจัดเก็บเหมือนกันแต่ถูกจัดเก็บในฐานข้อมูลต่างชนิดกัน มาใช้งานร่วมกันในการนำมาสร้างแบบจำลองต้นไม้ได้ ส่งผลให้แบบจำลองต้นไม้ที่ได้จากการทำเหมืองข้อมูลมีความสามารถในการทำนายผลข้อมูลที่ไม่เคยพบเห็นมาก่อนได้อย่างถูกต้องแม่นยำมากขึ้น เพราะยิ่งข้อมูลที่นำมาใช้ในการสร้างแบบจำลองมีจำนวนมาก ก็จะยิ่งส่งผลดีต่อประสิทธิภาพในการทำนายผลข้อมูลของแบบจำลอง คือผลลัพธ์ในการทำนายมีความครอบคลุมมากขึ้น ซึ่งจะทำให้ทำนายผลได้แม่นยำมากขึ้น และแบบจำลองมีความน่าเชื่อถือนั่นเอง นอกจากนั้นระบบสามารถแสดงผลแบบจำลองต้นไม้ในลักษณะของรูปภาพได้ ทำให้ผู้ใช้ทำการวิเคราะห์โมเดลได้ง่าย รวดเร็ว และสะดวกยิ่งขึ้น เกิดความผิดพลาดน้อยลง สุดท้ายระบบสามารถทำการบันทึกและอ่านแบบจำลองต้นไม้ที่ถูกจัดเก็บในรูปของเอกสาร PMML ซึ่งเป็นภาษามาตรฐาน โดยมีการใช้เทคนิค TreeModel และวิธีการจัดการกับข้อมูลในกรณีที่การประเมินผลของ Node มีค่าเป็น UNKNOWN ด้วยวิธี nullPrediction ได้ ทำให้สามารถนำแบบจำลองต้นไม้ที่ได้ ไปใช้งานร่วมกับระบบอื่นที่มีความสามารถในการรองรับเอกสาร PMML ที่มีการใช้เทคนิคเช่นเดียวกันกับโครงการพัฒนาระบบฉบับนี้ได้อีกด้วย ซึ่งผลดี คือทำให้เกิดความยืดหยุ่นในการแลกเปลี่ยนข้อมูล และการนำไปใช้ให้เกิดประโยชน์เพิ่มมากขึ้น

## 5.2 ข้อเสนอแนะ

โครงการพัฒนาระบบงานฉบับนี้ได้นำเสนอการจัดเก็บแบบจำลองต้นไม้ C4.5 ที่ได้จากการทำดาต้าไมนิ่ง ด้วยเอกสาร PMML โดยแบบจำลอง TreeModel ของ PMML มีวิธีการทำนายผลข้อมูลในกรณีที่การประเมินผลของ Node มีค่าเป็น UNKNOWN ในระหว่างทำนายผลข้อมูลหลายวิธีด้วยกัน ซึ่งแต่ละวิธีจะมีเทคนิคการจัดการที่แตกต่างกัน โครงการพัฒนาระบบฉบับนี้ได้ทำการเลือกวิธีการจัดการในกรณีที่การประเมินผลมีค่าเป็น UNKNOWN เพียงวิธีเดียวเท่านั้น คือ วิธี nullPrediction โดยโมเดลจะส่งการทำนายผลข้อมูลกลับมาเป็น no prediction นั่นเอง หากโครงการพัฒนาระบบฉบับนี้ได้รับการพัฒนาต่อ โดยให้ระบบมีความสามารถ หรือมีวิธีการทำนายผลข้อมูลในกรณีที่การประเมินผลมีค่าเป็น UNKNOWN ได้หลายวิธีมากขึ้น ก็จะทำให้ระบบมีความยืดหยุ่น และมีประสิทธิภาพมากยิ่งขึ้น

# บรรณานุกรม

ก้องศักดิ์ จงเกษมวงศ์. 2543. "การตัดเล็มอย่างอ่อนสำหรับต้นไม้ตัดสินใจโดยการใช้แบ็กพรอพาเกชันนิวรอลเน็ตเวิร์ก." วิทยานิพนธ์วิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์, จุฬาลงกรณ์มหาวิทยาลัย.

กำพล ปัญญาเวชมานิต. 2540. "การประยุกต์ใช้การเรียนรู้ของเครื่องกับการอนุมัติสินเชื่อ." วิทยานิพนธ์วิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์, จุฬาลงกรณ์มหาวิทยาลัย.

นฤพนต์ ว่องประชานุกูล. 2548. "วิธีที่เหมาะสมสำหรับการตัดกิ่งต้นไม้ตัดสินใจของการทำเหมืองข้อมูลทางด้านวิทยาศาสตร์." วิทยานิพนธ์วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์, มหาวิทยาลัยเทคโนโลยีสุรนารี.

นฤมล สมบูรณ์เงิน. 2544. "การพัฒนาระบบงานเพื่ออนุมัติสินเชื่อเบื้องต้นโดยใช้อัลกอริทึม C4.5." โครงการพัฒนาระบบงาน หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.

พวงทิพย์ แท่นแสง. 2550. "การทดสอบประสิทธิภาพการทำงานของอัลกอริทึมการไม่นิ่งกฎสำหรับจำแนก." สารนิพนธ์ วิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ.

ศิริกาญจนา พิลาบุตร. 2551. "การสร้างกฎข้อบังคับของฐานข้อมูลโดยการทำเหมืองข้อมูล." วิทยานิพนธ์วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์, มหาวิทยาลัยเทคโนโลยีสุรนารี.

PMML 4.0 - General Structure of a PMML Document. [Online]. Available: http://www.dmg.org/v4-0-1/GeneralStructure.html

PMML 4.0 – Trees. [Online]. Available: http://www.dmg.org/v4-0-1/TreeModel.html

Quinlan, J. Ross. 1993. **C4.5 Programs for Machine Learning.** California: Morgan Kaufmann.

# ภาคผนวก

# ภาคผนวก ก

# อัลกอริทึมในการวาดต้นไม้

# อัลกอริทึมในการวาดต้นไม้

โครงการพัฒนาระบบฉบับนี้ใช้วิธีการวาดต้นไม้ โดยอาศัยเทคนิคการท่องต้นไม้แบบ Post-Order

โดยก่อนวาด จะมีการกำหนดขนาดของ node ที่ต้องการวาดตามแนวแกน X และแนวแกน Y, กำหนดความสูงของกิ่ง, ระยะห่างระหว่าง node แต่ละ node ตามแนวแกน X และแนวแกน Y ไว้ล่วงหน้า เพื่อนำมาใช้ในการคำนวณหาตำแหน่งของแต่ละ node ที่จะวาด รวมถึงแต่ละกิ่งที่แตกออกด้วย โดย

ตำแหน่งแนวแกน X ของ node ใบ เกิดจากจำนวนของ node ใบที่ถูกวาดไปแล้ว คูณด้วยผลบวกของ ความยาว node กับ ระยะห่างระหว่าง node จากนั้นนำผลที่ได้มาบวกกับ ระยะห่างระหว่าง node

ตำแหน่งแนวแกน X ของ node ที่ไม่ใช่ใบ เกิดจากการคำนวณหาจุดกึ่งกลางระหว่างตำแหน่งของ node ลูกซ้ายสุด กับ ตำแหน่งขวาสุดของ node ลูกขวาสุด (ตำแหน่งของ node ลูกขวาสุด บวกกับ ความยาว node)

ตำแหน่งแนวแกน Y ของ node ที่ถูกวาดจะคำนวณได้จาก ความสูงของ node นั้นใน tree คูณด้วย ผลบวกของความสูงของกิ่งตามแนวแกน Y กับ ความสูงของ node ตามแนวแกน Y

กิ่งแต่ละกิ่ง เกิดจากการลากเส้น จากตำแหน่งกึ่งกลางของ node ถึงตำแหน่งกึ่งกลางของ parent node

อัลกอริทึมในการวาดต้นไม้แสดงได้ดังภาพที่ ก.1 โดยมีขั้นตอนหลักๆดังนี้

1. เริ่มจากวาด node ใบ ทางซ้ายสุดของ tree ก่อน
2. จากนั้นพิจารณาที่ sibling ของ node ใบ ทางซ้ายสุดของ tree ว่าเป็น node ใบ หรือไม่
3. ถ้า sibling เป็น node ใบ ก็จะวาด node ถัดต่อจาก node ใบที่วาดไปแล้ว แต่ถ้า sibling ไม่ใช่ node ใบ ให้ทำตามวิธีคล้ายวิธีเดิม คือ เริ่มวาด node ใบ ทางซ้ายสุด ของ sibling นั้นก่อน (วาดถัดต่อจาก node ใบที่วาดไปแล้ว) แล้วพิจารณาที่ sibling ของ node ใบ ไปเรื่อยๆ จนกว่าจะครบทุก sibling

4. จากนั้น วาด parent node แล้วเริ่มพิจารณาที่ sibling ของ parent node ที่เหลือ ทำตามขั้นตอนเหมือนที่กล่าวมา ไปเรื่อยๆ จนกระทั่งถึง root จึงหยุด สุดท้ายแล้ว จะได้รูปโมเดลต้นไม้ C4.5 Tree

```
        Algorithm
(1)   node = node ทางซ้ายสุดของ root
(2)   y = ความสูงของ node ใบ ทางซ้ายสุดของต้นไม้
(3)   while ( node <> root ) do
(4)     begin
(5)       คำนวณหาตำแหน่งของ parent node
(6)       for ( i = 0 to จำนวน node ลูก - 1 ) do
(7)         begin
(8)           if ( node ลูกที่ i เป็นใบ )
(9)             then วาดรูป node ที่ i พร้อมกับทำนายผล, บอกจำนวนข้อมูลใน node และ error
(10)            else เรียกตัวเอง โดยส่ง node ลูกที่ i ให้กับตัวแปร root
(11)          วาดกิ่ง โดยลากจาก node ลูกที่ i ไปยังตำแหน่งของ parent node
(12)        end
(13)      วาดรูป node ของ parent node พร้อมกับบอก attribute ที่มีค่า gain ratio สูงสุด และ จำนวนข้อมูล
(14)      if ( parent node <> root ) then
(15)        begin
(16)          if (grandparent node <> null) then
(17)            begin
(18)              คำนวณหาตำแหน่งของ grandparent node
(19)              วาดกิ่ง โดยลากจาก parent node ไปยัง grandparent node
(20)            end
(21)        end
(22)      node = parent node
(23)  end while
```

**ภาพที่ ก.1** แสดงอัลกอริทึมในการวาดต้นไม้

# ภาคผนวก ข

## The Predictive Model Markup Language (PMML)

# The Predictive Model Markup Language (PMML)

ในส่วนนี้จะกล่าวถึงเอกสาร PMML ที่ใช้สำหรับจัดเก็บโมเดลที่ได้จากการทำ Data Mining

PMML คือ ภาษามาตรฐาน (de facto standard language) ที่ใช้แสดงแบบจำลองการทำ Data Mining ช่วยให้รวมเหมืองข้อมูลที่แตกต่างกันเข้าเป็นภาษาเดียวกัน ซึ่งแนวทางนี้จะสามารถย้ายแบบจำลองไปยังส่วนอื่นๆได้อย่างง่ายดาย คือ เอกสาร PMML หนึ่งอาจถูกพัฒนาโดยระบบหนึ่ง แต่สามารถนำไปใช้งานในอีกระบบหนึ่งได้

PMML ได้รับการพัฒนาโดยนักพัฒนาการทำเหมืองข้อมูล (Data Mining Group: DMG) เพื่อรองรับการทำเหมืองข้อมูลให้เป็นมาตรฐาน PMML มีหลายรุ่น หรือหลาย version ด้วยกัน โดย version ล่าสุด คือ version 4.0 ซึ่งถูกปล่อยออกมาเมื่อเดือน มิถุนายน 2009

## ข.1 โครงสร้างทั่วไปของเอกสาร PMML 4.0

PMML ใช้ XML ในการแสดงแบบจำลองที่ได้จากการทำ Data Mining โดยโครงสร้างของแบบจำลองจะถูกอธิบายด้วย XML Schema เอกสาร PMML คือ เอกสาร XML ที่มี root element ตามชนิดของ PMML โครงสร้างทั่วไปของเอกสาร PMML ปรากฏดังภาพข้างล่าง

```
<?xml version="1.0"?>
<PMML version="4.0"
  xmlns="http://www.dmg.org/PMML-4_0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" >
  <Header copyright="Example.com"/>
  <DataDictionary> ... </DataDictionary>
  ... a model ...
</PMML>
```

**ภาพที่ ข.1** แสดงโครงสร้างทั่วไปของเอกสาร PMML

Namespace ใน PMML Schema มีนิยามดังที่แสดงข้างล่างนี้

```
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.dmg.org/PMML-4_0"
  xmlns="http://www.dmg.org/PMML-4_0"
  elementFormDefault="unqualified">
```

**ภาพที่ ข.2** แสดงนิยาม Namespace ใน PMML Schema

การสร้างเอกสาร PMML จะมีกฎ หรือการกำหนดรายละเอียดที่ใช้ในการอธิบายแบบจำลองของเทคนิคต่างๆ แตกต่างกัน โครงการพัฒนาระบบฉบับนี้ได้เลือกทำการศึกษาเฉพาะกฎการสร้างเอกสาร PMML ในการอธิบายโครงสร้าง TreeModel เพียงอย่างเดียวเท่านั้น PMML XSD มีนิยามดังที่แสดงข้างล่างนี้

```
<xs:element name="PMML">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Header"/>
      <xs:element ref="MiningBuildTask" minOccurs="0"/>
      <xs:element ref="DataDictionary"/>
      <xs:element ref="TransformationDictionary" minOccurs="0"/>
      <xs:sequence minOccurs="0" maxOccurs="unbounded">
       <xs:choice>
          <xs:element ref="AssociationModel"/>
          <xs:element ref="ClusteringModel"/>
          <xs:element ref="GeneralRegressionModel"/>
          <xs:element ref="MiningModel"/>
          <xs:element ref="NaiveBayesModel"/>
          <xs:element ref="NeuralNetwork"/>
          <xs:element ref="RegressionModel"/>
          <xs:element ref="RuleSetModel"/>
          <xs:element ref="SequenceModel"/>
          <xs:element ref="SupportVectorMachineModel"/>
          <xs:element ref="TextModel"/>
          <xs:element ref="TimeSeriesModel"/>
          <xs:element ref="TreeModel"/>
        </xs:choice>
      </xs:sequence>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="version" type="xs:string" use="required"/>
  </xs:complexType>
</xs:element>
```

**ภาพที่ ข.3** แสดง PMML XSD

element MiningBuildTask เป็น element ที่ใช้อธิบายองค์ประกอบของ Training Data ที่ใช้สร้าง model ซึ่งในส่วนนี้ PMML ไม่ได้ใช้ ไม่จำเป็นต้องมีก็ได้ แต่ในบางกรณี จะมีประโยชน์สำหรับการ maintenance และการทำให้เข้าใจ model element นี้ควรใช้เก็บรายละเอียดของงานที่ถูกนิยามด้วยมาตรฐานอื่น เช่น ใน SQL หรือ Java

```
<xs:element name="MiningBuildTask">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

**ภาพที่ ข.4** แสดงนิยามของ element MiningBuildTask

element DataDictionary และ element TransformationDictionary ใช้สำหรับกำหนดชื่อ field ต่างๆ element อื่นๆใน model สามารถอ้างอิงไปยัง field เหล่านี้ได้ด้วยชื่อ สำหรับเอกสาร PMML ที่เก็บ model มากกว่า 1 model สามารถใช้งาน field ที่อยู่ใน TransformationDictionary ร่วมกันได้

mining function ใน PMML มีทั้งหมด 6 function ด้วยกัน แต่ละ model จะมี attribute functionName ที่ใช้ในการระบุ mining function

PMML version 4.0 มี function ใหม่เพิ่มขึ้นจาก version ก่อนหน้าอีก 1 function คือ timeSeries

```
<xs:simpleType name="MINING-FUNCTION">
  <xs:restriction base="xs:string">
    <xs:enumeration value="associationRules"/>
    <xs:enumeration value="sequences"/>
    <xs:enumeration value="classification"/>
    <xs:enumeration value="regression"/>
    <xs:enumeration value="clustering"/>
    <xs:enumeration value="timeSeries"/>
  </xs:restriction>
</xs:simpleType>
```

**ภาพที่ ข.5** แสดงนิยามของ mining function

ทุกๆ Model ของ PMML จะมีโครงสร้างของ top-level model element คล้ายๆกัน ตามรูปแบบต่อไปนี้

```
<xs:element name="ExampleModel">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="MiningSchema"/>
      <xs:element ref="Output" minOccurs="0"/>
      <xs:element ref="ModelStats" minOccurs="0"/>
      <xs:element ref="Targets" minOccurs="0"/>
      <xs:element ref="LocalTransformations" minOccurs="0" />
      ...
      <xs:element ref="ModelVerification" minOccurs="0"/>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="modelName" type="xs:string" use="optional"/>
    <xs:attribute name="functionName" type="MINING-FUNCTION" use="required"/>
    <xs:attribute name="algorithmName" type="xs:string" use="optional"/>
  </xs:complexType>
</xs:element>
```

**ภาพที่ ข.6** แสดงนิยาม top-level model element

element MiningSchema ใช้เก็บรายการของ field ต่างๆ ที่ใช้ใน model นั้นๆ ซึ่ง field เหล่านี้จะถูกนิยามเอาไว้แล้วใน DataDictionary

element Output ใช้เก็บค่าผลลัพธ์ต่างๆ เช่น ค่าความเชื่อมั่น หรือ ค่าข้อมูลที่เป็นไปได้

element ModelStats ใช้เก็บสถิติของ field ที่ใช้

element Targets ใช้เก็บข้อมูลที่เกี่ยวข้องกับค่าที่เป็นเป้าหมาย

element LocalTransformations ใช้เก็บ derived field

element ModelVerification ใช้เก็บข้อมูลตัวอย่าง และผลลัพธ์ของ model

element Output, ModelStats, Targets , LocalTransformations, ModelVerification จะมีหรือไม่มีก็ได้ แต่ MiningSchema ต้องมี

attribute modelName ใช้เก็บชื่อ model

attribute functionName และ algorithmName ใช้เก็บชนิดของ model เพื่อระบุ algorithm ที่ใช้สร้าง model

## ข.2 ชนิดข้อมูลพื้นฐาน

ชนิดข้อมูล NUMBER เป็นชนิดข้อมูลที่ใช้เก็บตัวเลข ซึ่งตัวเลขอาจมีเครื่องหมายลบ, fraction และ exponent, สนับสนุนตัวเลขที่ถูกแทนด้วย INF, -INF และ NaN

```
<xs:simpleType name="NUMBER">
  <xs:restriction base="xs:double">
  </xs:restriction>
</xs:simpleType>
```

**ภาพที่ ข.7** แสดงนิยามชนิดข้อมูล NUMBER

ชนิดข้อมูล INT-NUMBER ใช้เก็บข้อมูล integer

```
<xs:simpleType name="INT-NUMBER">
  <xs:restriction base="xs:integer">
  </xs:restriction>
</xs:simpleType>
```

**ภาพที่ ข.8** แสดงนิยามชนิดข้อมูล INT-NUMBER

ชนิดข้อมูล REAL- NUMBER ใช้เก็บข้อมูลตัวเลข ชนิด float, long, double, scientific notation เช่น 1.23e4 แต่ไม่สนับสนุน INF, -INF และ NaN

```
<xs:simpleType name="REAL-NUMBER">
  <xs:restriction base="xs:double">
  </xs:restriction>
</xs:simpleType>
```

**ภาพที่ ข.9** แสดงนิยามชนิดข้อมูล REAL-NUMBER

ชนิดข้อมูล PROB-NUMBER ใช้เก็บข้อมูลชนิด REAL-NUMBER ที่มีค่าอยู่ในช่วงตั้งแต่ 0.0 ถึง 1.0 ปกติแล้วมักใช้เป็นตัวบอกค่าความเป็นไปได้

```
<xs:simpleType name="PROB-NUMBER">
  <xs:restriction base="xs:decimal">
  </xs:restriction>
</xs:simpleType>
```

**ภาพที่ ข.10** แสดงนิยามชนิดข้อมูล PROB-NUMBER

ชนิดข้อมูล PERCENTAGE-NUMBER ใช้เก็บข้อมูลชนิด REAL-NUMBER ที่มีค่าอยู่ในช่วงตั้งแต่ 0.0 ถึง 100.0

```
<xs:simpleType name="PERCENTAGE-NUMBER">
  <xs:restriction base="xs:decimal">
  </xs:restriction>
</xs:simpleType>
```

**ภาพที่ ข.11** แสดงนิยามชนิดข้อมูล PERCENTAGE-NUMBER

สังเกตได้ว่า entities เหล่านี้ ไม่มีผลกับ XML Parser ในการตรวจสอบชนิดข้อมูล แต่อย่างไรก็ตามในเอกสาร PMML จำเป็นที่จะต้องมีการกำหนดชนิดข้อมูลเหล่านี้

หลายๆ element จะมีการอ้างอิงไปยัง field ที่ประกาศใน schema syntax สังเกตได้ว่า model สามารถอ้างอิงไปยัง MiningFields ที่อยู่ใน MiningSchema หรืออ้างอิงไปยัง DerivedFields ตามที่มีการกำหนดอยู่ใน TransformationDictionary หรือ LocalTransformations

```
<xs:simpleType name="FIELD-NAME">
  <xs:restriction base="xs:string">
  </xs:restriction>
</xs:simpleType>
```

**ภาพที่ ข.12** แสดงนิยามชนิดข้อมูล FIELD-NAME

## ข.3 TreeModel

TreeModel ใน PMML ใช้นิยามโครงสร้างการทำนายผล หรือการแบ่งกลุ่ม แต่ละ node แสดงถึงการนิยาม rule ที่ใช้สำหรับเลือก node ถัดไป หรือแตกกิ่งต่อไป

### ข.3.1 โครงสร้าง TreeModel

```
<xs:element name="TreeModel">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="MiningSchema"/>
      <xs:element ref="Output" minOccurs="0" />
      <xs:element ref="ModelStats" minOccurs="0"/>
      <xs:element ref="ModelExplanation" minOccurs="0"/>
      <xs:element ref="Targets" minOccurs="0" />
      <xs:element ref="LocalTransformations" minOccurs="0" />
      <xs:element ref="Node"/>
      <xs:element ref="ModelVerification" minOccurs="0"/>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="modelName" type="xs:string" />
    <xs:attribute name="functionName" type="MINING-FUNCTION" use="required" />
    <xs:attribute name="algorithmName" type="xs:string" />
    <xs:attribute name="missingValueStrategy" type="MISSING-VALUE-STRATEGY" default="none"/>
    <xs:attribute name="missingValuePenalty" type="PROB-NUMBER" default="1.0"/>
    <xs:attribute name="noTrueChildStrategy" type="NO-TRUE-CHILD-STRATEGY" default="returnNullPrediction" />
    <xs:attribute name="splitCharacteristic" default="multiSplit">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="binarySplit"/>
          <xs:enumeration value="multiSplit"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
  </xs:complexType>
</xs:element>
```

**ภาพที่ ข.13** แสดงนิยามโครงสร้าง TreeModel

จากภาพข้างบน

TreeModel คือ จุดเริ่มต้นของการนิยามโมเดลต้นไม้

Node เป็น element ที่ใช้ในการกำหนดว่า node ปัจจุบันเป็นใบหรือไม่ ถ้าไม่จะมีการนิยามการแตกกิ่งต้นไม้ต่อไป

modelName คือ ชื่อ model

missingValueStrategy ระบุกลยุทธ์หรือวิธีการในการจัดการกับ missing value

missingValuePenalty กำหนดวิธีการคำนวณค่าความเชื่อมั่น เมื่อข้อมูลเป็น missing value

noTrueChildStrategy กำหนดวิธีทำนายผลข้อมูล หากเกิดสถานการณ์ที่การประเมินผลไม่สามารถเดินทางไปถึง leaf node ได้

splitCharacteristic เป็นการบอกว่า model ต้นไม้ที่สร้างเป็น binary tree หรือไม่ ถ้าใช่ value = binarySplit ถ้าไม่ใช่ value = multiSplit ( ค่า default คือ multiSplit )

แต่ละ Node จะประกอบไปด้วย

```
<xs:element name="Node">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded" />
      <xs:group ref="PREDICATE" />
      <xs:choice>
        <xs:sequence>
          <xs:element ref="Partition" minOccurs="0"/>
          <xs:element ref="ScoreDistribution" minOccurs="0" maxOccurs="unbounded"/>
          <xs:element ref="Node" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:group ref="EmbeddedModel"/>
      </xs:choice>
    </xs:sequence>
    <xs:attribute name="id" type="xs:string"/>
    <xs:attribute name="score" type="xs:string"/>
    <xs:attribute name="recordCount" type="NUMBER"/>
    <xs:attribute name="defaultChild" type="xs:string"/>
  </xs:complexType>
</xs:element>
```

**ภาพที่ ข.14** แสดงนิยาม element Node

จากภาพข้างบน

Partition เป็น element ที่ใช้จัดเก็บข้อมูลต่างๆทั้งหมด ของแต่ละ Node

id คือ ค่าของ Node เพื่อใช้ในการระบุ Node โดยในแต่ละ Node จะมีความเป็นเอกลักษณ์หรือแตกต่างกันภายใน model

score ใช้เก็บค่าข้อมูลที่ใช้ทำนายผลของ Node นั้นๆ

recordCount เก็บจำนวน training data ที่อยู่ในแต่ละ Node

defaultChild สามารถใช้ได้ในกรณีเดียว คือกรณีที่ missingValueStrategy ถูกกำหนดให้เป็น defaultChild ใน TreeModel element วิธีการคือ กำหนดค่า id ของ node ลูกที่ต้องการเลือกให้กับ attribute defaultChild ดังนั้นในกรณีที่ข้อมูลมีค่าเป็น missing value model จะท่อง tree ต่อไปยัง Node ลูกที่มี id ตรงกับที่กำหนดไว้ใน attribute defaultChild

EmbeddedModel ใช้ได้ในกรณีที่มี model อื่นมาเป็นส่วนประกอบด้วย ในส่วนนี้จะเป็นตัวอ้างอิงไปยัง model ที่ถูกฝังอยู่ใน Node นั้นๆ

### ข.3.2 การทำนายผลข้อมูล (Predicates)

แต่ละ Node จะมี PREDICATE เพียงตัวเดียวเท่านั้น ซึ่งอาจจะเป็น SimplePredicate, SetPredicate, CompoundPredicate, True หรือ False ก็ได้

```
<xs:group name="PREDICATE">
  <xs:choice>
    <xs:element ref="SimplePredicate" />
    <xs:element ref="CompoundPredicate" />
    <xs:element ref="SimpleSetPredicate" />
    <xs:element ref="True" />
    <xs:element ref="False" />
  </xs:choice>
</xs:group>

<xs:element name="SimplePredicate">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
    <xs:attribute name="field" type="FIELD-NAME" use="required"/>
    <xs:attribute name="operator" use="required">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="equal"/>
          <xs:enumeration value="notEqual"/>
          <xs:enumeration value="lessThan"/>
          <xs:enumeration value="lessOrEqual"/>
          <xs:enumeration value="greaterThan"/>
          <xs:enumeration value="greaterOrEqual"/>
          <xs:enumeration value="isMissing"/>
          <xs:enumeration value="isNotMissing"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="value" type="xs:string"/>
  </xs:complexType>
</xs:element>
```

**ภาพที่ ข.15** แสดงนิยาม PREDICATE

จากภาพข้างบน

SimplePredicate เป็น element ที่นิยาม rule ในรูปแบบของนิพจน์ boolean ซึ่งประกอบไปด้วย field, operator (booleanOperator) และ value

field เป็น attribute ที่ใช้เก็บชื่อ field ที่ใช้เป็นเงื่อนไขในการทำนาย

operator แบ่งออกเป็น equal (=), notEqual (≠), lessThan (<), lessOrEqual (≤), greaterThan (>), greaterOrEqual (≥)

value เป็น attribute ที่ใช้เก็บข้อมูลที่จะนำมาใช้ประเมินผลเปรียบเทียบ

**ตัวอย่างที่ ข.1** วิธีการเขียน age < 30 สามารถเขียนได้หลายวิธีดังนี้

```
<SimplePredicate field="age" operator="lessThan" value="30" >
<SimplePredicate value="30" operator="lessThan" field="age" >
<SimplePredicate operator="lessThan" value="30" field="age" >
```

**ภาพที่ ข.16** แสดงตัวอย่างที่ ข.1

Compound predicate

```
<xs:element name="CompoundPredicate">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded" />
      <xs:sequence minOccurs="2" maxOccurs="unbounded">
        <xs:group ref="PREDICATE" />
      </xs:sequence>
    </xs:sequence>
    <xs:attribute name="booleanOperator" use="required">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="or"/>
          <xs:enumeration value="and"/>
          <xs:enumeration value="xor"/>
          <xs:enumeration value="surrogate"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
  </xs:complexType>
</xs:element>
```

**ภาพที่ ข.17** แสดงนิยาม element CompoundPredicate

จากภาพข้างบน

Compound predicate เป็นการรวม element ที่ถูกนิยามตาม entity PREDICATE ตั้งแต่ 2 element ขึ้นไป แต่ละ element มีความสัมพันธ์กันด้วย booleanOperator ได้แก่ and, or, xor หรือ surrogate

booleanOperator ได้แก่ and, or, xor, surrogate ในส่วนของ surrogate อธิบายด้วยตัวอย่างดังนี้ surrogate (a, b) เทียบเท่าได้กับ if not unknown (a) then a else b

Simple set predicates

```
<xs:element name="SimpleSetPredicate">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded" />
      <xs:element ref="Array"/>
    </xs:sequence>
    <xs:attribute name="field" type="FIELD-NAME" use="required"/>
    <xs:attribute name="booleanOperator" use="required">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="isIn"/>
          <xs:enumeration value="isNotIn"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
  </xs:complexType>
</xs:element>
```

**ภาพที่ ข.18** แสดงนิยาม element SimpleSetPredicate

จากภาพข้างบน

SimpleSetPredicate เก็บค่าข้อมูลของ field เป็นเซตข้อมูลในภาพของ array

booleanOperator ได้แก่ isIn และ isNotIn

True เป็น element ที่ใช้ระบุค่าคงที่ boolean เป็น TRUE

```
<xs:element name="True">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

**ภาพที่ ข.19** แสดงนิยาม element True

False เป็น element ที่ใช้ระบุค่าคงที่ boolean เป็น FALSE

```
<xs:element name="False">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

**ภาพที่ ข.20** แสดงนิยาม element False

ตัวอย่างที่ ข.2 ((temperature >60)and(temperature<100)and(outlook="overcast")) จะได้

```
<CompoundPredicate booleanOperator="and" >
  <SimplePredicate field="temperature" operator="greaterThan"
                   value="60" />
  <SimplePredicate field="temperature" operator="lessThan"
                   value="100" />
  <SimplePredicate field="outlook" operator="equal"
                   value="overcast"/>
</CompoundPredicate>
```

**ภาพที่ ข.21** แสดงตัวอย่างที่ ข.2

ตัวอย่างที่ ข.3 (((temperature < 90) and (temperature > 50)) or (humidity ≥ 80)) จะได้

```
<CompoundPredicate booleanOperator="or" >
  <CompoundPredicate booleanOperator="and" >
    <SimplePredicate field="temperature" operator="lessThan" value="90" />
    <SimplePredicate field="temperature" operator="greaterThan" value="50" />
  </CompoundPredicate>
  <SimplePredicate field="humidity" operator="greaterOrEqual" value="80" />
</CompoundPredicate>
```

**ภาพที่ ข.22** แสดงตัวอย่างที่ ข.3

ตัวอย่างที่ ข.4

```
<CompoundPredicate booleanOperator="surrogate" >
  <CompoundPredicate booleanOperator="and" >
    <SimplePredicate field="temperature" operator="lessThan"
                     value="90" />
    <SimplePredicate field="temperature" operator="greaterThan"
                     value="50" />
  </CompoundPredicate>
  <SimplePredicate field="humidity" operator="greaterOrEqual"
                     value="80" />
  <False/>
</CompoundPredicate>
```

**ภาพที่ ข.23** แสดงตัวอย่างที่ ข.4

จากตัวอย่างจะทำการตรวจสอบตามเงื่อนไข (temperature < 90) and (temperature > 50) ถ้าการประเมินผลเป็น TRUE หรือ FALSE ก็จะได้ผลลัพธ์ของ surrogate ตามที่ประเมิน แต่ถ้าการประเมินผลเป็น UNKNOW เพราะค่าของ field temperature เป็น missing value ก็จะทำการประเมินตามเงื่อนไขที่ 2 คือ humidity ≥ 80 ถ้าค่าของ humidity เป็น missing value ผลลัพธ์สุดท้ายที่ได้ คือ FALSE

ScoreDistribution

element นี้ประกอบไปด้วยรายการข้อมูลที่ทำการทำนายผล

```
<xs:element name="ScoreDistribution">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
    <xs:attribute name="value" type="xs:string" use="required"/>
    <xs:attribute name="recordCount" type="NUMBER" use="required"/>
    <xs:attribute name="confidence" type="PROB-NUMBER"/>
  </xs:complexType>
</xs:element>
```

**ภาพที่ ข.24** แสดงนิยาม element ScoreDistribution

จากภาพข้างบน

value ใช้เก็บค่าข้อมูลที่ทำนายผล

recordCount ใช้เก็บจำนวนข้อมูลที่มีผลการทำนายเดียวกันกับค่าใน value

confidence เก็บค่าความเชื่อมั่นของผลการทำนาย

**Missing Value Strategies และ Penalties**

missingValueStrategy เป็น element ที่บอกถึงวิธีการที่ต้องใช้เมื่อการประเมินผลของ Node มีค่าเป็น UNKNOWN ในระหว่างทำนายผลข้อมูล

```
<xs:simpleType name="MISSING-VALUE-STRATEGY">
  <xs:restriction base="xs:string">
    <xs:enumeration value="lastPrediction" />
    <xs:enumeration value="nullPrediction" />
    <xs:enumeration value="defaultChild" />
    <xs:enumeration value="weightedConfidence" />
    <xs:enumeration value="aggregateNodes" />
    <xs:enumeration value="none" />
  </xs:restriction>
</xs:simpleType>
```

**ภาพที่ ข.25** แสดงนิยาม missingValueStrategy

จากภาพข้างบน

lastPrediction ถ้าการประเมินผลของ Node มีค่าเป็น UNKNOWN การทำนายผลข้อมูลจะสิ้นสุดทันที และผลการทำนายจะเป็นข้อมูล class ที่มีค่าความเชื่อมั่นสูงสุด

nullPrediction ถ้าการประเมินผลของ Node มีค่าเป็น UNKNOWN ผลการทำนายจะถูกยกเลิก และไม่ทำนายผล (ให้ค่า no prediction)

defaultChild ทุกๆ Node ที่ไม่ใช่ใบ จะถูกกำหนดค่า defaultChilde เอาไว้ เพื่อให้การท่องต้นไม้มีความต่อเนื่อง ถ้าการประเมินผลของ Node มีค่าเป็น UNKNOWN ถ้าหากเป็นเช่นนั้น Node ที่ถูกระบุไว้ใน defaultChilde จะเป็น Node ถัดไปที่ถูกประเมิน

weightedCondidence ถ้าการประเมินผลของ Node มีค่าเป็น UNKNOWN ในขณะท่องต้นไม้ ค่าความเชื่อมั่นของแต่ละ class ใน Node นั้น และ class ของ Node อื่นที่เป็น sibling Nodes (ยกเว้น sibling ที่มี field ที่เป็นเงื่อนไขต่างกันกับ Node) จะถูกนำมาใช้ในการคำนวณ เพื่อประเมินผลข้อมูล ค่าความเชื่อมั่นที่ได้จากการประเมินผลของ Node เกิดจากผลรวมของค่าความเชื่อมั่น ของแต่ละ class ของแต่ละ sibling Node คูณด้วยอัตราส่วนของจำนวนข้อมูลของ sibling กับจำนวนข้อมูลภายใน Node ข้อมูล class ที่มีค่าความเชื่อมั่นสูงสุดจะเป็นผลการทำนาย

aggregateNodes ถ้าการประเมินผลของ Node มีค่าเป็น UNKNOWN วิธี aggregateNode จะเป็นการรวม Node ทุก Node ที่เป็นใบ ที่อาจจะท่องไปถึง โดยดูจาก attribute เงื่อนไขที่ตรงกัน ผลรวมของค่าที่เป็นไปได้ ค่าใดมี recordCount สูงสุด คือ ผลการทำนาย

none มีการกำหนดผลการทำนายไว้แล้ว ถ้าทำการประเมินทุกๆเงื่อนไขแล้ว แต่ไม่มีค่าใดถูกส่งกลับ เนื่องจากเป็น missing value ผลการทำนายที่ถูกกำหนดไว้จะถูกส่งกลับ ตรงกันข้ามกับ lastPrediction ที่จะหยุดการประเมินทันทีที่ไม่สามารถประเมินผลได้

กรณีที่การประเมินผลไม่สามารถดำเนินการต่อไปได้ noTrueChildStrategy จะถูกเรียกใช้เพื่อจัดการกับปัญหาดังกล่าว

ในระหว่างการประเมินผลภายใน Node ถ้าไม่มีเงื่อนไขของ subNode ใดที่ให้ค่าเป็น TRUE และ missingValueStrategy (ถ้ามีการกำหนด) ไม่ถูกเรียกใช้ attribute noTrueChildStrategy ของ TreeModel จะเป็นตัวกำหนดว่าจะทำอะไรต่อไป

```
<xs:simpleType name="NO-TRUE-CHILD-STRATEGY">
  <xs:restriction base="xs:string">
    <xs:enumeration value="returnNullPrediction" />
    <xs:enumeration value="returnLastPrediction" />
  </xs:restriction>
</xs:simpleType>
```

**ภาพที่ ข.26** แสดง noTrueChildStrategy

จากภาพข้างบน

returnNullPrediction คือ การส่งค่ากลับว่าไม่ทำนายผล หรือ no prediction (default)

returnLastPrediction ถ้า parent มี attribute score ค่าที่ส่งกลับจะเป็นค่าตาม attribute score แต่ถ้าไม่จะส่งค่ากลับเป็น no prediction

ตัวอย่างที่ ข.5

```
<Node id="N1" score="0">
  <True />
  <Node id="T1" score="1">
    <SimplePredicate field="prob1" operator="greaterThan" value="0.33"/>
  </Node>
</Node>
```

**ภาพที่ ข.27** แสดงตัวอย่างที่ ข.5

จากตัวอย่างข้างบน ถ้าการประเมินผลมาถึง Node N1 แล้วข้อมูล field prob1 มีค่าน้อยกว่า หรือเท่ากับ 0.33 noTrueChildStrategy จะเป็นตัวกำหนดสิ่งที่ต้องทำ ตามที่ได้ระบุไว้ ถ้าระบุเป็น returnNullPrediction จะส่งค่ากลับเป็น no prediction ถ้าระบุเป็น returnLastPrediction จะส่งค่ากลับเป็น ค่า score ของ N1 (0)

### ข.3.3 ตัวอย่าง Tree Model

ตัวอย่าง Tree Model ที่มีการกำหนด Missing Value Strategies

```
<?xml version="1.0" ?>
<PMML version="4.0" xmlns="http://www.dmg.org/PMML-4_0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Header copyright="www.dmg.org" description="A very small
      tree model to demonstrate missing value handling and confidence calculation."/>
  <DataDictionary numberOfFields="4" >
    <DataField name="temperature" optype="continuous" dataType="double"/>
    <DataField name="humidity" optype="continuous" dataType="double"/>
    <DataField name="outlook" optype="categorical" dataType="string">
      <Value value="sunny"/>
      <Value value="overcast"/>
      <Value value="rain"/>
    </DataField>
    <DataField name="whatIdo" optype="categorical" dataType="string">
      <Value value="will play"/>
      <Value value="may play"/>
      <Value value="no play"/>
    </DataField>
  </DataDictionary>
  <TreeModel modelName="golfing" functionName="classification" missingValueStrategy="weightedConfidence" >
    <MiningSchema>
      <MiningField name="temperature"/>
      <MiningField name="humidity"/>
      <MiningField name="outlook"/>
      <MiningField name="whatIdo" usageType="predicted"/>
    </MiningSchema>
    <Node id="1" score="will play" recordCount="100" defaultChild="2">
      <True/>
      <ScoreDistribution value="will play" recordCount="60" confidence="0.6" />
      <ScoreDistribution value="may play" recordCount="30" confidence="0.3" />
      <ScoreDistribution value="no play" recordCount="10" confidence="0.1" />
      <Node id="2" score="will play" recordCount="50" defaultChild="3" >
        <SimplePredicate field="outlook" operator="equal" value="sunny"/>
        <ScoreDistribution value="will play" recordCount="40" confidence="0.8" />
        <ScoreDistribution value="may play" recordCount="2" confidence="0.04" />
        <ScoreDistribution value="no play" recordCount="8" confidence="0.16" />
        <Node id="3" score="will play" recordCount="40">
          <CompoundPredicate booleanOperator="surrogate" >
            <SimplePredicate field="temperature" operator="greaterOrEqual" value="50" />
            <SimplePredicate field="humidity" operator="lessThan" value="80" />
          </CompoundPredicate>
          <ScoreDistribution value="will play" recordCount="36" confidence="0.9" />
          <ScoreDistribution value="may play" recordCount="2" confidence="0.05" />
          <ScoreDistribution value="no play" recordCount="2" confidence="0.05" />
        </Node>
        <Node id="4" score="no play" recordCount="10" >
          <CompoundPredicate booleanOperator="surrogate" >
            <SimplePredicate field="temperature" operator="lessThan" value="50"/>
            <SimplePredicate field="humidity" operator="greaterOrEqual" value="80" />
          </CompoundPredicate>
          <ScoreDistribution value="will play" recordCount="4" confidence="0.4" />
          <ScoreDistribution value="may play" recordCount="0" confidence="0.0" />
          <ScoreDistribution value="no play" recordCount="6" confidence="0.6" />
        </Node>
      </Node>
      <Node id="5" score="may play" recordCount="50" >
        <CompoundPredicate booleanOperator="or" >
          <SimplePredicate field="outlook" operator="equal" value="overcast" />
          <SimplePredicate field="outlook" operator="equal" value="rain" />
        </CompoundPredicate>
        <ScoreDistribution value="will play" recordCount="20" confidence="0.4" />
        <ScoreDistribution value="may play" recordCount="28" confidence="0.56" />
        <ScoreDistribution value="no play" recordCount="2" confidence="0.04" />
      </Node>
    </Node>
  </TreeModel>
</PMML>
```



**ภาพที่ ข.28** แสดงตัวอย่าง Tree Model

ตัวอย่างที่ ข.6

ถ้า temperature = 45, outlook = “sunny”, humidity = 60 ไม่มี missing values

การท่องต้นไม้จะท่องไปถึง Node 4 และทำนายผลเป็น “no play” ค่าความเชื่อมั่น คือ 0.6

ตัวอย่างที่ ข.7

กำหนดให้ attribute missingValueStrategy มีค่าเป็น weightedConfidence

ถ้า outlook = “sunny” แต่ไม่ทราบค่า temperature และ humidity

การประเมินผลจะนำไปสู่ Node 2 แต่เนื่องจากไม่ทราบค่า temperature และ humidity การประเมินผลของ Node 3 จะได้เป็น UNKNOWN ดังนั้น missingValueHandlingStrategy weightedConfidence จะถูกเรียกให้มาทำงาน

ค่าความเชื่อมั่นของแต่ละ class ของแต่ละ child node ของ node 2 ที่มี attribute ที่เป็นเงื่อนไขตรงกัน (node 3 และ 4) จะถูกนำมาใช้ในการคำนวณ

Node 3 confidences:

```
conf("will play")=0.9
conf("may play")=0.05
conf("no play")=0.05
```

Node 4 confidences:

```
conf("will play")=0.4
conf("may play")=0.0
conf("no play")=0.6
```

Node 3 มีจำนวน 40 records และ Node 4 มีจำนวน 10 records จะได้

Node 2 confidences:

```
conf("will play")=(40/50)*0.9+(10/50)*0.4=0.72 + 0.08 = 0.8
conf("may play")=(40/50)*0.05+(10/50)*0.0=0.04
conf("will play")=(40/50)*0.05+(10/50)*0.6=0.04 + 0.12 = 0.16
```

การทำนายผลในกรณีนี้ คือ เลือกข้อมูล class ที่มีค่าความเชื่อมั่นสูงสุดมาทำนายผล คือ “will play” (0.16)

ตัวอย่างที่ ข.8

กำหนดให้ attribute missingValueStrategy มีค่าเป็น weightedConfidence

ถ้าไม่ทราบค่าทั้ง outlook, humidity, temperature

การประเมินผลของ Node 2 จะเป็น UNKNOWN

Node 2 confidences:

```
conf("will play")=0.8
conf("may play")=0.04
conf("no play")=0.16
```

Node 5 confidences

```
conf("will play")=0.4
conf("may play")=0.56
conf("no play")=0.04
```

Node 1 confidences:

```
conf("will play")=(50/100) * 0.8 + (50/100) * 0.4=0.4 + 0.2 = 0.6
conf("may play")=(50/100) * 0.04 + (50/100) * 0.56=0.02 + 0.28 = 0.3
conf("will play")=(50/100) * 0.16 +(50/100)* 0.04=0.08 + 0.02 = 0.1
```

กรณีนี้ทำนายผลเป็น “will play”

ตัวอย่างที่ ข.9

สมมติ เรากำหนดให้ attribute missingValueStrategy มีค่าเป็น defaultChild แล้วเพิ่มด้วย attribute missingValuePenalty พร้อมกับกำหนดค่าเป็น 0.8 ใน TreeModel

ถ้า temperature = 40, humidity = 70 แต่ไม่ทราบค่า outlook

การประเมินผลที่ Node 2 จะได้ UNKNOWN เนื่องจาก outlook เป็น missing value ดังนั้น missingValueHandlingStrategy defaultChild จึงถูกเรียกใช้ให้ทำงาน การประเมินผลจะดำเนินการต่อ โดยการเลือก Node ที่ได้กำหนดไว้แล้วใน defaultChild (node 2) ในกรณีนี้ผลการทำนาย คือ “no play" แต่ค่าความเชื่อมั่น จะได้ 0.6 คูณด้วย missingValuePenalty (0.8) นั่นคือ 0.48 นั่นเอง

ตัวอย่างที่ ข.10

สมมติ เรากำหนดให้ attribute missingValueStrategy มีค่าเป็น defaultChild แล้วเพิ่มด้วย attribute missingValuePenalty พร้อมกับกำหนดค่าเป็น 0.8 ใน TreeModel

ถ้า humidity = 70 แต่ไม่ทราบค่า outlook และ temperature

การประเมินผลที่ Node 2 จะได้เป็น UNKNOWN เนื่องจาก outlook มีค่าเป็น missing value ดังนั้น missingValueHandlingStrategy defaultChild จะถูกใช้เพื่อให้การประเมินผลเป็นไปอย่างต่อเนื่อง โดยการเลือก Node ตาม defaultChild คือ Node 2 จาก Node 2 Node ที่ถูกเลือกถัดไป คือ Node 3 เนื่องจากการทำงานของ surrorgate โดย humidity เป็นไปตามเงื่อนไขของ Node 3 ดังนั้นการทำนายผลที่ได้ คือ “will play” แต่ค่าความเชื่อมั่นที่ส่งกลับ คำนวณจาก 0.9 คูณ

ด้วย ค่าของ missingValuePenalty (0.8) สำหรับ 2 Node (Node 1, Node 2) ที่มีค่าเป็น missing value เพราะฉะนั้น จะได้ 0.9*0.8*0.8 = 0.576

ตัวอย่างที่ ข.11

สมมติ เรากำหนดให้ attribute missingValueStrategy มีค่าเป็น lastPrediction

ถ้า outlook = "sunny" แต่ไม่ทราบค่า temperature และ humidity

การประเมินผลจะไม่ถูกดำเนินการต่อจาก Node 2 เพราะการประเมินผลที่ Node 3 และ Node 4 ได้ผลลัพธ์เป็น UNKNOWN ดังนั้น missingValueHandlingStrategy lastPrediction จะถูกเรียกใช้ และผลการทำนาย คือ "will play" มีค่าความเชื่อมั่นเป็น 0.8

ตัวอย่างที่ ข.12

สมมติ เรากำหนดให้ attribute missingValueStrategy มีค่าเป็น nullPrediction

ถ้า outlook = "sunny" แต่ไม่ทราบค่า temperature และ humidity

การประเมินผลจะหยุดอยู่ที่ Node 2 เพราะ การประเมินผลที่ Node 3 ได้ผลลัพธ์เป็น UNKNOWN ดังนั้น missingValueHandlingStrategy nullPrediction จะถูกเรียกใช้ และส่งค่ากลับเป็น no prediction

ตัวอย่างที่ ข.13

สมมติ เรากำหนดให้ attribute missingValueStrategy มีค่าเป็น aggregateNodes

ถ้า temperature = "45" และ humidity = "90" แต่ไม่ทราบค่า outlook

การประเมินผลที่ Node 2 จะได้ค่าเป็น UNKNOWN ดังนั้น missingValueHandlingStrategy aggregateNodes จะถูกเรียกใช้ โดยสมมติว่า Node 2 ได้รับการประเมินผลต่อ (สมมติว่า เงื่อนไขของ Node 2 มีค่าเป็น TRUE) ภายใต้สมมติฐานนี้ พิจารณาเงื่อนไขของ Node 3 จะได้ค่าเป็น FALSE แต่เงื่อนไขของ Node 4 จะได้ค่าเป็น TRUE ส่วน sibling nodes ที่เหลือของ Node 2 จะต้องได้รับการประเมินผลไปด้วย คือ Node 5 และผลลัพธ์ที่ได้ คือ TRUE

Node 4 `recordCounts`:

```
recordCount("will play")=4
recordCount("may play")=0
recordCount("no play")=6
```

Node 5 `recordCounts`:

```
recordCount("will play")=20
recordCount("may play")=28
recordCount("no play")=2
```

นำ recordCount ของทั้ง 2 Node บวกกัน จะได้

```
recordCount("will play")=4 + 20=24
recordCount("may play")=0 + 28=28
recordCount("no play")=6 + 2=8
```

ผลการทำนาย ดูจาก recordCount สูงสุด ในกรณีนี้ผลลัพธ์ คือ “may play”

ค่าความเชื่อมั่น คำนวณได้ดังนี้

```
confidence("may play")
      = recordCount("may play") / (recordCount("will play")
        + recordCount("may play") + recordCount("no play"))
      = 28 / (24 + 28 + 8)     = 28 / 60
      ≈ 0.47
```

ตัวอย่างที่ ข.14

สมมติ เรากำหนดให้ attribute missingValueStrategy มีค่าเป็น none

ถ้าไม่ทราบค่า age

ปกติแล้วค่าของ age จะครอบคลุมด้วย Node 2 และ Node 3 และไม่มีทางที่จะท่องไปถึง Node 4 แต่เนื่องจาก missing value ไม่น้อยกว่า 30 และไม่มากกว่าหรือเท่ากับ 30 ดังนั้น Node 4 จะถูกนำไปใช้เสมอในกรณีที่ age เป็น missing value

```
...
<TreeModel modelName="golfing" functionName="classification" missingValueStrategy="none" >
  ...
  <Node id="1" score="will play" recordCount="100" >
     <True/>
     <Node id="2" score="will play" recordCount="50" >
        <SimplePredicate field="age" operator="lessThan" value="30"/>
     </Node>
     <Node id="3" score="will not play" recordCount="20" >
        <SimplePredicate field="age" operator="greaterOrEqual" value="30"/>
     </Node>
     <Node id="4" score="will play" recordCount="30" >
        <True/>
     </Node>
  </Node>
  ...
```

**ภาพที่ ข.29** แสดงตัวอย่างที่ ข.14

# ภาคผนวก ค

# ข้อมูลที่ใช้ทำการทดลอง

# ข้อมูลที่ใช้ทำการทดลอง

โครงการพัฒนาระบบฉบับนี้ ได้ทำการทดสอบการทำงานของระบบ โดยใช้ชุดข้อมูล 2 ชุด ด้วยกัน ได้แก่ ชุดข้อมูลการเล่นเทนนิส และชุดข้อมูลเห็ด

## ค.1 ชุดข้อมูลการเล่นเทนนิส

Class Attribute คือ classes: Play, Don't Play

Non Class Attribute:

1. Outlook: Sunny, Overcast, Rain
2. Temp: Numerical
3. Humidity: Numerical
4. Windy: True, False

Training Data: (231 record)

| | |
|---|---|
| Sunny, 75, 70, True, Play | Sunny, 80, 90, True, Don't Play |
| Sunny, 85, 85, False, Don't Play | Sunny, 72, 95, False, Don't Play |
| Sunny, 69, 70, False, Play | Overcast, 72, 90,True, Play |
| Overcast, 83, 78, False, Play | Overcast, 64, 65, True, Play |
| Overcast, 81, 75, False, Play | Rain, 71, 80, True, Don't Play |
| Rain, 69, 70, True, Don't Play | Rain, 76, 80, False, Play |
| Rain, 68, 80, False, Play | Rain, 70, 96, False, Play |
| Sunny, 64, 80, False, Play | Sunny, 72, 90, True, Play |
| Rain, 71, 95, False, Play | Rain, 70, 80, True, Don't Play |
| Overcast, 70, 70, False, Play | Overcast, 69, 70, True, Play |
| Overcast, 76, 90, True, Play | Rain, 72, 95, False, Play |
| Sunny, 80, 85, True, Don't Play | Rain, 70, 79, True, Don't Play |
| Rain, 71, 80, True, Don't Play | Sunny, 84, 90, True, Don't Play |
| Sunny, 85, 83, False, Don't Play | Sunny, 70, 75, True, Play |

Sunny, 71, 79, False, Play
Overcast, 69, 65, True, Play
Rain, 75, 75 True, Don't Play
Overcast, 84, 80, True, Play
Rain, 68, 70, True, Don't Play
Sunny, 80, 95, False, Play
Overcast, 85, 70, True, Don't Play
Overcast, 71, 85, False, Play
Rain, 70, 85, False, Play
Sunny, 81, 65, True, Play
Sunny, 76, 83, False, Don't Play
Rain, 80, 82, True, Don't Play
Overcast, 71, 65, False, Play
Sunny, 72, 75, True, Play
Rain, 81, 82, True, Play
Rain, 67, 70, True, Don't Play
Sunny, 70, 70, False, Play
Sunny, 80, 70, False, Play
Sunny, 75, 90, False, Play
Overcast, 75, 80, True, Don't Play
Overcast, 71, 65, True, Don't Play
Sunny, 85, 85, False, Don't Play
Rain, 71, 70, True, Don't Play
Overcast, 83, 78, False, Play
Overcast, 84, 75, False, Play
Sunny, 76, 68, True, Play
Sunny, 69, 70, False, Play
Rain, 69, 82, False, Play
Rain, 75, 80, False, Play
Overcast, 75, 90, True, Play
Overcast, 70, 83, False, Play
Rain, 68, 70, True, Don't Play
Overcast, 75, 75, False, Play
Sunny, 71, 85, False, Don't Play
Sunny, 72, 83, False, Play
Overcast, 68, 82, False, Play
Rain, 75, 90, True, Don't Play
Sunny, 83, 74, False, Play
Sunny, 67, 90, True, Don't Play
Rain, 84, 90, False, Play
Overcast, 67, 90, False, Play
Overcast, 81, 75, False, Play
Sunny, 68, 87, False, Play
Rain, 80, 75, False, Play
Rain, 81, 75, True, Don't Play
Sunny, 80, 65, False, Play
Sunny, 70, 70, True, Play
Overcast, 75, 90, True, Don't Play
Overcast, 80, 87, False, Play
Overcast, 80, 70, True, Don't Play
Rain, 71, 80, True, Don't Play
Overcast, 81, 75, False, Play
Overcast, 64, 65, True, Play
Sunny, 80, 87, False, Don't Play
Sunny, 69, 83, False, Play
Rain, 72, 79, False, Don't Play
Rain, 68, 90, True, Don't Play
Rain, 68, 80, False, Play

Rain, 75, 74, False, Play
Overcast, 68, 70, False, Play
Overcast, 72, 85, True, Don't Play
Overcast, 80, 79, True, Don't Play
Sunny, 85, 90, False, Play
Sunny, 85, 85, False, Don't Play
Sunny, 75, 87, True, Play
Rain, 80, 65, False, Play
Rain, 71, 80, True, Don't Play
Rain, 71, 80, False, Play
Sunny, 71, 82, False, Play
Sunny, 76, 87, True, Don't Play
Sunny, 72, 95, False, Play
Overcast, 81, 78, True, Don't Play
Overcast, 84, 75, True, Play
Sunny, 76, 68, True, Don't Play
Sunny, 80, 65, False, Play
Rain, 76, 70, False, Play
Rain, 70, 80, True, Don't Play
Rain, 75, 75, True, Play
Sunny, 71, 70, False, Don't Play
Sunny, 80, 96, False, Play
Overcast, 76, 80, False, Play
Overcast, 64, 68, True, Play
Sunny, 70, 79, False, Play
Sunny, 76, 65, False, Don't Play
Sunny, 72, 80, True, Don't Play
Sunny, 70, 83, False, Play
Overcast, 84, 96, False, Play
Overcast, 67, 85, False, Play
Overcast, 71, 65, False, Don't Play
Overcast, 72, 90, True, Play
Sunny, 67, 78, False, Play
Sunny, 68, 65, True, Don't Play
Sunny, 72, 95, False, Don't Play
Rain, 71, 65, False, Play
Rain, 70, 96, False, Play
Rain, 80, 70, True, Don't Play
Rain, 76, 79, True, Don't Play
Sunny, 70, 85, False, Play
Sunny, 70, 78, False, Don't Play
Overcast, 81, 78, False, Play
Overcast, 75, 80, False, Play
Overcast, 68, 79, False, Play
Sunny, 72, 80, False, Play
Sunny, 69, 95, True, Don't Play
Rain, 83, 90, True, Don't Play
Rain, 69, 78, True, Don't Play
Sunny, 67, 79, False, Play
Sunny, 69, 78, False, Play
Sunny, 75, 79, True, Don't Play
Overcast, 84, 78, True, Don't Play
Sunny, 75, 78, True, Play
Sunny, 71, 90, True, Don't Play
Sunny, 67, 79, True, Don't Play
Sunny, 80, 70, False, Play
Overcast, 69, 70, True, Don't Play
Overcast, 83, 74, True, Don't Play

Rain, 68, 79, False, Play
Rain, 83, 79, True, Don't Play
Rain, 84, 65, False, Play
Sunny, 70, 90, True, Play
Sunny, 75, 75, False, Play
Sunny, 75, 90, True, Play
Rain, 69, 82, False, Play
Overcast, 64, 74, True, Don't Play
Sunny, 68, 70, False, Play
Sunny, 69, 87, True, Don't Play
Sunny, 80, 90, True, Don't Play
Overcast, 85, 96, False, Play
Overcast, 67, 83, False, Play
Rain, 64, 68, True, Play
Rain, 69, 82, True, Don't Play
Sunny, 83, 74, False, Play
Sunny, 64, 96, True, Don't Play
Sunny, 69, 74, True, Don't Play
Overcast, 75, 74, True, Play
Overcast, 67, 95, True, Don't Play
Rain, 83, 96, True, Don't Play
Rain, 75, 68, False, Play
Rain, 70, 96, False, Don't Play
Sunny, 68, 74, False, Play
Sunny, 81, 82, False, Play
Sunny, 83, 79, True, Don't Play
Rain, 69, 74, True, Don't Play
Rain, 70, 68, False, Play
Overcast, 75, 75, False, Don't Play
Rain, 70, 90, True, Don't Play
Rain, 76, 68, False, Play
Rain, 69, 95, False, Play
Sunny, 85, 78, True, Play
Sunny, 84, 96, True, Don't Play
Rain, 64, 87, True, Play
Rain, 84, 79, True, Don't Play
Overcast, 83, 87, False, Play
Sunny, 67, 68, False, Play
Sunny, 75, 70, True, Play
Sunny, 84, 79, True, Don't Play
Overcast, 64, 95, True, Don't Play
Rain, 83, 82, False, Play
Rain, 85, 87, False, Play
Sunny, 64, 78, False, Play
Sunny, 67, 74, False, Play
Sunny, 64, 70, True, Don't Play
Overcast, 64, 95, False, Play
Overcast, 64, 87, False, Play
Rain, 64, 95, False, Don't Play
Rain, 69, 82, False, Play
Rain, 69, 90, True, Don't Play
Sunny, 64, 95, False, Play
Sunny, 67, 96, True, Play
Sunny, 85, 96, True, Don't Play
Sunny, 76, 68, False, Don't Play
Rain, 70, 87, False, Don't Play
Overcast, 72, 80, False, Don't Play
Overcast, 72, 83, True, Play

Overcast, 85, 78, True, Don't Play
Rain, 72, 90, True, Don't Play
Rain, 72, 90, True, Don't Play
Rain, 83, 68, True, Play
Rain, 75, 70, True, Play
Sunny, 70, 96, False, Play
Sunny, 80, 90, True, Don't Play
Sunny, 68, 80, False, Play
Overcast, 71, 79, False, Play
Overcast, 81, 80, False, Play
Overcast, 76, 68, True, Don't Play
Rain, 69, 83, True, Don't Play
Rain, 68, 95, False, Don't Play
Sunny, 69, 96, False, Play
Sunny, 70, 80, True, Don't Play

Rain, 75, 70, True, Don't Play
Rain, 69, 74, False, Play
Rain, 69, 70, False, Play
Rain, 70, 83, False, Play
Sunny, 81, 78, False, Play
Sunny, 76, 87, False, Don't Play
Sunny, 85, 85, True, Don't Play
Overcast, 72, 95, True, Don't Play
Overcast, 69, 82, False, Play
Overcast, 75, 90, True, Play
Rain, 70, 80, False, Play
Rain, 71, 70, False, Play
Sunny, 75, 68, True, Don't Play
Sunny, 81, 80, False, Play

Unseen Data: (14 record)

| | | |
|---|---|---|
| Sunny, 75, 70, True | Sunny, 80, 90, True | Sunny, 85, 85, False |
| Sunny, 72, 95, False | Sunny, 69, 70, False | Overcast, 72, 90, True |
| Overcast, 83, 78, False | Overcast, 64, 65, True | Overcast, 81, 75, False |
| Rain, 71, 80, True | Rain, 65, 70, True | Rain, 75, 80, False |
| Rain, 68, 80, False | Rain, 70, 96, False | |

## ค.2 ชุดข้อมูลเห็ด

Class Attribute คือ classes: edible = e, poisonous = p

Non Class Attribute:

1. cap-shape: bell = b, conical = c, convex = x, flat = f, knobbed = k, sunken = s
2. cap-surface: fibrous = f, grooves = g , scaly = y, smooth = s
3. cap-color: brown = n, buff = b, cinnamon = c, gray = g, green = r, pink = p,

purple = u, red = e, white = w, yellow = y

4. bruises: bruises = t, no = f
5. odor: almond = a, anise = l, creosote = c, fishy = y, foul = f, musty = m, none = n, pungent = p, spicy = s
6. gill-attachment: attached = a, descending = d, free = f, notched = n
7. gill-spacing: close = c, crowded = w, distant = d
8. gill-size: broad = b, narrow = n
9. gill-color: black = k, brown = n, buff = b, chocolate = h, gray = g, green = r, orange = o, pink = p, purple = u, red = e, white = w, yellow = y
10. stalk-shape: enlarging = e, tapering = t
11. stalk-root: bulbous = b, club = c, cup = u, equal = e, rhizomorphs = z, rooted = r
12. stalk-surface-above-ring: ibrous = f, scaly = y, silky = k, smooth = s
13. stalk-surface-below-ring: ibrous = f, scaly = y, silky = k, smooth = s
14. stalk-color-above-ring: brown = n, buff = b, cinnamon = c, gray = g, orange = o, pink = p, red = e, white = w, yellow = y
15. stalk-color-below-ring: brown = n, buff = b, cinnamon = c, gray = g, orange = o, pink = p, red = e, white = w, yellow = y
16. veil-type: partial = p, universal = u
17. veil-color: brown = n, orange = o, white = w, yellow = y
18. ring-number: none = n, one = o, two = t
19. ring-type: cobwebby = c, evanescent = e, flaring = f, large = l, none = n, pendant = p, sheathing = s, zone = z
20. spore-print-color: black = k, brown = n, buff = b, chocolate = h, green = r, orange = o, purple = u, white = w, yellow = y
21. population: abundant = a, clustered = c, numerous = n, scattered = s, several = v, solitary = y
22. habitat: grasses = g, leaves = l, meadows = m, paths = p, urban = u, waste = w, woods = d

Relation mushroom:

attribute 'cap-shape' { 'b', 'c', 'f', 'k', 's', 'x'}

attribute 'cap-surface' { 'f', 'g', 's', 'y'}

attribute 'cap-color' { 'b', 'c', 'e', 'g', 'n', 'p', 'r', 'u', 'w', 'y'}

attribute 'bruises' { 'f', 't'}

attribute 'odor' { 'a', 'c', 'f', 'l', 'm', 'n', 'p', 's', 'y'}

attribute 'gill-attachment' { 'a', 'd', 'f', 'n'}

attribute 'gill-spacing' { 'c', 'd', 'w'}

attribute 'gill-size' { 'b', 'n'}

attribute 'gill-color' { 'b', 'e', 'g', 'h', 'k', 'n', 'o', 'p', 'r', 'u', 'w', 'y'}

attribute 'stalk-shape' { 'e', 't'}

attribute 'stalk-root' { 'b', 'c', 'e', 'r', 'u', 'z'}

attribute 'stalk-surface-above-ring' { 'f', 'k', 's', 'y'}

attribute 'stalk-surface-below-ring' { 'f', 'k', 's', 'y'}

attribute 'stalk-color-above-ring' { 'b', 'c', 'e', 'g', 'n', 'o', 'p', 'w', 'y'}

attribute 'stalk-color-below-ring' { 'b', 'c', 'e', 'g', 'n', 'o', 'p', 'w', 'y'}

attribute 'veil-type' { 'p', 'u'}

attribute 'veil-color' { 'n', 'o', 'w', 'y'}

attribute 'ring-number' { 'n', 'o', 't'}

attribute 'ring-type' { 'c', 'e', 'f', 'l', 'n', 'p', 's', 'z'}

attribute 'spore-print-color' { 'b', 'h', 'k', 'n', 'o', 'r', 'u', 'w', 'y'}

attribute 'population' { 'a', 'c', 'n', 's', 'v', 'y'}

attribute 'habitat' { 'd', 'g', 'l', 'm', 'p', 'u', 'w'}

attribute 'class' { 'e', 'p'}

Training Data: (401 record)

'x','s','n','t','p','f','c','n','k','e','e','s','s','w','w','p','w','o','p','k','s','u','p'

'x','s','y','t','a','f','c','b','k','e','c','s','s','w','w','p','w','o','p','n','n','g','e'

'b','s','w','t','l','f','c','b','n','e','c','s','s','w','w','p','w','o','p','n','n','m','e'

'x','y','w','t','p','f','c','n','n','e','e','s','s','w','w','p','w','o','p','k','s','u','p'

'x','s','g','f','n','f','w','b','k','t','e','s','s','w','w','p','w','o','e','n','a','g','e'

'x','y','y','t','a','f','c','b','n','e','c','s','s','w','w','p','w','o','p','k','n','g','e'

'b','s','w','t','a','f','c','b','g','e','c','s','s','w','w','p','w','o','p','k','n','m','e'

'b','y','w','t','l','f','c','b','n','e','c','s','s','w','w','p','w','o','p','n','s','m','e'

'x','y','w','t','p','f','c','n','p','e','e','s','s','w','w','p','w','o','p','k','v','g','p'

'b','s','y','t','a','f','c','b','g','e','c','s','s','w','w','p','w','o','p','k','s','m','e'

'x','y','y','t','l','f','c','b','g','e','c','s','s','w','w','p','w','o','p','n','n','g','e'

'b','s','y','t','a','f','c','b','w','e','c','s','s','w','w','p','w','o','p','n','s','g','e'

'x','y','w','t','p','f','c','n','k','e','e','s','s','w','w','p','w','o','p','n','v','u','p'

'x','f','n','f','n','f','w','b','n','t','e','s','f','w','w','p','w','o','e','k','a','g','e'

's','f','g','f','n','f','c','n','k','e','e','s','s','w','w','p','w','o','p','n','y','u','e'

'f','f','w','f','n','f','w','b','k','t','e','s','s','w','w','p','w','o','e','n','a','g','e'

'x','s','n','t','p','f','c','n','n','e','e','s','s','w','w','p','w','o','p','k','s','g','p'

'x','y','w','t','p','f','c','n','n','e','e','s','s','w','w','p','w','o','p','n','s','u','p'

'x','s','n','t','p','f','c','n','k','e','e','s','s','w','w','p','w','o','p','n','s','u','p'

'b','s','y','t','a','f','c','b','k','e','c','s','s','w','w','p','w','o','p','n','s','m','e'

'x','y','n','t','p','f','c','n','n','e','e','s','s','w','w','p','w','o','p','n','v','g','p'

'b','y','y','t','l','f','c','b','k','e','c','s','s','w','w','p','w','o','p','n','s','m','e'

'b','y','w','t','a','f','c','b','w','e','c','s','s','w','w','p','w','o','p','n','n','m','e'

'b','s','w','t','l','f','c','b','g','e','c','s','s','w','w','p','w','o','p','k','s','m','e'

'f','s','w','t','p','f','c','n','n','e','e','s','s','w','w','p','w','o','p','n','v','g','p'

'x','y','y','t','a','f','c','b','n','e','c','s','s','w','w','p','w','o','p','n','n','m','e'

'x','y','w','t','l','f','c','b','w','e','c','s','s','w','w','p','w','o','p','n','n','m','e'

'f','f','n','f','n','f','c','n','k','e','e','s','s','w','w','p','w','o','p','k','y','u','e'

'x','s','y','t','a','f','w','n','n','t','b','s','s','w','w','p','w','o','p','n','v','d','e'

'b','s','y','t','l','f','c','b','g','e','c','s','s','w','w','p','w','o','p','n','n','m','e'

'x','y','w','t','p','f','c','n','k','e','e','s','s','w','w','p','w','o','p','n','s','u','p'

'x','y','y','t','l','f','c','b','n','e','c','s','s','w','w','p','w','o','p','n','n','m','e'

'x','y','n','t','l','f','c','b','p','e','r','s','y','w','w','p','w','o','p','n','y','p','e'

'b','y','y','t','l','f','c','b','n','e','c','s','s','w','w','p','w','o','p','n','s','m','e'

'x','f','y','t','l','f','w','n','w','t','b','s','s','w','w','p','w','o','p','n','v','d','e'

's','f','g','f','n','f','c','n','k','e','e','s','s','w','w','p','w','o','p','k','v','u','e'

'x','y','n','t','p','f','c','n','w','e','e','s','s','w','w','p','w','o','p','n','s','u','p'

'x','f','y','t','a','f','w','n','p','t','b','s','s','w','w','p','w','o','p','n','v','d','e'

'b','s','y','t','l','f','c','b','k','e','c','s','s','w','w','p','w','o','p','k','s','m','e'

'b','y','y','t','a','f','c','b','n','e','c','s','s','w','w','p','w','o','p','n','s','g','e'

'x','y','y','t','l','f','c','b','n','e','r','s','y','w','w','p','w','o','p','k','y','p','e'

'x','f','n','f','n','f','c','n','g','e','e','s','s','w','w','p','w','o','p','k','y','u','e'

'x','y','w','t','p','f','c','n','p','e','e','s','s','w','w','p','w','o','p','n','v','g','p'

'x','s','y','t','a','f','c','b','w','e','c','s','s','w','w','p','w','o','p','k','n','m','e'

'x','y','w','t','a','f','c','b','n','e','c','s','s','w','w','p','w','o','p','n','n','g','e'

'x','y','y','t','l','f','c','b','k','e','c','s','s','w','w','p','w','o','p','k','s','m','e'

'x','s','w','t','l','f','c','b','w','e','c','s','s','w','w','p','w','o','p','n','n','m','e'

'x','y','y','t','l','f','c','b','n','e','r','s','y','w','w','p','w','o','p','n','s','p','e'

'f','y','y','t','l','f','c','b','w','e','r','s','y','w','w','p','w','o','p','k','s','p','e'

'x','y','n','t','a','f','c','b','w','e','r','s','y','w','w','p','w','o','p','k','s','g','e'

'x','s','w','t','l','f','c','b','k','e','c','s','s','w','w','p','w','o','p','k','s','g','e'

'b','s','w','t','l','f','c','b','k','e','c','s','s','w','w','p','w','o','p','n','n','m','e'

'x','y','n','t','p','f','c','n','k','e','e','s','s','w','w','p','w','o','p','n','v','u','p'

'x','s','w','t','p','f','c','n','k','e','e','s','s','w','w','p','w','o','p','k','v','u','p'

'b','y','y','t','a','f','c','b','w','e','c','s','s','w','w','p','w','o','p','k','s','m','e'

'f','f','g','f','n','f','w','b','n','t','e','s','s','w','w','p','w','o','e','n','a','g','e'

'b','s','w','t','a','f','c','b','w','e','c','s','s','w','w','p','w','o','p','n','n','g','e'

'x','s','y','t','l','f','c','b','k','e','c','s','s','w','w','p','w','o','p','k','n','g','e'

'x','y','n','t','a','f','c','b','p','e','r','s','y','w','w','p','w','o','p','k','y','p','e'

's','f','g','f','n','f','c','n','k','e','e','s','s','w','w','p','w','o','p','n','v','u','e'

'b','y','y','t','a','f','c','b','k','e','c','s','s','w','w','p','w','o','p','n','s','m','e'

'b','s','y','t','l','f','c','b','g','e','c','s','s','w','w','p','w','o','p','n','s','m','e'

'b','y','y','t','l','f','c','b','g','e','c','s','s','w','w','p','w','o','p','n','n','m','e'

'b','y','w','t','l','f','c','b','n','e','c','s','s','w','w','p','w','o','p','n','s','g','e'

'f','s','n','f','n','f','w','b','k','t','e','s','s','w','w','p','w','o','e','k','a','g','e'

'x','s','w','t','l','f','c','b','n','e','c','s','s','w','w','p','w','o','p','k','s','g','e'

'f','y','y','t','a','f','c','b','w','e','r','s','y','w','w','p','w','o','p','n','s','g','e'

'x','y','y','t','a','f','c','b','w','e','c','s','s','w','w','p','w','o','p','k','n','g','e'

'x','f','g','f','n','f','c','n','p','e','e','s','s','w','w','p','w','o','p','n','v','u','e'

'f','f','y','t','l','f','w','n','p','t','b','s','s','w','w','p','w','o','p','n','v','d','e'

'b','y','w','t','l','f','c','b','g','e','c','s','s','w','w','p','w','o','p','n','s','m','e'

'f','f','y','t','l','f','w','n','w','t','b','s','s','w','w','p','w','o','p','n','v','d','e'

'x','y','n','t','a','f','c','b','p','e','r','s','y','w','w','p','w','o','p','k','s','p','e'

'b','s','y','t','a','f','c','b','k','e','c','s','s','w','w','p','w','o','p','k','s','g','e'

'f','s','y','t','l','f','w','n','p','t','b','s','s','w','w','p','w','o','p','n','v','d','e'

'x','s','w','t','l','f','w','n','n','t','b','s','s','w','w','p','w','o','p','u','v','d','e'

'f','y','n','t','l','f','c','b','p','e','r','s','y','w','w','p','w','o','p','n','y','p','e'

'x','y','n','t','p','f','c','n','w','e','e','s','s','w','w','p','w','o','p','n','v','u','p'

'f','y','n','t','a','f','c','b','n','e','r','s','y','w','w','p','w','o','p','n','y','g','e'

'x','s','n','f','n','f','w','b','k','t','e','f','s','w','w','p','w','o','e','n','s','g','e'

'f','f','g','f','n','f','c','n','n','e','e','s','s','w','w','p','w','o','p','n','y','u','e'

'x','f','g','f','n','f','w','b','n','t','e','s','s','w','w','p','w','o','e','n','s','g','e'

'x','y','y','t','l','f','c','b','w','e','r','s','y','w','w','p','w','o','p','k','s','g','e'

'x','s','n','f','n','f','w','b','k','t','e','s','s','w','w','p','w','o','e','k','s','g','e'

'b','s','w','t','a','f','c','b','w','e','c','s','s','w','w','p','w','o','p','k','s','g','e'

'x','s','w','t','l','f','c','b','n','e','c','s','s','w','w','p','w','o','p','n','s','g','e'

'f','y','n','t','l','f','c','b','w','e','r','s','y','w','w','p','w','o','p','k','y','g','e'

's','f','n','f','n','f','c','n','n','e','e','s','s','w','w','p','w','o','p','n','v','u','e'

'x','f','n','f','n','f','c','n','n','e','e','s','s','w','w','p','w','o','p','n','y','u','e'

'b','s','w','t','l','f','c','b','k','e','c','s','s','w','w','p','w','o','p','k','s','g','e'

'x','y','y','t','a','f','c','b','g','e','c','s','s','w','w','p','w','o','p','k','s','g','e'

'x','y','y','t','l','f','c','b','g','e','c','s','s','w','w','p','w','o','p','k','n','m','e'

'x','s','n','f','n','f','w','b','n','t','e','s','s','w','w','p','w','o','e','n','a','g','e'

'x','s','w','t','a','f','c','b','g','e','c','s','s','w','w','p','w','o','p','n','s','g','e'

'f','y','n','t','l','f','c','b','p','e','r','s','y','w','w','p','w','o','p','n','s','g','e'

'x','s','y','t','a','f','c','b','n','e','c','s','s','w','w','p','w','o','p','k','n','g','e'

'b','s','w','t','a','f','c','b','g','e','c','s','s','w','w','p','w','o','p','n','s','g','e'

'x','y','w','t','a','f','c','b','g','e','c','s','s','w','w','p','w','o','p','k','s','g','e'

'x','f','n','f','n','f','w','b','p','t','e','f','s','w','w','p','w','o','e','k','s','g','e'

'b','s','y','t','l','f','c','b','n','e','c','s','s','w','w','p','w','o','p','k','n','g','e'

'f','y','y','t','l','f','c','b','w','e','r','s','y','w','w','p','w','o','p','n','s','g','e'

'x','y','y','t','a','f','c','b','n','e','r','s','y','w','w','p','w','o','p','k','y','p','e'

'b','y','w','t','l','f','c','b','g','e','c','s','s','w','w','p','w','o','p','n','n','g','e'

'x','y','y','t','a','f','c','b','n','e','c','s','s','w','w','p','w','o','p','k','n','m','e'

'x','y','y','t','a','f','c','b','w','e','r','s','y','w','w','p','w','o','p','n','y','g','e'

'b','y','w','t','l','f','c','b','n','e','c','s','s','w','w','p','w','o','p','k','s','m','e'

'b','y','w','t','a','f','c','b','g','e','c','s','s','w','w','p','w','o','p','n','s','m','e'

'x','s','y','t','a','f','c','b','k','e','c','s','s','w','w','p','w','o','p','k','n','m','e'

'x','s','y','t','l','f','c','b','w','e','c','s','s','w','w','p','w','o','p','k','n','g','e'

's','f','g','f','n','f','c','n','g','e','e','s','s','w','w','p','w','o','p','k','y','u','e'

'x','f','w','t','a','f','w','n','w','t','b','s','s','w','w','p','w','o','p','u','v','d','e'

'x','s','y','t','a','f','c','b','n','e','c','s','s','w','w','p','w','o','p','k','n','m','e'

'x','y','w','t','p','f','c','n','n','e','e','s','s','w','w','p','w','o','p','n','v','u','p'

'x','y','y','t','l','f','c','b','p','e','r','s','y','w','w','p','w','o','p','n','s','g','e'

's','f','g','f','n','f','c','n','p','e','e','s','s','w','w','p','w','o','p','n','y','u','e'

'x','y','y','t','l','f','c','b','w','e','r','s','y','w','w','p','w','o','p','k','y','g','e'

'x','s','y','t','l','f','w','n','p','t','b','s','s','w','w','p','w','o','p','u','v','d','e'

's','f','n','f','n','f','c','n','k','e','e','s','s','w','w','p','w','o','p','n','y','u','e'

'x','s','w','t','p','f','c','n','k','e','e','s','s','w','w','p','w','o','p','k','v','g','p'

'x','y','w','t','a','f','c','b','g','e','c','s','s','w','w','p','w','o','p','n','n','m','e'

'f','y','n','t','p','f','c','n','p','e','e','s','s','w','w','p','w','o','p','k','v','g','p'

'f','s','g','f','n','f','w','b','k','t','e','s','s','w','w','p','w','o','e','n','a','g','e'

'x','s','y','t','l','f','c','b','g','e','c','s','s','w','w','p','w','o','p','n','s','m','e'

'x','s','w','f','n','f','w','b','n','t','e','s','f','w','w','p','w','o','e','k','s','g','e'

'b','s','y','t','a','f','c','b','w','e','c','s','s','w','w','p','w','o','p','n','n','g','e'

'f','f','g','f','n','f','w','b','h','t','e','s','s','w','w','p','w','o','e','n','a','g','e'

'x','s','w','t','l','f','c','b','n','e','c','s','s','w','w','p','w','o','p','k','n','g','e'

'b','s','w','t','l','f','c','b','n','e','c','s','s','w','w','p','w','o','p','n','s','m','e'

'b','s','w','t','l','f','c','b','w','e','c','s','s','w','w','p','w','o','p','n','s','g','e'

'b','y','w','t','l','f','c','b','w','e','c','s','s','w','w','p','w','o','p','n','s','m','e'

'f','s','w','t','l','f','w','n','w','t','b','s','s','w','w','p','w','o','p','u','v','d','e'

'x','y','y','t','l','f','c','b','g','e','c','s','s','w','w','p','w','o','p','k','s','m','e'

'f','s','w','t','a','f','w','n','p','t','b','s','s','w','w','p','w','o','p','n','v','d','e'

'x','y','w','t','p','f','c','n','w','e','e','s','s','w','w','p','w','o','p','n','v','u','p'

'f','f','w','t','l','f','w','n','w','t','b','s','s','w','w','p','w','o','p','n','v','d','e'

'x','y','y','t','a','f','c','b','n','e','c','s','s','w','w','p','w','o','p','n','s','g','e'

'x','s','n','t','p','f','c','n','p','e','e','s','s','w','w','p','w','o','p','n','v','g','p'

'b','s','y','t','l','f','c','b','w','e','c','s','s','w','w','p','w','o','p','n','n','g','e'

'x','y','n','t','a','f','c','b','w','e','r','s','y','w','w','p','w','o','p','k','y','p','e'

'b','y','y','t','l','f','c','b','g','e','c','s','s','w','w','p','w','o','p','k','n','m','e'

's','f','n','f','n','f','c','n','k','e','e','s','s','w','w','p','w','o','p','n','v','u','e'

'f','y','n','t','a','f','c','b','w','e','r','s','y','w','w','p','w','o','p','k','y','p','e'

'x','y','y','t','a','f','c','b','k','e','c','s','s','w','w','p','w','o','p','k','n','g','e'

'x','f','g','f','n','f','w','b','k','t','e','f','f','w','w','p','w','o','e','k','s','g','e'

'f','f','w','f','n','f','w','b','k','t','e','s','f','w','w','p','w','o','e','n','a','g','e'

'x','y','y','t','l','f','c','b','w','e','c','s','s','w','w','p','w','o','p','n','n','m','e'

'b','s','y','t','l','f','c','b','k','e','c','s','s','w','w','p','w','o','p','k','n','g','e'

'b','y','w','t','a','f','c','b','g','e','c','s','s','w','w','p','w','o','p','k','n','m','e'

'x','y','w','t','a','f','c','b','w','e','c','s','s','w','w','p','w','o','p','n','s','g','e'

'x','s','n','f','n','f','w','b','p','t','e','f','s','w','w','p','w','o','e','n','a','g','e'

'x','y','w','t','l','f','c','b','g','e','c','s','s','w','w','p','w','o','p','n','s','g','e'

's','f','n','f','n','f','c','n','k','e','e','s','s','w','w','p','w','o','p','k','v','u','e'

'x','s','w','t','a','f','w','n','w','t','b','s','s','w','w','p','w','o','p','u','v','d','e'

'x','y','n','t','l','f','c','b','w','e','r','s','y','w','w','p','w','o','p','k','s','g','e'

'b','y','y','t','a','f','c','b','k','e','c','s','s','w','w','p','w','o','p','n','n','g','e'

'x','y','w','t','a','f','c','b','k','e','c','s','s','w','w','p','w','o','p','n','n','g','e'

'b','y','w','t','a','f','c','b','n','e','c','s','s','w','w','p','w','o','p','k','s','m','e'

'b','s','y','t','a','f','c','b','g','e','c','s','s','w','w','p','w','o','p','k','s','g','e'

'b','s','y','t','l','f','c','b','g','e','c','s','s','w','w','p','w','o','p','k','s','m','e'

'b','y','y','t','a','f','c','b','n','e','c','s','s','w','w','p','w','o','p','k','n','g','e'

'x','f','n','f','n','f','c','n','k','e','e','s','s','w','w','p','w','o','p','n','y','u','e'

'f','y','n','t','l','f','c','b','n','e','r','s','y','w','w','p','w','o','p','n','y','g','e'

'x','y','w','t','a','f','c','b','k','e','c','s','s','w','w','p','w','o','p','n','s','g','e'

'f','y','y','t','l','f','c','b','w','e','r','s','y','w','w','p','w','o','p','n','y','p','e'

'b','s','w','t','a','f','c','b','w','e','c','s','s','w','w','p','w','o','p','n','s','g','e'

'b','s','w','t','a','f','c','b','w','e','c','s','s','w','w','p','w','o','p','k','s','m','e'

'x','y','n','t','l','f','c','b','w','e','r','s','y','w','w','p','w','o','p','k','y','g','e'

'b','s','w','t','a','f','c','b','k','e','c','s','s','w','w','p','w','o','p','k','s','g','e'

'x','f','g','f','n','f','c','n','g','e','e','s','s','w','w','p','w','o','p','n','y','u','e'

'b','s','y','t','l','f','c','b','k','e','c','s','s','w','w','p','w','o','p','n','s','g','e'

'x','f','y','t','l','f','w','n','n','t','b','s','s','w','w','p','w','o','p','u','v','d','e'

'b','y','y','t','a','f','c','b','w','e','c','s','s','w','w','p','w','o','p','k','s','g','e'

'f','y','y','t','l','f','c','b','p','e','r','s','y','w','w','p','w','o','p','n','s','g','e'

'b','y','w','t','l','f','c','b','w','e','c','s','s','w','w','p','w','o','p','k','n','m','e'

'b','y','w','t','a','f','c','b','k','e','c','s','s','w','w','p','w','o','p','k','n','m','e'

'b','y','y','t','a','f','c','b','g','e','c','s','s','w','w','p','w','o','p','n','s','g','e'

'x','y','y','t','l','f','c','b','g','e','c','s','s','w','w','p','w','o','p','n','n','m','e'

'b','s','y','t','l','f','c','b','g','e','c','s','s','w','w','p','w','o','p','n','n','g','e'

'x','y','w','t','p','f','c','n','p','e','e','s','s','w','w','p','w','o','p','n','v','u','p'

's','f','n','f','n','f','c','n','g','e','e','s','s','w','w','p','w','o','p','n','y','u','e'

'f','f','n','f','n','f','c','n','g','e','e','s','s','w','w','p','w','o','p','k','v','u','e'

'x','s','y','t','a','f','c','b','g','e','c','s','s','w','w','p','w','o','p','n','s','m','e'

'f','y','n','t','a','f','c','b','p','e','r','s','y','w','w','p','w','o','p','k','s','p','e'

'x','y','w','t','p','f','c','n','k','e','e','s','s','w','w','p','w','o','p','k','s','g','p'

'b','s','w','t','l','f','c','b','g','e','c','s','s','w','w','p','w','o','p','n','s','m','e'

'f','f','g','f','n','f','c','n','p','e','e','s','s','w','w','p','w','o','p','k','v','u','e'

'b','y','y','t','l','f','c','b','g','e','c','s','s','w','w','p','w','o','p','n','n','g','e'

'x','y','n','t','a','f','c','b','w','e','r','s','y','w','w','p','w','o','p','n','y','p','e'

'x','f','w','f','n','f','w','b','p','t','e','s','f','w','w','p','w','o','e','k','s','g','e'

'x','s','w','t','l','f','w','n','w','t','b','s','s','w','w','p','w','o','p','n','v','d','e'

'b','s','w','t','l','f','c','b','w','e','c','s','s','w','w','p','w','o','p','n','s','m','e'

'f','s','y','t','a','f','w','n','w','t','b','s','s','w','w','p','w','o','p','n','v','d','e'

'x','s','y','t','a','f','c','b','k','e','c','s','s','w','w','p','w','o','p','n','n','m','e'

'f','f','g','f','n','f','c','n','g','e','e','s','s','w','w','p','w','o','p','n','y','u','e'

'b','s','y','t','a','f','c','b','n','e','c','s','s','w','w','p','w','o','p','k','s','g','e'

'x','s','w','t','l','f','c','b','g','e','c','s','s','w','w','p','w','o','p','n','n','m','e'

'x','y','w','t','l','f','c','b','n','e','c','s','s','w','w','p','w','o','p','n','s','m','e'

'f','s','w','t','a','f','w','n','n','t','b','s','s','w','w','p','w','o','p','n','v','d','e'

'x','y','y','t','l','f','c','b','w','e','c','s','s','w','w','p','w','o','p','n','s','g','e'

'b','s','w','t','l','f','c','b','w','e','c','s','s','w','w','p','w','o','p','k','s','g','e'

'x','s','w','t','a','f','c','b','g','e','c','s','s','w','w','p','w','o','p','k','n','g','e'

'x','f','w','f','n','f','w','b','h','t','e','f','s','w','w','p','w','o','e','k','s','g','e'

'f','y','n','t','l','f','c','b','n','e','r','s','y','w','w','p','w','o','p','k','y','p','e'

'x','s','w','t','p','f','c','n','k','e','e','s','s','w','w','p','w','o','p','n','v','u','p'

'b','s','w','t','a','f','c','b','n','e','c','s','s','w','w','p','w','o','p','n','n','g','e'

'b','s','w','t','a','f','c','b','k','e','c','s','s','w','w','p','w','o','p','k','s','m','e'

'b','y','w','t','l','f','c','b','n','e','c','s','s','w','w','p','w','o','p','n','n','g','e'

'b','y','w','t','a','f','c','b','w','e','c','s','s','w','w','p','w','o','p','k','n','g','e'

'x','s','y','t','a','f','c','b','w','e','c','s','s','w','w','p','w','o','p','k','s','g','e'

'b','s','w','t','l','f','c','b','w','e','c','s','s','w','w','p','w','o','p','k','n','m','e'

'x','f','y','t','a','f','w','n','n','t','b','s','s','w','w','p','w','o','p','n','v','d','e'

'x','f','g','f','n','f','c','n','n','e','e','s','s','w','w','p','w','o','p','n','y','u','e'

'f','y','y','t','a','f','c','b','n','e','r','s','y','w','w','p','w','o','p','n','s','g','e'

'b','s','w','t','l','f','c','b','n','e','c','s','s','w','w','p','w','o','p','k','n','m','e'

'x','s','y','t','a','f','c','b','g','e','c','s','s','w','w','p','w','o','p','k','s','g','e'

'x','y','y','t','a','f','c','b','k','e','c','s','s','w','w','p','w','o','p','k','s','g','e'

'x','y','w','t','l','f','c','b','w','e','c','s','s','w','w','p','w','o','p','n','s','g','e'

's','f','g','f','n','f','c','n','p','e','e','s','s','w','w','p','w','o','p','n','v','u','e'

'x','s','w','t','a','f','c','b','n','e','c','s','s','w','w','p','w','o','p','n','n','m','e'

'x','s','w','t','p','f','c','n','k','e','e','s','s','w','w','p','w','o','p','k','s','g','p'

'x','y','y','t','a','f','c','b','n','e','r','s','y','w','w','p','w','o','p','n','s','g','e'

'f','f','w','t','a','f','w','n','p','t','b','s','s','w','w','p','w','o','p','n','v','d','e'

'x','y','w','t','l','f','c','b','g','e','c','s','s','w','w','p','w','o','p','k','n','m','e'

'b','y','w','t','l','f','c','b','n','e','c','s','s','w','w','p','w','o','p','k','s','g','e'

'x','s','w','t','a','f','c','b','g','e','c','s','s','w','w','p','w','o','p','n','n','g','e'

'x','s','y','t','a','f','c','b','g','e','c','s','s','w','w','p','w','o','p','n','s','g','e'

'x','y','n','t','p','f','c','n','p','e','e','s','s','w','w','p','w','o','p','k','v','u','p'

'b','s','y','t','a','f','c','b','n','e','c','s','s','w','w','p','w','o','p','n','n','g','e'

'x','f','g','f','n','f','c','n','n','e','e','s','s','w','w','p','w','o','p','k','y','u','e'

'x','y','w','t','p','f','c','n','k','e','e','s','s','w','w','p','w','o','p','k','s','u','p'

'x','y','y','t','l','f','c','b','n','e','r','s','y','w','w','p','w','o','p','n','y','p','e'

'f','f','n','f','n','f','c','n','k','e','e','s','s','w','w','p','w','o','p','k','v','u','e'

'b','s','w','t','l','f','c','b','k','e','c','s','s','w','w','p','w','o','p','n','n','g','e'

'x','f','w','t','l','f','w','n','w','t','b','s','s','w','w','p','w','o','p','n','v','d','e'

'x','y','w','t','l','f','c','b','w','e','c','s','s','w','w','p','w','o','p','k','s','g','e'

'b','y','y','t','l','f','c','b','n','e','c','s','s','w','w','p','w','o','p','n','s','g','e'

'x','y','y','t','l','f','c','b','n','e','r','s','y','w','w','p','w','o','p','k','s','g','e'

'f','y','y','t','a','f','c','b','p','e','r','s','y','w','w','p','w','o','p','k','s','p','e'

'f','y','y','t','a','f','c','b','w','e','r','s','y','w','w','p','w','o','p','k','y','p','e'

'x','s','w','t','l','f','c','b','w','e','c','s','s','w','w','p','w','o','p','k','s','g','e'

'x','s','w','t','a','f','c','b','w','e','c','s','s','w','w','p','w','o','p','n','n','m','e'

'x','s','w','t','p','f','c','n','n','e','e','s','s','w','w','p','w','o','p','k','v','u','p'

'f','f','w','t','a','f','w','n','p','t','b','s','s','w','w','p','w','o','p','u','v','d','e'

'x','s','w','t','l','f','c','b','w','e','c','s','s','w','w','p','w','o','p','n','s','g','e'

'x','s','w','t','l','f','w','n','p','t','b','s','s','w','w','p','w','o','p','u','v','d','e'

'x','y','w','t','a','f','c','b','n','e','c','s','s','w','w','p','w','o','p','k','s','m','e'

'f','y','y','t','l','f','c','b','w','e','r','s','y','w','w','p','w','o','p','k','y','p','e'

'x','s','n','f','n','f','w','b','p','t','e','f','s','w','w','p','w','o','e','k','s','g','e'

'f','y','y','t','a','f','c','b','w','e','r','s','y','w','w','p','w','o','p','n','y','g','e'

'x','s','n','t','p','f','c','n','p','e','e','s','s','w','w','p','w','o','p','n','s','g','p'

's','f','n','f','n','f','c','n','g','e','e','s','s','w','w','p','w','o','p','n','v','u','e'

'b','y','y','t','l','f','c','b','w','e','c','s','s','w','w','p','w','o','p','n','s','m','e'

'b','s','w','t','l','f','c','b','n','e','c','s','s','w','w','p','w','o','p','n','s','g','e'

'b','y','w','t','l','f','c','b','k','e','c','s','s','w','w','p','w','o','p','n','n','m','e'

'f','f','n','f','n','f','c','n','k','e','e','s','s','w','w','p','w','o','p','n','v','u','e'

'x','s','w','t','a','f','c','b','w','e','c','s','s','w','w','p','w','o','p','k','n','m','e'

'b','y','w','t','l','f','c','b','g','e','c','s','s','w','w','p','w','o','p','k','n','g','e'

'b','y','w','t','a','f','c','b','g','e','c','s','s','w','w','p','w','o','p','n','n','m','e'

'f','y','n','t','l','f','c','b','w','e','r','s','y','w','w','p','w','o','p','n','s','g','e'

'x','y','w','t','p','f','c','n','k','e','e','s','s','w','w','p','w','o','p','k','v','g','p'

'x','s','w','t','a','f','c','b','n','e','c','s','s','w','w','p','w','o','p','n','n','g','e'

'x','y','w','t','a','f','c','b','n','e','c','s','s','w','w','p','w','o','p','n','s','m','e'

'f','f','w','t','l','f','w','n','w','t','b','s','s','w','w','p','w','o','p','u','v','d','e'

'f','f','g','f','n','f','c','n','g','e','e','s','s','w','w','p','w','o','p','k','v','u','e'

'f','s','g','f','n','f','w','b','p','t','e','s','s','w','w','p','w','o','e','k','a','g','e'

'x','y','y','t','a','f','c','b','g','e','c','s','s','w','w','p','w','o','p','n','s','m','e'

'b','s','w','t','a','f','c','b','k','e','c','s','s','w','w','p','w','o','p','n','s','m','e'

'f','y','n','t','p','f','c','n','w','e','e','s','s','w','w','p','w','o','p','k','s','u','p'

'x','s','w','t','l','f','c','b','n','e','c','s','s','w','w','p','w','o','p','k','n','m','e'

'x','s','y','t','l','f','c','b','k','e','c','s','s','w','w','p','w','o','p','n','n','m','e'

'x','y','w','t','p','f','c','n','w','e','e','s','s','w','w','p','w','o','p','n','s','g','p'

'f','y','n','t','a','f','c','b','p','e','r','s','y','w','w','p','w','o','p','k','y','g','e'

's','f','n','f','n','f','c','n','p','e','e','s','s','w','w','p','w','o','p','n','v','u','e'

'b','s','y','t','l','f','c','b','w','e','c','s','s','w','w','p','w','o','p','k','n','g','e'

'b','y','w','t','l','f','c','b','n','e','c','s','s','w','w','p','w','o','p','k','n','m','e'

'f','y','n','t','a','f','c','b','n','e','r','s','y','w','w','p','w','o','p','k','s','g','e'

'b','y','y','t','a','f','c','b','g','e','c','s','s','w','w','p','w','o','p','k','s','m','e'

'b','y','w','t','a','f','c','b','g','e','c','s','s','w','w','p','w','o','p','k','s','g','e'

'x','y','n','t','l','f','c','b','n','e','r','s','y','w','w','p','w','o','p','n','y','p','e'

'f','f','g','f','n','f','c','n','p','e','e','s','s','w','w','p','w','o','p','n','y','u','e'

'x','f','g','f','n','f','c','n','g','e','e','s','s','w','w','p','w','o','p','k','y','u','e'

'b','y','y','t','l','f','c','b','n','e','c','s','s','w','w','p','w','o','p','k','s','g','e'

'x','s','y','t','l','f','c','b','n','e','c','s','s','w','w','p','w','o','p','k','s','m','e'

'b','y','w','t','l','f','c','b','k','e','c','s','s','w','w','p','w','o','p','n','s','g','e'

'x','s','w','t','a','f','c','b','g','e','c','s','s','w','w','p','w','o','p','k','s','m','e'

'b','s','y','t','a','f','c','b','g','e','c','s','s','w','w','p','w','o','p','n','n','m','e'

'x','s','y','t','a','f','c','b','k','e','c','s','s','w','w','p','w','o','p','k','s','m','e'

'x','f','g','f','n','f','w','b','p','t','e','f','f','w','w','p','w','o','e','k','s','g','e'

'f','s','y','t','a','f','w','n','p','t','b','s','s','w','w','p','w','o','p','n','v','d','e'

'x','y','w','t','p','f','c','n','p','e','e','s','s','w','w','p','w','o','p','k','s','g','p'

'x','f','w','f','n','f','w','b','k','t','e','f','s','w','w','p','w','o','e','k','a','g','e'

'b','y','w','t','a','f','c','b','w','e','c','s','s','w','w','p','w','o','p','k','s','g','e'

'b','y','y','t','a','f','c','b','n','e','c','s','s','w','w','p','w','o','p','n','n','m','e'

'x','s','y','t','l','f','c','b','k','e','c','s','s','w','w','p','w','o','p','k','s','m','e'

'b','y','w','t','l','f','c','b','n','e','c','s','s','w','w','p','w','o','p','k','n','m','e'

'x','s','y','t','l','f','w','n','w','t','b','s','s','w','w','p','w','o','p','n','v','d','e'

'b','y','w','t','a','f','c','b','n','e','c','s','s','w','w','p','w','o','p','k','s','g','e'

'x','y','w','t','l','f','c','b','g','e','c','s','s','w','w','p','w','o','p','k','s','g','e'

'x','f','n','t','n','f','c','b','p','t','b','s','s','g','p','p','w','o','p','n','y','d','e'

'b','y','w','t','l','f','c','b','k','e','c','s','s','w','w','p','w','o','p','n','s','m','e'

'x','s','y','t','l','f','c','b','g','e','c','s','s','w','w','p','w','o','p','n','n','m','e'

'f','y','n','t','a','f','c','b','p','e','r','s','y','w','w','p','w','o','p','k','y','g','e'

'x','f','w','f','n','f','w','b','n','t','e','s','s','w','w','p','w','o','e','n','s','g','e'

'x','s','y','t','l','f','c','b','w','e','c','s','s','w','w','p','w','o','p','k','s','g','e'

'x','y','w','t','p','f','c','n','n','e','e','s','s','w','w','p','w','o','p','n','v','g','p'

'x','y','y','t','l','f','c','b','n','e','c','s','s','w','w','p','w','o','p','k','s','g','e'

'x','f','w','t','a','f','w','n','p','t','b','s','s','w','w','p','w','o','p','n','v','d','e'

'b','s','y','t','a','f','c','b','g','e','c','s','s','w','w','p','w','o','p','n','n','g','e'

'x','y','w','t','p','f','c','n','k','e','e','s','s','w','w','p','w','o','p','k','v','u','p'

'x','s','y','t','a','f','w','n','w','t','b','s','s','w','w','p','w','o','p','n','v','d','e'

'x','y','w','t','a','f','c','b','k','e','c','s','s','w','w','p','w','o','p','k','n','m','e'

'x','f','w','t','l','f','w','n','n','t','b','s','s','w','w','p','w','o','p','n','v','d','e'

'f','s','w','t','l','f','w','n','w','t','b','s','s','w','w','p','w','o','p','n','v','d','e'

'x','y','y','t','l','f','c','b','k','e','c','s','s','w','w','p','w','o','p','n','s','m','e'

'f','f','y','t','a','f','w','n','w','t','b','s','s','w','w','p','w','o','p','u','v','d','e'

'x','s','w','t','a','f','c','b','w','e','c','s','s','w','w','p','w','o','p','k','s','g','e'

'b','y','y','t','a','f','c','b','n','e','c','s','s','w','w','p','w','o','p','k','s','m','e'

'x','y','n','t','l','f','c','b','p','e','r','s','y','w','w','p','w','o','p','n','s','p','e'

'b','y','w','t','l','f','c','b','k','e','c','s','s','w','w','p','w','o','p','k','n','g','e'

'x','s','y','t','a','f','c','b','n','e','c','s','s','w','w','p','w','o','p','n','s','m','e'

'x','y','n','t','p','f','c','n','n','e','e','s','s','w','w','p','w','o','p','n','v','u','p'

'b','s','y','t','l','f','c','b','w','e','c','s','s','w','w','p','w','o','p','k','n','g','e'

'b','y','w','t','a','f','c','b','g','e','c','s','s','w','w','p','w','o','p','k','n','g','e'

'x','y','n','t','p','f','c','n','w','e','e','s','s','w','w','p','w','o','p','k','v','u','p'

'b','s','w','t','l','f','c','b','n','e','c','s','s','w','w','p','w','o','p','k','s','m','e'

'b','y','y','t','l','f','c','b','w','e','c','s','s','w','w','p','w','o','p','n','n','m','e'

'b','y','y','t','a','f','c','b','n','e','c','s','s','w','w','p','w','o','p','k','s','g','e'

'x','y','w','t','l','f','c','b','g','e','c','s','s','w','w','p','w','o','p','n','n','g','e'

'x','f','n','t','n','f','c','b','p','t','b','s','s','p','w','p','w','o','p','k','y','d','e'

'x','y','n','t','a','f','c','b','w','e','r','s','y','w','w','p','w','o','p','k','y','g','e'

'f','s','n','t','p','f','c','n','w','e','e','s','s','w','w','p','w','o','p','k','v','u','p'

'x','y','w','t','l','f','c','b','k','e','c','s','s','w','w','p','w','o','p','n','s','g','e'

'x','y','y','t','a','f','c','b','k','e','c','s','s','w','w','p','w','o','p','n','n','m','e'

'f','y','y','t','a','f','c','b','n','e','r','s','y','w','w','p','w','o','p','n','s','p','e'

'x','y','n','t','a','f','c','b','w','e','r','s','y','w','w','p','w','o','p','n','s','p','e'

'f','y','n','t','a','f','c','b','w','e','r','s','y','w','w','p','w','o','p','n','y','g','e'

'x','y','w','t','a','f','c','b','w','e','c','s','s','w','w','p','w','o','p','n','n','m','e'

'f','f','n','f','n','f','w','b','h','t','e','s','s','w','w','p','w','o','e','k','s','g','e'

Unseen Data: (23 record)

'x','s','y','t','l','f','c','b','k','e','c','s','s','w','w','p','w','o','p','n','n','m'

'x','y','w','t','p','f','c','n','w','e','e','s','s','w','w','p','w','o','p','n','s','g'

'f','y','n','t','a','f','c','b','p','e','r','s','y','w','w','p','w','o','p','k','y','g'

's','f','n','f','n','f','c','n','p','e','e','s','s','w','w','p','w','o','p','n','v','u'

'b','s','y','t','l','f','c','b','w','e','c','s','s','w','w','p','w','o','p','k','n','g'

'b','y','w','t','l','f','c','b','n','e','c','s','s','w','w','p','w','o','p','k','n','m'

'f','y','n','t','a','f','c','b','n','e','r','s','y','w','w','p','w','o','p','k','s','g'

'b','y','y','t','a','f','c','b','g','e','c','s','s','w','w','p','w','o','p','k','s','m'

'b','y','w','t','a','f','c','b','g','e','c','s','s','w','w','p','w','o','p','k','s','g'

'x','y','n','t','l','f','c','b','n','e','r','s','y','w','w','p','w','o','p','n','y','p'

'f','f','g','f','n','f','c','n','p','e','e','s','s','w','w','p','w','o','p','n','y','u'

'x','f','g','f','n','f','c','n','g','e','e','s','s','w','w','p','w','o','p','k','y','u'

'b','y','y','t','l','f','c','b','n','e','c','s','s','w','w','p','w','o','p','k','s','g'

'x','s','y','t','l','f','c','b','n','e','c','s','s','w','w','p','w','o','p','k','s','m'

'b','y','w','t','l','f','c','b','k','e','c','s','s','w','w','p','w','o','p','n','s','g'

'x','s','w','t','a','f','c','b','g','e','c','s','s','w','w','p','w','o','p','k','s','m'

'b','s','y','t','a','f','c','b','g','e','c','s','s','w','w','p','w','o','p','n','n','m'

'x','s','y','t','a','f','c','b','k','e','c','s','s','w','w','p','w','o','p','k','s','m'

'x','f','g','f','n','f','w','b','p','t','e','f','f','w','w','p','w','o','e','k','s','g'

'f','s','y','t','a','f','w','n','p','t','b','s','s','w','w','p','w','o','p','n','v','d'

'x','y','w','t','p','f','c','n','p','e','e','s','s','w','w','p','w','o','p','k','s','g'

'x','f','w','f','n','f','w','b','k','t','e','f','s','w','w','p','w','o','e','k','a','g'

'b','y','w','t','a','f','c','b','w','e','c','s','s','w','w','p','w','o','p','k','s','g'

# ประวัติผู้เขียน

| | |
|---|---|
| **ชื่อผู้เขียน** | นางสาวปริญา คำเกลี้ยง |
| **วันเดือนปีเกิด** | 20 เมษายน 2525 |
| **สถานที่เกิด** | นครราชสีมา |
| **วุฒิการศึกษาระดับปริญญาตรี** | วิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์)<br>คณะวิทยาศาสตร์<br>มหาวิทยาลัยรามคำแหง |
| **ปีที่สำเร็จการศึกษา** | ปีการศึกษา 2549 |