การพัฒนาโปรแกรมการจัดหมวดหมู่ข้อมูลด้วยอัลกอริทึม CHAID

SOFTWARE DEVELOPMENT FOR CLASSIFICATION WITH CHAID ALGORITHM

โดย

ชัยธวัช ธีรานุสนธิ์

CHAITAWAT TEERANUSOUN

อาจารย์ที่ปรึกษา

รศ.ดร.วรพจน์ กรีสุระเดช

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน
หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
คณะเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ภาคฤดูร้อน ปีการศึกษา 2550

SOFTWARE DEVELOPMENT FOR CLASSIFICATION WITH CHAID ALGORITHM

CHAITAWAT TEERANUSOUN

A SYSTEM DEVELOPMENT PROJECT

OF THE REQUIREMENT FOR THE DEGREE OF

MASTER OF SCIENCE PROGRAM IN INFORMATION TECHNOLOGY

FACULTY OF INFORMATION TECNOLOGY

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

SUMMER / 2007

หัวข้อ	การพัฒนาโปรแกรมการจัดหมวดหมู่ข้อมูลด้วยอัลกอริทึม CHAID
นักศึกษา	นายชัยธวัช ธีรานุสนธิ์
รหัสนักศึกษา	46066709
ปริญญา	วิทยาศาสตรมหาบัณฑิต
สาขาวิชา	เทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2550
อาจารย์ที่ปรึกษา	รศ. ดร.วรพจน์ กรีฑุระเดช

บทคัดย่อ

การดำเนินธุรกิจในปัจจุบันมีการแข่งขันกันสูง ผู้บริหารจำเป็นต้องสร้างความได้เปรียบด้วยการใช้เทคนิค และกลยุทธ์ต่างๆ มาช่วยให้การดำเนินธุรกิจอยู่เหนือคู่แข่ง โดยอาศัยความรู้และสารสนเทศทั้งจากภายในและภายนอกองค์กร จึงได้มีการนำเทคนิคดาต้าไมนิ่งมาใช้เพื่อวิเคราะห์ข้อมูลในฐานข้อมูลให้ได้สารสนเทศและความรู้ที่ซ่อนอยู่

โครงการพัฒนาระบบงานชิ้นนี้ เป็นการพัฒนาระบบการจัดหมวดหมู่ข้อมูล อาศัยแบบจำลองที่มีโครงสร้างเป็นแขนงการตัดสินใจ (Decision Tree) ซึ่งถูกสร้างโดยอัลกอริทึม CHAID

การทำงานของอัลกอริทึม CHAID นั้น ใช้การคำนวณทางสถิติเพื่อระบุค่านัยสำคัญหรือความสัมพันธ์ของแอ็ตทริบิวแต่ละตัวกับแอ็ตทริบิวเป้าหมาย ค่าที่ได้จะนำมาใช้เลือกแอ็ตทริบิวที่จะสร้างเป็นตัวคัดแยกในโนดแต่ละโนดของแขนงการตัดสินใจ

จุดเด่นของอัลกอริทึม CHAID คือความสามารถในการยุบรวม (Merge) Categories ที่อยู่ในแอตทริบิว ซึ่งจะช่วยลดการแตกโนดที่ไม่จำเป็น แขนงการตัดสินใจที่ได้จึงมีขนาดไม่ใหญ่เกิดความจำเป็น (มีเฉพาะโนดที่มีความหมายเท่านั้น)

ผลลัพธ์ที่ได้จากโปรแกรมนี้ มี 2 ลักษณะ คือ 1) การจัดหมวดหมู่ของข้อมูลใหม่ที่เรายังไม่รู้ค่าหมวดหมู่ที่แน่นอน และ 2) กฎความสัมพันธ์ของข้อมูล ซึ่งเป็นสารสนเทศ (Information) ที่อยู่ในฐานข้อมูล ผู้ใช้สามารถนำไปวิเคราะห์ – ตีความ เพื่อให้ความรู้และนำไปประยุกต์ใช้ต่อไปได้

Title	Software Development for Classification with CHAID Algorithm
Student	Mr.Chaitawat Teeranusoun
Student ID.	46066709
Degree	Master of Science
Programme	Information Science
Academic Year	2007
Advisor	Assoc. Prof. Dr.Worapoj Kreesuradej

ABSTRACT

Now operating business has high competition. Executive need to create something that has an advantage with several techniques and strategies. To help business above competition is implement knowledge and information from inside and outside organization. This implement knowledge and information use data mining techniques to analysis data in database, to get information, and to seek hidden knowledge.

This project develops a tool that can arrange classification with model that has decision tree structure and is created by CHAID algorithm.

The process of CHAID algorithm is statistic calculation to define important value or relative value between each attribute and target attribute. Then, these values can be implemented to choose attribute that is created to filter in each node of tree.

The advantage of CHAID algorithm can merge category inside attribute that can decrease unnecessarily distributed node, and this affect to Decision Tree Structure which does not has too large size.

The benefit of this program has 2 points. First, it is classification of data that we cannot absolutely know of its class. Second, the rule of relative data is information in database. User can analysis and learn it to get knowledge and take it to implement.

กิตติกรรมประกาศ

โครงการพัฒนาระบบงานชิ้นนี้สำเร็จลุล่วงได้ โดยอาศัยความรู้ คำแนะนำ และคำปรึกษาจาก รศ.ดร.วรพจน์ กรีสุระเดช ซึ่งเป็นอาจารย์ที่ปรึกษา ข้าพเจ้ารู้สึกทราบซึ้งในความอนุเคราะห์จากท่านอาจารย์ และขอขอบพระคุณเป็นอย่างสูง

ขอขอบคุณ อาจารย์วารุนี เครือคล้าย ที่ได้ให้คำแนะนำเรื่องคณิตศาสตร์และสถิติที่เกี่ยวข้องกับโครงการพัฒนาระบบงานชิ้นนี้

ขอขอบคุณ เอื้อ จา กบ มี่ ก้อย แปน อ้อม และเพื่อนๆ IS16.1 ทุกคนที่ให้คำแนะนำ ความช่วยเหลือต่างๆ และกำลังใจแก่ข้าพเจ้าเสมอมา

สุดท้ายนี้ข้าพเจ้าขอกราบขอบพระคุณ บิดา มารดา และครอบครัวของข้าพเจ้าที่เป็นกำลังใจ และให้การสนับสนุนทุกสิ่งทุกอย่าง ช่วยทำให้ข้าพเจ้าสามารถทำโครงการพัฒนาระบบชิ้นนี้สำเร็จลุล่วงได้

คุณค่าและประโยชน์อันพึงมาโครงการพัฒนาระบบชิ้นนี้ ข้าพเจ้าขอมอบแด่ผู้มีพระคุณทุกท่าน

ชัยธวัช ธีรานุสนธิ์

สารบัญ

หน้า

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

สารบัญรูป

สารบัญรูป (ต่อ)

บทที่ 1
บทนำ

1.1 หลักการและเหตุผล

ในปัจจุบันการจัดเก็บข้อมูลและการทำฐานข้อมูลได้รับการพัฒนาให้มีประสิทธิภาพสูง มีการจัดเก็บข้อมูลไว้เป็นจำนวนมาก หน่วยงานและองค์กรต่างๆ พยายามนำข้อมูลที่มีจำนวนมากนี้ทั้งจากภายในและภายนอกองค์กรมาวิเคราะห์เพื่อให้ได้สารสนเทศและความรู้ใหม่ๆ มาใช้เพิ่มประสิทธิภาพการทำงาน

ดาต้าไมน์นิ่ง (Data Mining) เป็นเครื่องมือที่ช่วยวิเคราะห์และค้นหาสารสนเทศและองค์ความรู้ (Knowledge) จากข้อมูลจำนวนมาก ผลลัพท์จากการทำไมน์นิ่งนอกจากจะแสดงให้เห็นรูปแบบและความสัมพันธ์ของข้อมูลแล้ว ยังสามารถนำไปสร้างแบบจำลอง (Model) เพื่อประยุกต์ใช้ในงานด้านต่างๆ เช่น การจัดหมวดหมู่ (Classification) และการพยากรณ์ (Prediction) ข้อมูลในอนาคต

สารสนเทศและองค์ความรู้ที่ได้จากดาต้าไมน์นิ่งถูกนำมาใช้เพื่อสนับสนุนการตัดสินใจ เป็นการพัฒนาการทำงาน สร้างโอกาสและความความได้เปรียบในการดำเนินธุรกิจ

1.2 วัตถุประสงค์ / เป้าหมาย

โครงการพัฒนาระบบงานนี้มีวัตถุประสงค์และเป้าหมายดังต่อไปนี้

1. เพื่อศึกษาหลักการของดาต้าไมน์นิ่งและการจัดหมวดหมู่
2. เพื่อศึกษาการทำงานของอัลกอริทึม CHAID (CHi-squared Automatic Interaction Detector)
3. เพื่อศึกษาการพัฒนาโปรแกรมการจัดหมวดหมู่โดยใช้อัลกอริทึม CHAID
4. เพื่อเป็นแนวทางในการประยุกต์ใช้ดาต้าไมน์นิ่งเพื่อสนับสนุนการตัดสินใจ

1.3 ขอบเขตของการดำเนินงาน

โครงการพัฒนาระบบงานนี้เป็นการศึกษาหลักการนำอัลกอริทึม CHAID มาใช้พัฒนาโปรแกรมการจัดหมวดหมู่ข้อมูล โดยโปรแกรมที่ได้มีการทำงาน 4 ระยะคือ

1. เตรียมข้อมูลสำหรับสร้างแบบจำลอง
2. สร้างแบบจำลอง (Model) ในรูปแบบแขนงการตัดสินใจ (Decision Tree) และการบันทึกแบบจำลองลงในฐานข้อมูล

3. แมพข้อมูลและจัดรูปแบบข้อมูลให้ตรงกันกับแบบจำลอง
4. จัดหมวดหมู่ข้อมูล (Classify) ข้อมูลใหม่โดยอาศัยแบบจำลองที่สร้างขึ้นในขั้นตอนที่สอง

1.4 ขั้นตอนการดำเนินงาน

การพัฒนาระบบงานนี้มีขั้นตอนการทำงานดังนี้

1. ศึกษาหลักการและกระบวนการทำงานของดาต้าไมน์นิ่ง
2. ศึกษาอัลกอริทึม CHAID เพื่อนำมาประยุกต์ใช้กับระบบ
3. รวบรวมและเตรียมข้อมูล รวมทั้งกำหนดเครื่องมือที่จะนำมาใช้ในการพัฒนาระบบ โดยในโครงการพัฒนาระบบงานนี้ได้เลือกใช้เครื่องมือดังนี้
   - เครื่องมือที่ใช้ในการพัฒนาโปรแกรมคือ MS. Visual Studio 2005
   - ระบบฐานข้อมูลคือ MS. SQL Server 2005 Developer Edition
   - เครื่องคอมพิวเตอร์โน๊ตบุค CPU Intel® Core™ 2 Duo 2.0 GHz Memory 1.0 GB
   - Operating System: Microsoft Windows XP 2002
4. วิเคราะห์และออกแบบระบบ
5. ออกแบบโครงสร้างข้อมูล ฐานข้อมูล และส่วนติดต่อกับผู้ใช้
6. เขียนโปรแกรม
7. ทดสอบการทำงานของโปรแกรม
8. สรุปผลการศึกษา

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. สามารถเข้าใจหลักการและการทำงานของดาต้าไมน์นิ่ง
2. สามารถสร้างระบบการจัดหมวดหมู่ข้อมูลโดยใช้อัลกอริทึม CHAID
3. การศึกษาและพัฒนานี้สามารถเป็นแนวทางในการประยุกต์ใช้ดาต้าไมน์นิ่งกับงานทางด้านอื่นๆ ได้ต่อไป

บทที่ 2
ดาต้าไมน์นิ่ง

ดาต้าไมน์นิ่ง คือ การค้นหารูปแบบและความสัมพันธ์ของข้อมูลในฐานข้อมูล ดาต้าไมน์นิ่ง จะทำการสำรวจและวิเคราะห์ข้อมูลดังกล่าวด้วยวิธีแบบอัตโนมัติ หรือกึ่งอัตโนมัติ เพื่อให้ได้สารสนเทศในรูปแบบที่มีความหมายอยู่ในรูปของกฎ สารสนเทศที่ได้นี้แสดงให้เห็นถึงความรู้ต่างๆ ที่มีประโยชน์ซึ่งถูกเก็บอยู่ในฐานข้อมูล มีความถูกต้อง และสามารถนำไปใช้จริงได้

อาจเรียกได้ว่า ดาต้าไมน์นิ่งเป็นการค้นหาความรู้ในฐานข้อมูล (Knowledge Discovery in Database : KDD)

2.1 การทำงานของดาต้าไมน์นิ่ง (Data Mining Process)

ดาต้าไมน์นิ่งเป็นกระบวนการในการค้นหารูปแบบและความสัมพันธ์ของข้อมูลที่ซ่อนอยู่ เพื่อสร้างความรู้ใหม่เกี่ยวกับข้อมูลนั้นๆ โดยใช้การวิเคราะห์ทางสถิติและเทคนิคในการสร้างแบบจำลอง

ดาต้าไมน์นิ่งนำเอาวิธีการสร้างแบบจำลอง (Model) มาช่วยในการค้นหารูปแบบและความสัมพันธ์ของข้อมูล แบบจำลองเป็นเสมือนการจำลองสถานการณ์จริง ซึ่งแบบจำลองที่ดีจะมีประโยชน์ในการทำความเข้าใจกับธุรกิจและบอกได้ถึงสิ่งที่ควรปฏิบัติเพื่อทำให้เกิดความสำเร็จในธุรกิจ

2.2 งานของดาต้าไมน์นิ่ง

เราสามารถจำแนกงานทางด้านดาต้าไมน์นิ่งออกเป็น 4 ประเภท ได้แก่

1. **การวิเคราะห์ความสัมพันธ์ (Association Analysis)** เป็นการหาความสัมพันธ์ของข้อมูลต่างๆ เช่น การหาความสัมพันธ์ของการซื้อสินค้าชนิดหนึ่งกับสินค้าประเภทอื่น
2. **การจัดหมวดหมู่และการทำนายค่า (Classification and Prediction)** เป็นการนำข้อมูลที่เรารู้หมวดหมู่หรือคลาส (Class) ของมันแล้ว มาหารูปแบบ (Patten) หรือความสัมพันธ์ของคลาสกับแอ็ตทริบิว (Attribute) อื่นๆ แล้วใช้รูปแบบหรือความสัมพันธ์ดังกล่าวมาสร้างแบบจำลอง (Model) แล้วใช้แบบจำลองนี้ไปจัดหมวดหมู่หรือทำนายค่าข้อมูลใหม่ที่เรายังไม่ทราบคลาส
3. **การแบ่งกลุ่มข้อมูล (Clustering)** เป็นการจัดแบ่งข้อมูลเป็นกลุ่มๆ ตามลักษณะความคล้ายคลึงกันหรือการเกาะกลุ่มกันของข้อมูล

4. **การตรวจสอบค่าเบี่ยงเบน (Deviation Detection)** เป็นการหาค่าที่มีความแตกต่างไปจากค่ามาตรฐานว่ามีค่ามากน้อยเพียงใด แบบจำลองของการตรวจสอบค่าเบี่ยงเบนจะที่ใช้เทคนิคทางสถิติ (Statistics) เพื่อวัดความน่าเชื่อถือของข้อมูล และการแสดงให้เห็นภาพ (Visualization) ซึ่งเป็นการแสดงผลข้อมูลออกมาในรูปแบบกราฟิก เช่น แผนภูมิแท่ง แผนภูมิวงกลม เพื่อให้ผู้ใช้สามารถเข้าใจได้ง่าย

2.3 รูปแบบการเก็บข้อมูลที่สามารถทำดาต้าไมน์นิ่ง

1) Relational Databases: ฐานข้อมูลเชิงสัมพันธ์ คือฐานข้อมูลที่เก็บข้อมูลในลักษณะตารางซึ่งมีความสัมพันธ์ของแอตทริบิวระหว่างตาราง เช่น MySQL, MS. Access, MS. SQL-Server
2) Data Warehouses: คลังข้อมูล เป็นระบบการเก็บข้อมูลจากอดีตถึงปัจจุบัน บนฐานข้อมูลเชิงปฏิบัติการขององค์กร ซึ่งเป็นแหล่งข้อมูลส่วนกลางที่ได้ถูกคัดเลือก ปรับปรุง จัดรูปแบบ และรวบรวม เพื่อนำมาประมวลผล วิเคราะห์ และสรุปเป็นสารสนเทศ
3) Transactional Databases: ฐานข้อมูลรายการเปลี่ยนแปลง เก็บรายละเอียดของข้อมูลการดำเนินกิจการขององค์กร เช่น การซื้อ – ขาย การขนส่ง การจัดเก็บสินค้า อาจเรียกว่า ฐานข้อมูลเชิงปฏิบัติการ (Operational Database)
4) Advanced Database Systems and Advanced Database Applications: ระบบฐานข้อมูลชั้นสูง เป็นการรวมแนวคิดแบบจำลองฐานข้อมูลชนิดต่างๆ เช่น ฐานข้อมูลเชิงเชิงสัมพันธ์ ฐานข้อมูลเชิงวัตถุ(เป็นข้อมูลที่มีลักษณะเป็น Object) ฐานข้อมูลเชิงลำดับชั้น (มีการถ่ายทอดคุณสมบัติหลายชั้น) ฐานข้อมูลแบบกระจาย (Distributes Database)
5) Object – Oriented Databases: ฐานข้อมูลเชิงวัตถุ เป็นระบบฐานข้อมูลที่จัดเก็บข้อมูลแสดงอยู่ในรูปแบบของ Object ทำให้ระบบฐานข้อมูลมีความสามารถทำงานร่วมกับภาษาโปรแกรมเชิงวัตถุ (Object Programming Languages) ได้
6) Object – Relational Databases: ฐานข้อมูลเชิงวัตถุสัมพันธ์ เป็นการรวมข้อดีของแนวคิดฐานข้อมูลเชิงสัมพันธ์และฐานข้อมูลเชิงวัตถุ เป็นฐานข้อมูลเชิงสัมพันธ์ที่สามารถรองรับและประมวลผลข้อมูลเชิงวัตถุได้
7) Temporal Databases and Time – Series Databases: เป็นระบบข้อมูลที่บันทึกรายละเอียดเกี่ยวกับเวลาที่เกิดกิจกรรมหรือการเปลี่ยนแปลงขึ้น เพื่อนำไปใช้ประมวลผลเปรียบเทียบต่อไป
8) Spatial Databases: ฐานข้อมูลปริภูมิ เป็นฐานข้อมูลที่สามารถเก็บข้อมูลเชิงภูมิศาสตร์ (Geographic) ได้ (ใช้กับเทคนิคดาต้าไมน์นิ่งชั้นสูง)

9) Text Databases and Multimedia Databases: ฐานข้อมูลที่เก็บข้อมูลในรูปแบบตัวอักษรและสื่อสนเทศต่างๆ

10) Heterogeneous Databases: แบบจำลองเชิงลดหลั่น มีลักษณะคล้ายแผนภูมิต้นไม้หัวกลับ ประกอบด้วย node ต่างๆ node ที่อยู่ชั้นบนสุดคือ root node ส่วน node ที่อยู่ชั้นที่สูงกว่า node อื่นคือ parent และ node ที่อยู่ชั้นต่ำกว่า คือ child โดยที่ 1 parent มี child ได้หลาย node แต่ child 1 node จะมี parent ได้เพียง node เดียว

11) The World Wide Web: คือข้อมูลที่ออนไลน์อยู่ในอินเตอร์เน็ต เช่น web site และเอกสารออนไลน์ต่างๆ

2.4 รูปแบบข้อมูลตัวแปรในการทำดาต้าไมน์นิ่ง (Type of Attributes)

1. ตัวแปรแบบลำดับขั้น (Categorical) จำแนกออกเป็น

1.1. Nominal ข้อมูลที่ลำดับไม่มีความสำคัญ เช่น สถานภาพการแต่งงาน (โสด สมรส หย่า)

1.2. Ordinal เป็นตัวแปรที่มีการจัดลำดับ เช่น อุณหภูมิ hot, mild, cool โดยที่ hot > mild > cool

2. ตัวแปรแบบปริมาณ (Quantitative) จำแนกออกเป็น

2.1. ข้อมูลต่อเนื่อง (Continuous) ค่าที่เก็บจะเป็นตัวเลขจำนวนจริง (Real number) เช่น ข้อมูลรายได้

2.2. ข้อมูลไม่ต่อเนื่อง (Discrete) ค่าที่เก็บจะเป็นเลขจำนวนเต็ม (Integer) เช่น ข้อมูลจำนวนพนักงาน

2.5 ขั้นตอนในการทำดาต้าไมน์นิ่ง

1. การกำหนดวัตถุประสงค์ (Business Objective Determination)

การกำหนดวัตถุประสงค์ เป็นการทำความเข้าใจปัญหาและความต้องการทางธุรกิจของหน่วยงานหรือองค์กร ขั้นตอนนี้เป็นการพิจารณาว่าจะนำดาต้าไมน์นิ่งไปแก้ไขปัญหาใด ลักษณะและประเภทของข้อมูลที่มีอยู่สามารถทำการค้นหาสารสนเทศที่ต้องการได้หรือไม่

2. การเตรียมข้อมูล (Data Preparation)

การจัดเตรียมข้อมูล คือ การจัดรูปแบบข้อมูลให้อยู่ในฟอร์มมาตรฐาน (Standard Form) ที่มีความเหมาะสม และเตรียมคุณสมบัติของข้อมูลเพื่อนำไปสู่ประสิทธิภาพสูงสุดของการทำไมน์นิ่ง

การเตรียมข้อมูลประกอบด้วย 3 ขั้นตอนได้แก่

2.1. **การเลือกข้อมูล (Data Selection)**

การเลือกข้อมูลคือการเลือกตารางและแอตทริบิวของข้อมูลที่มีความหมาย สามารถนำไปทำไมน์นิ่งได้

2.2. **การประมวลผลข้อมูลเบื้องต้น (Data Preprocessing)**

2.2.1. **การกำจัดค่าว่าง (Data Cleaning)** เป็นขั้นตอนในการกำจัด missing value , noisy data และ inconsistency data

2.2.2. **การรวมข้อมูล Data Integration** คือการรวมฐานข้อมูลหรือไฟล์ข้อมูลจากหลายๆ แหล่งเข้าเป็นชุดเดียวกัน เพื่อความสำดวกต่อการทำไมน์นิ่ง

2.3. **การแปลงข้อมูล (Data Transformation)** คือการจัดรูปแบบข้อมูลให้อยู่ในรูปแบบที่เหมาะสมในการทำไมน์นิ่งตามอัลกอริทึมของดาต้าไมน์นิ่งที่เลือกใช้ เช่น การแปลงข้อมูลจากแอตทริบิวชนิด Numeric ให้เป็นแบบ Categorical

3. **การทำดาต้าไมน์นิ่ง**

คือกระบวนการประมวลผลข้อมูลตามอัลกอริทึมที่ได้กำหนดไว้

4. **การวิเคราะห์ผลลัพธ์ที่ได้**

เป็นการวิเคราะห์และแปรความหมายจากผลที่ได้จากการทำไมน์นิ่ง เช่น การศึกษาพฤติกรรมของลูกค้าซึ่งอาจจะไม่ได้คำตอบออกมาตรงๆ แต่จะได้รูปแบบความสัมพันธ์จำนวนมาก ต้องอาศัยความรู้และประสบการณ์มาวิเคราะห์และตีความกฎเหล่านี้ ขั้นตอนนี้อาจต้องใช้เวลานาน เราอาจเลือกวิเคราะห์เฉพาะผลลัพธ์ของดาต้าไมน์นิ่งที่น่าสนใจ คือเป็นผลลัพธ์ที่ได้ง่ายต่อความเข้าใจ เป็นสารสนเทศที่ไม่เคยรู้มาก่อน และมีความสมเหตุสมผล

5. **การนำสารสนเทศไปใช้ประโยชน์**

การนำความรู้ที่ได้ไปใช้ประโยชน์เป็นขั้นตอนสุดท้ายของดาต้าไมน์นิ่ง คำตอบที่ได้จากการวิเคราะห์และตีความผลลัพธ์จากการทำไมน์นิ่งจะถูกนำมารวมกันกับความรู้ที่มีอยู่เดิม เพื่อให้เกิดความรู้ใหม่ จากนั้นนำความรู้ใหม่ที่ได้นี้ไปใช้ให้เกิดประโยชน์ ในขั้นตอนนี้จะมีหลักอยู่ 2 ประการคือ

- แสดงแนวคิดที่ค้นพบใหม่
- นำความรู้ใหม่ที่พบไปใช้ให้เกิดประโยชน์สูงสุด

2.6 การจัดหมวดหมู่

การจัดหมวดหมู่เป็นการนำข้อมูลมาจำแนกว่าตรงกับประเภทใดที่เรากำหนดไว้ โดยการจัดหมวดหมู่ประกอบด้วยการสำรวจจุดเด่นของข้อมูลที่ปรากฏ นำมาสร้างเป็นแบบจำลอง และนำแบบจำลองดังกล่าวไปใช้แบ่งหมวดหมู่

ตัวอย่างของงานจัดหมวดหมู่ ได้แก่ การแยกประเภทเครดิตของผู้ขอเงินกู้ โดยแบ่งตามความเสี่ยงสูง กลาง ต่ำ

2.7 การสร้างแบบจำลองสำหรับการจัดหมวดหมู่ (Predictive Modeling)

เป็นการสร้างแบบจำลองจากข้อมูลที่มีอยู่เพื่อใช้ทำนายความเป็นไปได้ โดยการสังเกตจากลักษณะของข้อมูล คือจะวิเคราะห์ข้อมูลที่มีอยู่เพื่อหารูปแบบ (Patten) หรือความสัมพันธ์กับข้อมูลต่างๆ กับคลาสหรือข้อมูลเป้าหมาย (Target Attribute) แล้วสร้างเป็นแบบจำลอง จากนั้นนำแบบจำลองที่ได้ไปวิเคราะห์ข้อมูลที่เราต้องการทราบหมวดหมู่หรือคลาส วิธีการเช่นนี้เรียกว่า Supervised Learning

เพื่อให้ได้แบบจำลองที่มีความแม่นยำละเชื่อถือได้ ข้อมูลที่ใช้สร้างแบบจำลองต้องมีความถูกต้อง และเป็นความจริง เนื่องจากระบบคาดว่าไม่นิ่งจะทำการถอดรูปแบบ – ความสัมพันธ์ของข้อมูลนั้น แล้วนำไปสร้างเป็นแบบจำลอง ข้อมูลที่ถูกต้องเท่านั้นจึงจะสามารถสร้างแบบจำลองที่ให้ผลลัพธ์ได้ถูกต้อง

การสร้างแบบจำลองพยากรณ์แบ่งเป็น 2 ขั้นตอน คือ

- Training Phase เป็นการสร้างแบบจำลองโดยอาศัยข้อมูลในอดีต ซึ่งต้องใช้ข้อมูลมากถึง 80% ของข้อมูลทั้งหมด
- Testing Phase คือการทดสอบแบบจำลองที่สร้างว่ามีความถูกต้องหรือไม่ โดยจะนำข้อมูลส่วนที่เหลือประมาณ 20% มาเป็นตัวทดสอบ

แบบจำลองที่ใช้ในการทำ Classification มีอยู่หลายรูปแบบ ที่รู้จักกันทั่วไป ได้แก่

1. แขนงการตัดสินใจ (Decision Tree)

Decision Tree หรือแขนงการตัดสินใจ เป็นแผนภูมิรูปภาพที่มีโครงสร้างเหมือนต้นไม้หัวกลับ ประกอบด้วยหน่วย (Node) ต่างๆ ได้แก่ โนดภายใน (internal node) แสดงการทดสอบเงื่อนไขของรายละเอียดต่างๆ ของข้อมูลแต่ละระเบียน กิ่งแขนงแสดงผลของการทดสอบใบแสดงกลุ่มหรือหมวดหมู่ที่ได้กำหนดไว้ และโนดที่อยู่ชั้นบนสุดเรียกว่า "โนดราก" (root node)

ชุดข้อมูลที่นำมาจัดหมวดหมู่โดยแขนงการตัดสินใจจะถูกนำเข้ามาทางหน่วยรากซึ่งอยู่ชั้นบนสุด และถูกทดสอบตามเงื่อนไขในแต่ละโนดตามลำดับจนมาถึงโนดใบ (leaf node)ที่แสดงผลของการจัดหมวดหมู่ วิธีการดังกล่าวนี้ สามารถแปลงไปเป็น "กฎการจัดหมวดหมู่" ได้ง่าย

2. เมมโมรีเบสรีซันนิ่ง (Memory-Based Reasoning: MBR)

เมมโมรีเบสรีซันนิ่ง คือ วิธีดาต้าไมน์นิ่งทางตรง (Direct Data Mining) อาศัยข้อเท็จจริงที่ทราบค่าแล้วในแบบจำลองมาใช้ในการทำนายค่าของข้อมูลที่ยังไม่ทราบค่า เมมโมรีเบสรีซันนิ่งจะค้นหาสิ่งที่มีค่าใกล้เคียงที่สุดกับข้อเท็จจริงที่ยังไม่ทราบค่า (Nearest Neighbor) เพื่อใช้ในการจัดหมวดหมู่หรือการทำนายค่าออกมา

3. นิวรอนเน็ตเวิร์ค (Neural Network)

นิวรอนเน็ตเวิร์ค คือระบบการประมวลผลข้อมูล ที่นำคุณสมบัติของไบโอลอจิกคอล นิวรอนเน็ตเวิร์ค (Biological Neural Network) มาใช้ ถูกพัฒนาขึ้นโดยแบบจำลองทางคณิตศาสตร์ ซึ่งจำลองมาจากการเรียนรู้ของมนุษย์

นิวรอนเน็ตเวิร์คประกอบไปด้วยหน่วยจำนวนมาก เรียกว่า นิวรอน ยูนิต เซล หรือ โหนด แต่ละนิวรอนต่อกันโดยใช้คอนเนคชั่นลิ้งค์ (Connection Link) ที่มีค่าน้ำหนัก (Weight) ของมันอยู่ โดยค่าน้ำหนักจะแสดงถึงรายละเอียดที่เน็ตเวิร์คใช้ในการแก้ปัญหา

นิวรอนเน็ตเวิร์คเป็นเทคนิคที่ได้รับความนิยมอย่างกว้างขวาง โดยนำไปแก้ปัญหาต่างๆ เช่น การเก็บและการเรียกข้อมูล, การแยกประเภทของข้อมูล, การเปลี่ยนจากรูปแบบของอินพุทไปเป็นรูปแบบของเอาท์พุท และการตรวจสอบรูปแบบของข้อมูลที่มีความคล้ายคลึงกับความคิดของมนุษย์

2.8 ประโยชน์ของดาต้าไมน์นิ่ง

การบวนการดาต้าไมน์นิ่งมีด้วยกันหลากหลายวิธีการ ขึ้นอยู่กับความเหมาะสมของข้อมูลและความต้องการ แต่ทุกวิธีล้วนมีประโยชน์ในทำนองเดียวกัน คือ

1. รูปแบบและการแสดงผลที่ได้ง่ายต่อการเข้าใจ ผู้ที่ไม่มีความรู้พื้นฐานทางสถิติก็สามารถแปรความจากตัวแบบได้ สามารถนำสารสนเทศที่ได้ไปใช้ในกระบวนการทางธุรกิจได้

2. สามารถวิเคราะห์ข้อมูลจำนวนมากได้

3. ดาต้าไมน์นิ่งสามารถค้นพบในสิ่งที่เราคาดไม่ถึง เป็นข้อเท็จจริงที่เราไม่รู้มาก่อน เนื่องจากเทคนิคที่หลากหลายของดาต้าไมน์นิ่งจะค้นหาและตรวจสอบรูปแบบ ความสัมพันธ์และความเชื่อมโยงของตัวแปรแต่ละตัว การค้นหาจากตัวแปรหลายๆที่นำมารวมกันนี้ ช่วยให้ค้นพบสารสนเทศใหม่ๆ

4. สามารถนำดาต้าไมน์นิ่งมาใช้ในการค้นหารูปแบบและคำตอบจากข้อมูลที่เป็นปัจจุบัน

5. สามารถนำดาต้าไมน์นิ่งมาใช้ในการพยากรณ์แนวโน้มในอนาคตจากข้อมูลในอดีตและปัจจุบัน

บทที่ 3
อัลกอริทึม CHAID

3.1 อัลกอริทึม CHAID

อัลกอริทึม CHAID (CHi-squared Automatic Interaction Detector) (Kass, G.V. 1980 : 119-127) เป็นอัลกอริทึมหนึ่งที่ใช้สร้างแบบจำลองที่มีลักษณะเป็นแขนงการตัดสินใจ (Decision Tree) มีความสามารถในการจัดหมวดหมู่ข้อมูลซึ่งมีลักษณะเป็นลำดับขั้น (categories) การแตกกิ่งแขนงสามารถแตกกิ่งได้หลายกิ่งจากโนดๆ เดียว

ข้อมูลที่ต้องการจัดหมวดหมู่จะถูกนำเข้ามาทางโนดราก (root node) และจะถูกทดสอบเงื่อนไขโดยตัวคัดแยกที่อยู่ในโนดต่างๆ ผลลัพธ์จากการทดสอบในแต่ละโนดมีลักษณะเป็น categories จากนั้นข้อมูลจะถูกส่งไปตามกิ่งแขนงและจะถูกทดสอบเงื่อนไขโดยตัวคัดแยกในโนดภายใน (internal node) ไปเรื่อยๆ จนถึงโนดใบ (leaf node) ซึ่งเป็นหมวดหมู่ของข้อมูลนั้น

3.1.1 การทำงานของอัลกอริทึม CHAID

CHAID เป็นอัลกอริทึมที่ใช้หลักการทางสถิติคือการทดสอบ Chi-square เพื่อหาความสัมพันธ์ระหว่างตัวแปรต้นหรือแอตทริบิวนั้นๆ กับผลลัพธ์หรือแอตทริบิวเป้าหมายซึ่งก็คือหมวดหมู่ที่เราต้องการ แอตทริบิวที่มีความสัมพันธ์กับผลลัพธ์หรือแอตทริบิวเป้าหมายสูง จะมีความสามารถในการคัดแยกสูงด้วย

อัลกอริทึม CHAID ทำการสร้างแขนงการตัดสินใจโดยสร้างตัวคัดแยกจากแอตทริบิวที่มีความสามารถในการคัดแยกสูงที่สุด แล้วแบ่งกลุ่มข้อมูลออกตาม categories ของแอตทริบิวนั้น จากนั้นทำการสร้างตัวคัดแยกใหม่เพื่อแบ่งกลุ่มข้อมูลเหล่านี้อีก ทำเช่นนี้ไปเรื่อยๆ จนกว่าค่าของตัวแอตทริบิวเป้าหมาย (ผลลัพธ์หรือหมวดหมู่ที่เราต้องการ) ของข้อมูลในแต่ละโนดจะมีค่าเหมือนกันหรืออยู่ใน categories เดียวกัน

ตัวคัดแยกเหล่านี้ถูกบรรจุไว้ในโนดต่างๆ ในการสร้างแขนงการตัดสินใจ จะสร้างโนดราก (root node) เป็นโนดแรก จากนั้นจึงสร้างโนดภายใน (internal node) และโนดใบ (leaf node) ตามลำดับ

เนื่องจากอัลกอริทึม CHAID รองรับข้อมูลที่มีลักษณะเป็น category แต่ categories บาง categories มีนัยสำคัญต่ำ จำเป็นต้องรวม categories เหล่านี้เข้าไว้ด้วยกันเพื่อให้มีนัยสำคัญมากขึ้น โดยพิจารณาจาก ค่า P-Value ที่คำนวณได้จากการทดสอบ Chi-square เป็นตัววัดความมีนัยสำคัญ ซึ่งอัลกอริทึม CHAID กำหนดค่าความน่าเชื่อถือไว้ที่ 95% (ค่า P-Value ต่ำกว่า 0.05) กระบวนการนี้เรียกว่า "การรวมลำดับขั้น" (Merging categories) การสร้างกิ่งแขนงเพื่อแสดงผลลัพธ์จากการทดสอบเงื่อนไขจะสร้างตาม categories ที่ได้ทำการ merge แล้ว

การสร้างแขนงการตัดสินใจโดยใช้อัลกอริทึม CHAID มีขั้นตอนดังนี้

- ขั้นที่ 1 คำนวณลักษณะของผลลัพธ์หรือแอตทริบิวเป้าหมาย (Target Attribute) ของชุดข้อมูลตัวอย่าง (Training Data) แบ่งเป็นลำดับขั้น (Categories) ตามหมวดหมู่ที่ต้องการ
- ขั้นที่ 2 เปรียบเทียบความสามารถในการคัดแยกระหว่างแอตทริบิวแต่ละตัวกับแอตทริบิวเป้าหมายโดยการนำแอตทริบิวแต่ละตัวมาทำการ merge categories แล้วหาค่า P-Value แอตทริบิวที่มีความสามารถในการคัดแยกมากที่สุดคือแอตทริบิวที่มีค่า P-Value น้อยที่สุด จะถูกใช้เป็นตัวคัดแยกในโนดของแขนงการตัดสินใจ มีกระบวนการดังนี้

i. หาความมีนัยสำคัญของแอตทริบิวต่างๆ โดยเลือกแอตทริบิวมา 1 ตัว จับคู่ categories ของแอตทริบิวนั้นโดยจับคู่แบบพบกันหมด (Combination) ครั้งละ 2 categories ทำการทดสอบ Chi-square หาค่า P-Value ระหว่าง 2 categories นั้นกับ categories ของแอตทริบิวเป้าหมาย ถ้าค่า P-Value ที่ได้มีค่ามากกว่า 0.05 จะหมายถึงคู่ categories นั้นมีนัยสำคัญต่ำ ต้องทำการ merge คู่ categories นั้นให้เป็น categories เดียวกัน โดย merge คู่ categories ที่มีค่า P-Value มากที่สุด เมื่อรวม categories กันแล้วจะทำการจับคู่ทดสอบ Chi-square หาค่า P-Value อีกครั้ง ถ้ายังมีคู่ categories ที่ค่า P-Value มีค่ามากกว่า 0.05 ก็จะทำการ merge categories อีก และจะทำซ้ำไปเรื่อยๆ จนกว่าทุกคู่ของ categories มีนัยสำคัญมากเพียงพอ (P-Value < 0.05) หรือจนกว่าจะเหลือ categories เพียง 2 categories -

ii. เมื่อทำการรวม categories เสร็จแล้ว จะทำการคำนวณค่า Bonferroni multiplier เพื่อปรับปรุงค่า P-Value โดยแบ่งประเภทของแอตทริบิวเป็น 3 ประเภท ได้แก่

1. Monotonic คือOrdinal categorical หรือข้อมูลอันดับ เป็นข้อมูลที่สามารถเรียงลำดับความมากน้อยได้ เช่น ช่วงอายุ รายได้
2. Free คือ Nominal categorical (นามบัญญัติ) เป็นการระบุลักษณะที่ไม่ใช่จำนวนหรือค่าที่ไม่สามารถนำมาคำนวณได้ เช่น เพศ ศาสนา
3. Floating คือ Ordinal categorical ที่มีลำดับขั้นหนึ่งเป็นลำดับขั้นที่ว่างหรือไม่สามารถระบุช่วงค่าได้ เช่น missing categories หรือ unknown level

ค่า Bonferroni multiplier คำนวณได้จากสมการดังต่อไปนี้

$$B_{Monotonic} = \binom{c-1}{r-1}$$

$$B_{free} = \sum_{i=0}^{r-1} (-1)^i \frac{(r-i)^c}{i!(r-i)!}$$

$$B_{Floating} = \binom{c-2}{r-2} + r\binom{c-2}{r-1}$$

เมื่อ c คือ จำนวน categories เริ่มต้นก่อนการ merge และ

r คือ จำนวน categories สุดท้าย (จำนวน categories หลังจากที่ได้ทำการรวมแล้ว)

- ขั้นที่ 3 ทำการหาค่าความสามารถในการคัดแยกของแอตทริบิวตัวอื่นตามขั้นตอนที่ i. และ ii. แล้วนำค่า P-Value ที่ถูกปรับปรุงแล้วมาเปรียบเทียบกัน แอตทริบิวที่ค่า P-Value น้อยที่สุดจะถูกนำมาใช้เป็นตัวคัดแยกในโนดต่างๆ โดยเริ่มต้นที่โนดราก
- ขั้นที่ 4 เมื่อสร้างตัวคัดแยกในโนดรากแล้ว เราใช้ตัวคัดแยกนี้แบ่งกลุ่มข้อมูลออกตาม categories ที่ถูก mergeไว้ เป็นการแตกกิ่งของแขนงการตัดสินใจ
- ขั้นที่ 5 นำข้อมูลในกิ่งแขนงที่แตกมาจากโดนข้างบนมาสร้างตัวคัดแยกในโนด โดยใช้แอตทริบิวที่เหลือ ทำตามขั้นตอนที่ 1 – 4 ไปเรื่อยๆจนกว่าข้อมูลจะถูกแบ่งกลุ่มอย่างสมบูรณ์ (ค่าของแอตทริบิวเป้าหมายในแต่ละกิ่งแขนงมีลักษณะเหมือนกัน หรืออยู่ใน categories เดียวกัน) โนดที่อยู่ที่ปลายกิ่งแขนงซึ่งถูกแบ่งกลุ่มอย่างสมบูรณ์แล้ว เรียกว่า โนดใบ (leaf node) เมื่อมีโนดใบอยู่ทุกปลายของกิ่งแขนง แสดงว่าข้อมูลทั้งหมดถูกแบ่งกลุ่มอย่างสมบูรณ์ จึงจบกระบวนการการสร้างแขนงการตัดสินใจ

3.2 ตัวอย่าง

กำหนดให้ชุดข้อมูลตัวอย่างมีจำนวน 100 ระเบียน ค่าของแอตทริบิวเป้าหมาย "Y" มี 4 categories คือ 1, 2, 3 และ 4 มีแอตทริบิว 2 ตัว ได้แก่ X1 และ X2 มีค่าดังนี้ X1= 1, 2, 3, 4 และ X2 = 0, 1, 2

แบ่งลักษณะของผลลัพธ์หรือค่าในแอตทริบิวเป้าหมายออกตาม categories ดังตารางที่ 3.1

ตารางที่ 3.1 ลักษณะของผลลัพธ์หรือค่าในแอตทริบิวเป้าหมาย แบ่งตาม categories

Categories	%	n
1.	35.00	35
2.	8.00	8
3.	35.00	35
4.	22.00	22
Total	(100.00)	100

เลือกแอตทริบิว "X1" มาทดสอบ Chi-Square (ตารางที่ 3.2)

ตารางที่ 3.2 แบ่งค่าของแอตทริบิว "X1" และแอตทริบิวเป้าหมาย "Y" ตาม categories

X1 \ Y	1	2	3	4	รวมแนวนอน
1	23	5	19	4	51
2	12	2	15	13	42
3	0	1	0	1	2
4	0	0	1	4	5
รวมแนวตั้ง	35	8	35	22	100

Chi-Square = 25.63559 d.f. = 9 (P-Value = 0.002342955)

จับคู่ categories ของแอตทริบิว "X1" ทดสอบ Chi-Square หาค่า P-Value (ตารางที่ 3.3 – ตารางที่ 3.8)

ตารางที่ 3.3 จับคู่ categories 1 และ 2 ของแอตทริบิว "X1" เปรียบเทียบกับแอตทริบิว "Y"

X1 \ Y	1	2	3	4	รวมแนวนอน
1	23	5	19	4	51
2	12	2	15	13	42
รวมแนวตั้ง	35	7	34	17	93

Chi-Square = 9.193281 d.f. = 3 (P-Value = 0.02682849)

ตารางที่ 3.4 จับคู่ categories 1 และ 3 ของแอตทริบิว "X1" เปรียบเทียบกับแอตทริบิว "Y"

X1 \ Y	1	2	3	4	รวมแนวนอน
1	23	5	19	4	51
3	0	1	0	1	2
รวมแนวตั้ง	23	6	19	5	53

Chi-Square = 8.019281 d.f. = 3 (P-Value = 0.0456149)

ตารางที่ 3.5 จับคู่ categories 1 และ 4 ของแอตทริบิว "X1" เปรียบเทียบกับแอตทริบิว "Y"

X1 \ Y	1	2	3	4	รวมแนวนอน
1	23	5	19	4	51
4	0	0	1	4	5
รวมแนวตั้ง	23	5	20	8	56

Chi-Square = 19.72078 d.f. = 3 (P-Value = 0.0001939266)

ตารางที่ 3.6 จับคู่ categories 2 และ 3 ของแอตทริบิว "X1" เปรียบเทียบกับแอตทริบิว "Y"

X1 \ Y	1	2	3	4	รวมแนวนอน
2	12	2	15	13	42
3	0	1	0	1	2
รวมแนวตั้ง	12	3	15	14	44

Chi-Square = 7.23356 d.f. = 3 (P-Value = 0.0648145)

ตารางที่ 3.7 จับคู่ categories 2 และ 4 ของแอตทริบิว "X1" เปรียบเทียบกับแอตทริบิว "Y"

X1 \ Y	1	2	3	4	รวมแนวนอน
2	12	2	15	13	42
4	0	0	1	4	5
รวมแนวตั้ง	12	2	16	17	47

Chi-Square = 4.962482 d.f. = 3 (P-Value = 0.1745651)

ตารางที่ 3.8 จับคู่ categories 3 และ 4 ของแอตทริบิว "X1" เปรียบเทียบกับแอตทริบิว "Y"

X1 \ Y	2	3	4	รวมแนวนอน
3	1	0	1	2
4	0	1	4	5
รวมแนวตั้ง	1	1	5	7

Chi-Square = 3.08 d.f. = 2 (P-Value = 0.2143811)

ค่า P-Value ที่แอตทริบิว "X1" = 3, 4 มีค่ามากที่สุดและมากกว่า 0.05 จึงทำการรวม 2 categories นี้ เข้าไว้ด้วยกัน แล้วทำการจับคู่ทดสอบ Chi-Square หาค่า P-Value อีกครั้ง ดังในตารางที่ 3.9 – 3.11

ตารางที่ 3.9 จับคู่ categories 1 และ 3, 4 ของแอตทริบิว "X1" เปรียบเทียบกับแอตทริบิว "Y"

X1 \ Y	1	2	3	4	รวมแนวนอน
1	23	5	19	4	51
3,4	0	1	1	5	7
รวมแนวตั้ง	23	6	20	9	58

Chi-Square = 20.25577 d.f. = 3 (P-Value = 0.0001502343)

ตารางที่ 3.10 จับคู่ categories 2 และ 3,4 ของแอตทริบิว "X1" เปรียบเทียบกับแอตทริบิว "Y"

X1 \ Y	1	2	3	4	รวมแนวนอน
2	12	2	15	15	42
3,4	0	1	1	5	7
รวมแนวตั้ง	**12**	**3**	**16**	**18**	**49**

Chi-Square = 6.408565 d.f. = 3 (P-Value = 0.09333908)

ตารางที่ 3.11 จับคู่ categories 1 และ 2 ของแอตทริบิว "X1" เปรียบเทียบกับแอตทริบิว "Y"

X1 \ Y	1	2	3	4	รวมแนวนอน
1	23	5	19	4	51
2	12	2	15	13	42
รวมแนวตั้ง	**35**	**7**	**34**	**17**	**93**

Chi-Square = 9.193281 d.f. = 3 (P-Value = 0.02682849)

ค่า P-Value ที่แอตทริบิว "X1" = 2 และ 3, 4 มีค่ามากที่สุดและมากกว่า 0.05 จึงทำการรวม 2 categories นี้ เข้าไว้ด้วยกัน แล้วทำการจับคู่ทดสอบ Chi-Square หาค่า P-Value อีกครั้ง ดังในตารางที่ 3.12

ตารางที่ 3.12 จับคู่ categories 1 และ 2, 3, 4 ของแอตทริบิว "X1" เปรียบเทียบกับแอตทริบิว "Y"

X1 \ Y	1	2	3	4	รวมแนวนอน
1	23	5	19	4	51
2,3,4	12	3	16	18	49
รวมแนวตั้ง	**35**	**8**	**35**	**22**	**100**

Chi-Square = 13.08861 d.f. = 3 (P-Value = 0.004448843)

ค่า P-Value หลังจากการรวม categories ครั้งนี้มีค่าน้อยกว่า 0.05 แล้วจึงหยุดขั้นตอนการ merge categories จากนั้นทำการหาค่าความมีนัยสำคัญของแอตทริบิวตัวอื่นตามขั้นตอนที่ 2 เลือกแอตทริบิว X2 ทำการทดสอบ Chi-Square หาค่า P-Value (ตารางที่ 3.13)

ตารางที่ 3.13 แบ่งค่าของแอตทริบิว "X2" ตาม categories เปรียบเทียบกับแอตทริบิว "Y"

X2 \ Y	1	2	3	4	รวมแนวนอน
0	33	8	34	22	97
1	1	0	1	0	2
2	1	0	0	0	1
รวมแนวตั้ง	35	8	35	22	100

Chi-Square = 2.76778 d.f. = 6 (P-Value = 0.8372577)

จับคู่ categories ของแอตทริบิว "X2" ทดสอบ Chi-Square หาค่า P-Value (ตารางที่ 3.14 – ตารางที่ 3.16)

ตารางที่ 3.14 จับคู่ categories 0 และ 1 ของแอตทริบิว "X2" เปรียบเทียบกับแอตทริบิว "Y"

X2 \ Y	1	2	3	4	รวมแนวนอน
0	33	8	34	22	97
1	1	0	1	0	2
รวมแนวตั้ง	34	8	35	22	99

Chi-Square = 0.8881097 d.f. = 3 (P-VALUE = 0.8282962)

ตารางที่ 3.15 จับคู่ categories 0 และ 2 ของแอตทริบิว "X2" เปรียบเทียบกับแอตทริบิว "Y"

X2 \ Y	1	2	3	4	รวมแนวนอน
0	33	8	34	22	97
2	1	0	0	0	1
รวมแนวตั้ง	34	8	34	22	98

Chi-Square = 1.901759 d.f. = 3 (P-Value = 0.5930452)

ตารางที่ 3.16 จับคู่ categories 1 และ 2 ของแอตทริบิว "X2" เปรียบเทียบกับแอตทริบิว "Y"

X2 \ Y	1	3	รวมแนวนอน
1	1	1	2
2	1	0	1
รวมแนวตั้ง	2	1	3

Chi-Square = 0.75 d.f. = 1 (P-Value = 0.3864762)

ค่า P-Value ที่แอตทริบิว "X2" = 0, 1 มีค่ามากที่สุดและมากกว่า 0.05 จึงทำการรวม 2 categories นี้ เข้าไว้ด้วยกัน แล้วทำการจับคู่ categories ทดสอบ Chi-Square หาค่า P-Value อีกครั้ง (ตารางที่ 3.17)

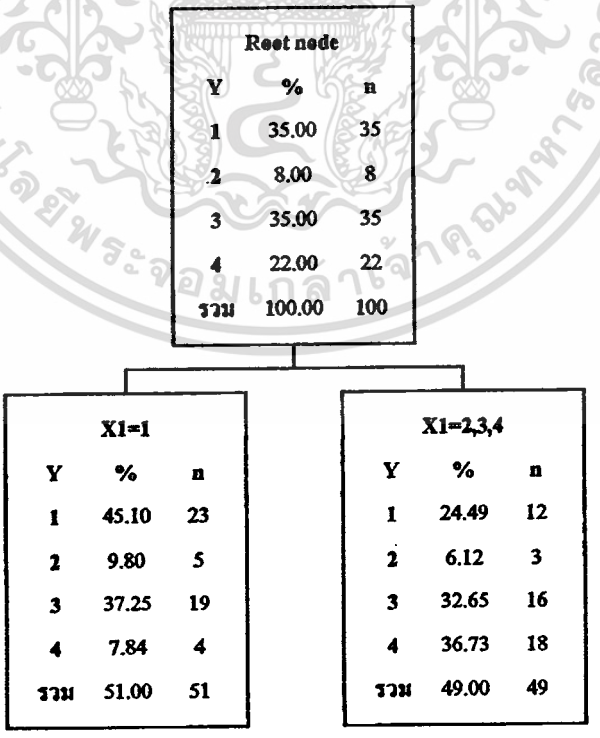
ตารางที่ 3.17 จับคู่ categories 0, 1 และ 2 ของแอตทริบิว "X2" เปรียบเทียบกับแอตทริบิว "Y"

X2 \ Y	1	2	3	4	รวมแนวนอน
0 , 1	34	8	35	22	99
2	1	0	0	0	1
รวมแนวตั้ง	35	8	35	22	100

Chi-Square = 1.875902 d.f. = 3 (P-Value = 0.5985587)

ในกรณีนี้ ค่า P-Value ถูกแก้ไขให้เท่ากับ 1 เนื่องจากค่า P-Value เดิม เมื่อนำมาคูณกับ d.f. แล้ว มีค่ามากกว่า 1 (0.5985587 x 3 = 1.7956761)

เปรียบเทียบความสามารถในการคัดแยกของแอตทริบิวทั้ง 2 พบว่าแอตทริบิว X1 มีความสามารถในการคัดแยกมากกว่าแอตทริบิว X2 (ค่า P-Value ของ X1 น้อยกว่า X2) จึงกำหนดให้ใช้แอตทริบิว X1 เป็นตัวคัดแยกในโนดรากโดยแตกกิ่งแขนง 2 กิ่งแสดงผลการทดสอบของแอตทริบิว X1 คือ X1 = 1 และ X1 = 2, 3, 4 ดังรูปที่ 3.1



รูปที่ 3.1 ข้อมูลตัวอย่างถูกแบ่งตาม category

จากนั้นนำข้อมูลที่ถูกคัดแยกจากโนดรากมาสร้างตัวคัดแยกในโนดตัวต่อไปและทำซ้ำไปเรื่อยๆ จนกว่าข้อมูลทั้งหมดถูกคัดแยกอย่างสมบูรณ์แล้ว (ค่าของแอตทริบิว Y ในแต่ละโนดอยู่ใน categories เดียวกัน) หรือแอตทริบิวทั้งหมดถูกใช้เป็นตัวคัดแยก กระบวนการการสร้างแขนงการตัดสินใจจึงสิ้นสุด

3.3 ประสิทธิภาพของอัลกอริทึม CHAID

ประสิทธิภาพของระบบดาต้าไมน์นิ่งนั้น เกิดจากปัจจัยหลายอย่างประกอบกัน ได้แก่ ข้อมูลที่นำมาสร้างแบบจำลอง การเตรียมข้อมูล(Data Preparation) ประเภทและชนิดของอัลกอริทึมที่ใช้ ความเข้ากันได้ของลักษณะข้อมูลกับอัลกอริทึมที่ใช้ ความสามารถของโปรแกรม และความสามารถของผู้วิเคราะห์ผล เป็นต้น

จากการศึกษา พบว่าการสร้างแบบจำลองที่มีโครงสร้างเป็นแขนงการตัดสินใจ (Decision Tree) โดยใช้อัลกอริทึม CHAID นี้ มีข้อดีและข้อจำกัดดังนี้

3.3.1 จุดเด่นของอัลกอริทึม CHAID

- ผลที่ได้จากอัลกอริทึม CHAID คือแบบจำลองที่มีโครงสร้างเป็นแขนงการตัดสินใจ และกฎเงื่อนไข ซึ่งผู้ใช้สามารถนำไปวิเคราะห์ต่อยอดได้ง่าย
- อัลกอริทึม CHAID มีความสามารถในการยุบรวม (Merge) Categories ที่อยู่ในแอตทริบิว ซึ่งจะช่วยลดการแตก node ที่ไม่จำเป็นลง (มีเฉพาะโนดที่มีความหมายเพียงพอเท่านั้น) แขนงการตัดสินใจที่ได้จึงมีขนาดไม่ใหญ่เกินความจำเป็น สามารถที่จะรองรับข้อมูลที่ค่าของแอตทริบิวหรือตัวแปรมีรายละเอียดมาก (มีหลาย categories)

3.3.2 ข้อจำกัดของอัลกอริทึม CHAID

- ถ้าข้อมูลที่นำมาสร้างแบบจำลองมีจำนวน Categories น้อยเกินไป ระบบจะไม่สามารถนำความสามารถของอัลกอริทึม CHAID มาใช้ได้เต็มที่ เนื่องจากอัลกอริทึมไม่สามารถยุบรวม categories ได้
- ข้อจำกัดของการทดสอบ Chi-Square ในการหาค่าความสัมพันธ์คือ จำนวนข้อมูลในแต่ละเซลไม่ควรต่ำกว่า 5 ระเบียน
- การยุบรวม (Merge) categories ของอัลกอริทึม CHAID มาจากทฤษฎีทางสถิติ คือ การวิเคราะห์ปัจจัย (Factor Analysis) คือการรวม categories ที่มีนัยสำคัญต่ำเข้าด้วยกันเพื่อให้มีค่านัยสำคัญรวมสูงขึ้น แต่ไม่สามารถสรุปได้ว่า categories ที่ถูกยุบรวมกันนั้น มีความสัมพันธ์หรือเกี่ยวข้องกันจริงๆ

บทที่ 4
การวิเคราะห์และออกแบบโปรแกรม

4.1 การวิเคราะห์และออกแบบโปรแกรม

ระบบการจัดหมวดหมู่ข้อมูลนี้ โปรแกรมจะทำการติดต่อกับ Relational Database Management Systems คือ Microsoft SQL Server 2005 เพื่อจะทำการเลือก ฐานข้อมูล ตาราง และแอตทริบิวต่างๆ ที่ต้องการมาใช้ในการคำนวณสร้างแบบจำลองซึ่งมีลักษณะเป็นแขนงการตัดสินใจ (Decision Tree) โดยใช้อัลกอริทึม CHAID

ก่อนที่จะนำข้อมูลในฐานข้อมูลมาสร้างแบบจำลองนั้น ระบบจำเป็นต้องจัดเตรียมข้อมูลให้อยู่ในรูปแบบที่เหมาะสม (Data Preparation) ก่อน ซึ่งประกอบด้วย 3 ขั้นตอนคือ 1. การเลือกข้อมูล (Data Selection) 2. การกำจัดค่าว่าง (Data Cleaning) และ 3. การแปลงข้อมูล (Data Transformation) หลังจากนั้น จึงนำข้อมูลดังกล่าวมาสร้างแบบจำลอง

ในการสร้างแบบจำลองที่มีโครงสร้างเป็นแขนงการตัดสินใจ โปรแกรมจะคำนวณค่าความสามารถในการคัดแยกของตัวแปรต้นหรือแอตทริบิวแต่ละตัวเพื่อสร้างเป็นตัวคัดแยกของโนดต่างๆ ใน Tree ซึ่งระบบต้องบันทึกข้อมูลเหล่านี้ไว้ จึงเลือกโครงสร้างข้อมูลแบบ Structure ที่มีสมาชิกเป็น ArrayList มาใช้การจัดเก็บข้อมูลของโนด เพื่อให้มีความยืดหยุ่นและเหมาะสม แล้วใช้อัลกอริทึมในสร้างการสร้าง Tree แบบ First Child-Next Sibling

เมื่อสร้างแบบจำลองแล้ว จะมีการทดสอบแบบจำลองแบบจำลอง โดยทดลองจัดหมวดหมู่ข้อมูลที่เหลือ(ข้อมูลที่เราทราบคลาสหรือหมวดหมู่แล้ว) เปรียบเทียบค่าและแสดงผลให้ผู้ใช้ทราบ

เมื่อได้แบบจำลองที่มีความถูกต้องแล้ว เราสามารถนำแบบจำลองนี้ไปใช้จัดหมวดหมู่ข้อมูลใหม่ หรือข้อมูลที่ยังไม่รู้หมวดหมู่

นอกจากนี้ เราสามารถนำแบบจำลองมาวิเคราะห์เพื่อตีความหมายเป็นสารสนเทศหรือความรู้ใหม่ๆ ได้

4.2 Process Model

4.2.1 Use Case Diagram

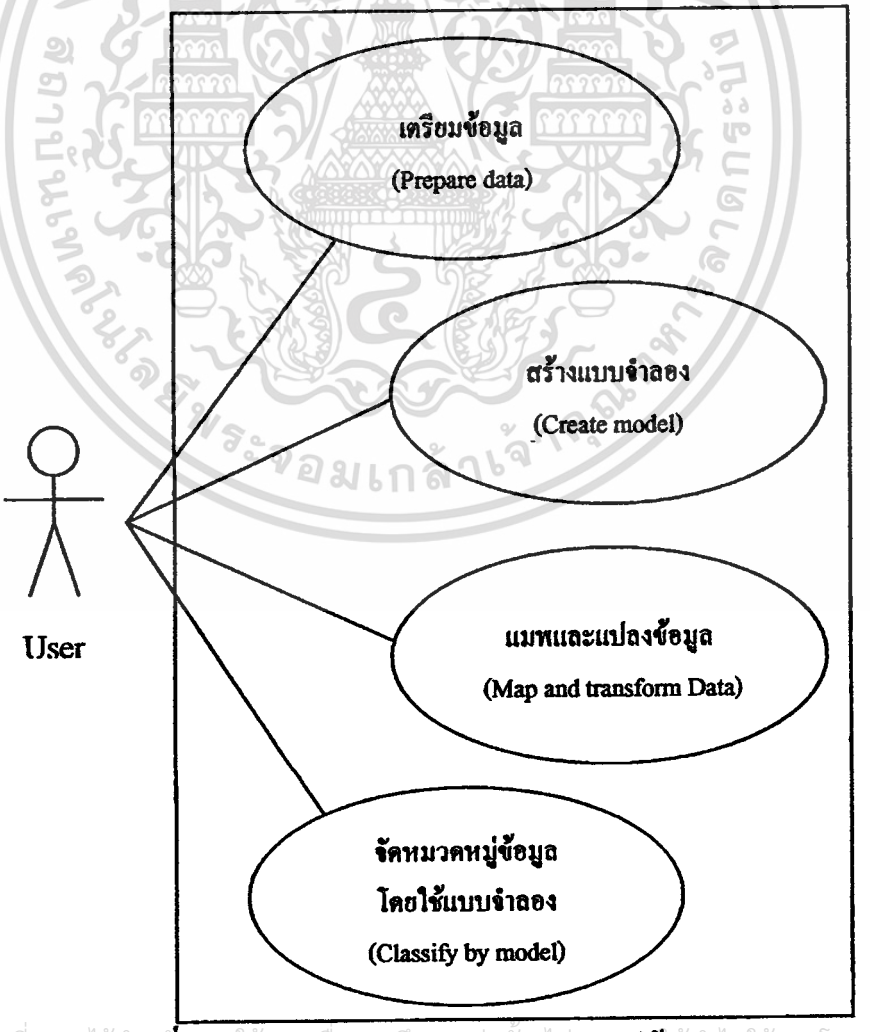
Use Case Diagram แสดงถึงระบบการทำงานว่าทำอะไรได้บ้าง มีการทำงานหลักๆ อะไร ผู้ใช้ใช้อะไรได้บ้าง ซึ่งตัวแปรแกรมได้แบ่งฟังก์ชันการทำงานออกเป็น 4 ส่วนหลักคือ 1) การเตรียมข้อมูลเพื่อใช้สร้างแบบจำลอง 2) การสร้างแบบจำลองพยากรณ์ และ 3) แมพข้อมูลและแปลงข้อมูล 4) การจัดหมวดหมู่ข้อมูลด้วยแบบจำลองพยากรณ์ ดังรูปที่ 4.1

1) เตรียมข้อมูลเพื่อใช้สร้างแบบจำลอง โดยระบบจะทำการติดต่อกับ RDBMS คือ Microsoft SQL Server 2005 เพื่อทำการเลือกข้อมูลที่ต้องการสำหรับการสร้างแขนงการตัดสินใจ รวมถึงการกำจัดค่าว่างและการแปลงข้อมูล เพื่อให้ได้ชุดข้อมูลที่มีความเหมาะสมในการสร้างแบบจำลองพยากรณ์

2) การสร้างแบบจำลองพยากรณ์ ระบบจะนำชุดข้อมูลที่เตรียมไว้มาสร้างแบบจำลองตามอัลกอริทึมที่บรรจุอยู่ภายใน แล้วทำการบันทึกข้อมูลโครงสร้างของแบบจำลองที่สร้างขึ้นลงในฐานข้อมูล หลังจากนั้นจะมีการทดสอบความถูกต้องในการจัดหมวดหมู่ข้อมูลของแบบจำลอง

3) แมพข้อมูลและแปลงข้อมูล เป็นการจัดรูปแบบของข้อมูลใหม่ให้ตรงกันกับข้อมูลในแบบจำลอง เพื่อให้ข้อมูลใหม่สามารถเข้ากันกับแบบจำลองได้

4) การจัดหมวดหมู่ข้อมูลด้วยแบบจำลองพยากรณ์ เป็นการนำข้อมูลใหม่ที่จัดรูปแบบแล้ว แต่ยังไม่รู้หมวดหมู่มาจัดหมวดหมู่ด้วยแบบจำลอง ข้อมูลเหล่านี้จะถูกทดสอบด้วยแขนงการตัดสินใจ (ภายในแบบจำลอง) ซึ่งจะได้ผลลัพธ์เป็นหมวดหมู่ของข้อมูลนั้น



รูปที่ 4.1 Use Case ของโปรแกรมจัดหมวดหมู่ข้อมูล

Use Case Description (ตารางที่ 4.1 – ตารางที่ 4.4)

ตารางที่ 4.1 Use Case Description ของ Use Case เตรียมข้อมูล

Use Case	1. เตรียมข้อมูล (Prepare data)
Brief Description	การเตรียมข้อมูลสำหรับการสร้างแบบจำลอง ประกอบด้วยการเข้าถึงฐานข้อมูล การเลือกข้อมูล การกำจัดค่าว่าง และการแปลงข้อมูล
Actor	ผู้ใช้ (User)
Pre – condition	-
Post – condition	- ชุดข้อมูลที่ผ่านการปรับปรุงแล้ว มีความเหมาะสมที่จะนำไปสร้างแบบจำลอง
Primary scenario	1) ผู้ใช้กรอกชื่อเซิร์ฟเวอร์และชื่อฐานข้อมูล 2) ระบบทำการตรวจสอบการเชื่อมต่อฐานข้อมูล 3) ระบบแสดงตารางในฐานข้อมูล 4) ผู้ใช้เลือกตารางภายในฐานข้อมูลที่เลือกไว้ 5) ระบบแสดงชื่อของฐานข้อมูลและตารางที่เลือก 6) ระบบทำการดึงข้อมูลจากฐานข้อมูล และแสดงชื่อเซิร์ฟเวอร์ ชื่อฐานข้อมูล และชื่อตาราง 7) ระบบแสดงชื่อของแอตทริบิวจากตารางที่เลือก 8) ผู้ใช้เลือกแอตทริบิวจากตารางที่ปรากฏ เพื่อทำ Data Mining 9) ระบบแสดงรายละเอียดของแอตทริบิวต่างๆ ภายในตาราง 10) ผู้ใช้เลือกแอตทริบิวในการทำ Data Cleaning (กำจัดค่าว่าง) 11) ระบบแสดงข้อมูลภายในแอตทริบิวทั้งหมด ได้แก่ รายการข้อมูล, ชื่อแอตทริบิว, ชนิดข้อมูล, จำนวนระเบียน, ค่าสูงสุด, ค่าต่ำสุด และค่าเฉลี่ย 12) ผู้ใช้เลือกการกำจัดค่าว่างแบบอัตโนมัติ 13) ระบบทำการกำจัดค่าว่างแบบอัตโนมัติ 14) ระบบแสดงข้อมูลทั้งหมดจากการกำจัดค่าว่าง 15) ระบบแสดงหน้าต่างสำหรับการทำ Data Transformation 16) ผู้ใช้เลือกแอตทริบิว สำหรับการทำ Data Transformation 17) ระบบทำการแปลงตัวเลขเป็นตัวอักษร 18) ระบบแสดงชื่อแอตทริบิว, ชนิดข้อมูล, จำนวนเรคอร์ด, ค่าสูงสุด, ค่าเฉลี่ย, ค่าต่ำสุด, และจำนวนกลุ่ม (2 categories และ3 categories) 19) ระบบแสดงข้อมูลทั้งหมดที่ได้จากการแปลงข้อมูล

ตารางที่ 4.1 (ต่อ)

Alternative flow	2a) ระบบไม่สามารถเชื่อมต่อฐานข้อมูลได้ : กลับไปข้อที่ 1 12a) ผู้ใช้เลือกทำการแก้ไขข้อมูลด้วยตนเอง 12.1a) ระบบแสดงข้อมูลค่าว่างของแอตทริบิวเพื่อทำการแก้ไขข้อมูล ได้แก่ ข้อมูลทั่วไป (ลบเรคอร์ดที่มีค่าว่าง และใส่ค่า Unknown) และ ข้อมูลตัวเลข (ลบเรคอร์ดที่มีค่าว่าง, ใส่ค่าสูงสุด, ใส่ค่าต่ำสุด และใส่ค่าเฉลี่ย) 18a) ผู้ใช้เลือกจำนวนกลุ่ม 2 กลุ่มในการแปลงข้อมูล และระบบจะแสดงค่าของกลุ่ม(Low, High) 18b) ผู้ใช้เลือกจำนวนกลุ่ม 3 กลุ่มในการแปลงข้อมูล และระบบจะแสดงค่าของกลุ่ม(Low, Medium, High)

ตารางที่ 4.2 Use Case Description ของ Use Case สร้างแบบจำลอง

Use Case	2. สร้างแบบจำลองพยากรณ์ (Create model)
Brief Description	การสร้างแบบจำลองพยากรณ์จากชุดข้อมูลที่เตรียมไว้ โดยใช้อัลกอริทึม CHAID
Actor	ผู้ใช้ (User)
Pre – condition	- เตรียมชุดข้อมูลเรียบร้อยแล้ว
Post – condition	- ได้แบบจำลองเพื่อใช้ในการพยากรณ์
Primary scenario	1) ระบบแสดงหน้าต่างโครงสร้างต้นไม้ 2) ผู้ใช้ทำการเลือกแอตทริบิวที่ใช้เป็นแอตทริบิวเป้าหมาย 3) ผู้ใช้ทำการกำหนดค่าความคลาดเคลื่อนในการ merge categories 4) ผู้ใช้กำหนดจำนวนระเบียนขั้นต่ำในการแตกโนดลูกได้ 5) ระบบแสดงโครงสร้างต้นไม้ รายละเอียดของโนด ข้อมูลความคลาดเคลื่อนของโครงสร้างต้นไม้ที่ได้จากการทดสอบโดยคิดเป็นร้อยละ และรายละเอียดของการแปลงข้อมูลจากตัวเลขเป็นตัวอักษร 6) ผู้ใช้ตั้งชื่อและบันทึกแบบจำลอง
Alternative flow	6a) ผู้ใช้ไม่บันทึกแบบจำลอง

ตารางที่ 4.3 Use Case Description ของ Use Case แมพข้อมูลและแปลงข้อมูล

Use Case	3.แมพข้อมูลและแปลงข้อมูล (Map and transform data)
Brief Description	จัดรูปแบบข้อมูลใหม่ก่อนที่จะนำไปจัดหมวดหมู่โดยใช้แบบจำลองพยากรณ์
Actor	ผู้ใช้ (User)
Pre – condition	- แบบจำลองพยากรณ์สำหรับจัดหมวดหมู่ข้อมูล - ข้อมูลใหม่ที่จะนำมาจัดหมวดหมู่
Post – condition	- ข้อมูลใหม่ที่มีรูปแบบเดียวกันกับแบบจำลองพยากรณ์ได้
Primary scenario	1) เลือกแบบจำลองที่ได้ทำสร้างและบันทึกไว้ 2) ระบบแสดงข้อมูลของแขนงการตัดสินใจ (ร้อยละความผิดพลาดของแขนงการตัดสินใจและโครงสร้างของแขนงการตัดสินใจ) 3) ผู้ใช้กรอกชื่อเซิร์ฟเวอร์และชื่อฐานข้อมูลใหม่ 4) ระบบทำการตรวจสอบการเชื่อมต่อฐานข้อมูล 5) ผู้ใช้ทำการแมพข้อมูลโดยการเลือกตารางและแอตทริบิวที่ต้องการ 6) ระบบทำการแมพข้อมูลและทำการลบค่าว่างของข้อมูล 7) ระบบทำการแปลงข้อมูลที่เป็นตัวเลขให้เป็นตัวอักษร โดยการแบ่งกลุ่มเป็น 2 หรือ 3 categories ในลักษณะเดียวกันกับแบบจำลอง)
Alternative flow	4a) ระบบไม่สามารถเชื่อมต่อฐานข้อมูลได้ : กลับไปข้อที่ 3

ตารางที่ 4.4 Use Case Description ของ Use Case จัดหมวดหมู่ข้อมูลโดยใช้แบบจำลอง

Use Case	4. จัดหมวดหมู่ข้อมูลโดยใช้แบบจำลอง (Classify by model)
Brief Description	จัดหมวดหมู่ข้อมูลใหม่หรือข้อมูลที่ยังไม่รู้หมวดหมู่ด้วยแบบจำลองพยากรณ์
Actor	ผู้ใช้ (User)
Pre – condition	- แบบจำลองพยากรณ์สำหรับจัดหมวดหมู่ข้อมูล
Post – condition	- หมวดหมู่ของข้อมูลใหม่
Primary scenario	1) ระบบแสดงข้อมูลทั้งหมดของตาราง 2) ผู้ใช้เลือก "ตรวจสอบความเข้ากันได้ของข้อมูลใหม่กับแบบจำลอง" 3) ระบบตรวจสอบความเข้ากันของข้อมูลใหม่กับแบบจำลอง 4) ระบบทำการจัดหมวดหมู่ข้อมูลและแสดงผลลัพธ์ของการจัดหมวดหมู่พร้อมกับร้อยละความผิดพลาดของแขนงการตัดสินใจ
Alternative flow	3a) ข้อมูลใหม่กับแบบจำลองไม่สามารถเข้ากันได้ : จบการทำงาน

4.2.2 **Activity Diagram**

1) Activity Diagram ของการเตรียมข้อมูล (Use case เตรียมข้อมูล)

กระบวนการการเตรียมข้อมูลเริ่มจากติดต่อฐานข้อมูลที่เก็บข้อมูลสำหรับทำคาต้าไมน์นิ่ง จากนั้นทำการเลือกข้อมูลที่จะใช้สร้างแบบจำลอง โดยเลือกตารางข้อมูลและแอตทริบิว แล้วทำการกำจัดค่าว่าง ซึ่งทำได้ 5 วิธี ได้แก่ 1) ลบระเบียนที่มีค่าว่าง 2) ใส่ค่าเฉลี่ย 3) ใส่ค่าสูงสุด 4) ใส่ค่าต่ำสุด และ 5) ใส่ค่า unknown

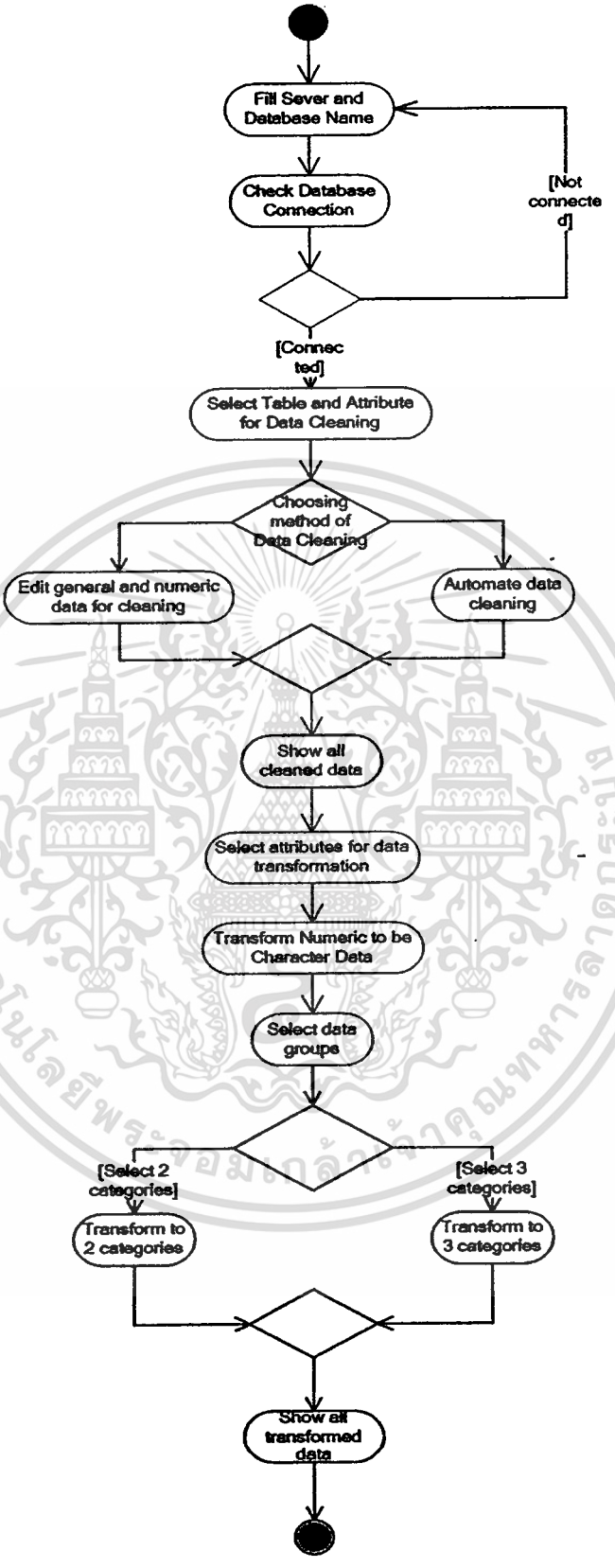
ด้วยเหตุที่อัลกอริทึม CHAID รองรับข้อมูลที่มีลักษณะเป็น category เท่านั้น เมื่อผู้ใช้ทำการกำจัดค่าว่างเสร็จแล้ว จึงต้องทำการแปลงข้อมูลที่มีลักษณะเป็นตัวเลขหรือค่าต่อเนื่อง ให้เป็น category โดยโปรแกรมมีทางเลือกในการแปลงค่าอยู่ 2 ทางคือ แปลงเป็น 2 categories (High, Low) และ 3 categories (High, Medium, Low) ดังรูปที่ 4.2

2) Activity Diagram ของการสร้างแบบจำลองพยากรณ์ (Use Case สร้างแบบจำลอง)

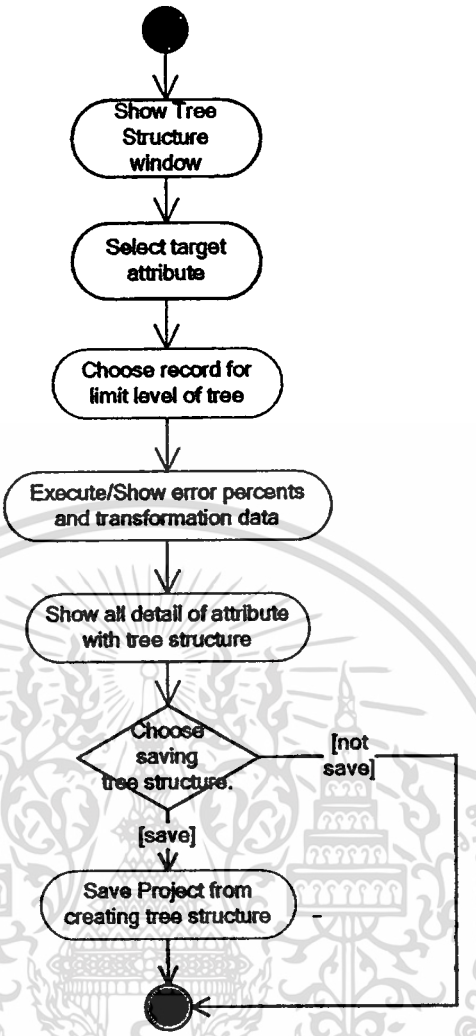
ในการสร้างแบบจำลองพยากรณ์ โปรแกรมจะใช้ข้อมูลที่ผ่านกระบวนการเตรียมข้อมูล (Data Preparation) แล้ว โดยทำการแบ่งข้อมูลนั้นเป็น 2 ชุด คือ 1) ชุดข้อมูลตัวอย่าง (Training data) มีจำนวนคิดเป็น 80 เปอร์เซ็นของข้อมูลทั้งหมด ข้อมูลตัวอย่างนี้ เป็นข้อมูลที่ใช้สร้างแขนงการตัดสินใจ และ 2) ชุดข้อมูลทดสอบ (Testing data) เป็นข้อมูลที่ใช้เพื่อทดสอบความถูกต้องของแขนงการตัดสินใจ มีจำนวนคิดเป็น 20 เปอร์เซ็น ซึ่งโปรแกรมจะทำการสุ่มเลือกข้อมูลทั้ง 2 ชุดนี้โดยอัตโนมัติ

กระบวนการสร้างแบบจำลองพยากรณ์เริ่มจาก 1) ผู้ใช้เลือกแอตทริบิวเป้าหมาย 2) ผู้ใช้ทำการกำหนดค่าความผิดพลาด (ความยืดหยุ่น) ในการ merge categories และ 3) กำหนดจำนวนระเบียนขั้นต่ำในการแตกโนดของแขนงการตัดสินใจ

เมื่อโปรแกรมทำการสร้างแขนงการตัดสินใจเสร็จแล้ว จะแสดงโครงสร้าง tree และค่าความผิดพลาดของแขนงการตัดสินใจ ดังรูปที่ 4.3



รูปที่ 4.2 Activity Diagram ของการเตรียมข้อมูล

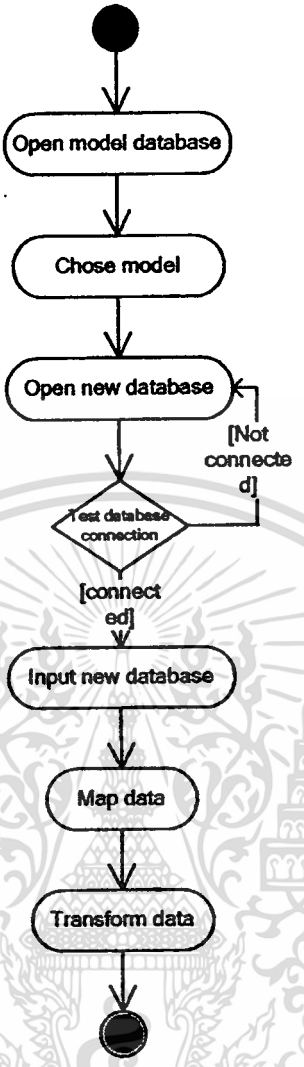


รูปที่ 4.3 Activity Diagram ของการสร้างแบบจำลองพยากรณ์

3) Activity Diagram ของการแมพข้อมูลและแปลงข้อมูล (Use Case แมพข้อมูลและแปลงข้อมูล)

ในการจัดหมวดหมู่ข้อมูลโดยใช้แบบจำลองพยากรณ์ ข้อมูลที่จะนำมาจัดหมวดหมู่จะต้องนำมาจับคู่แอตทริบิวให้ตรงกันกับแอตทริบิวของแบบจำลอง (Map data) และแปลงข้อมูลให้มีลักษณะเดียวกันกับแบบจำลอง

การแมพข้อมูลและแปลงข้อมูลมีขั้นตอนการทำงาน ... ขั้นตอน ได้แก่ 1) เลือกแบบจำลองพยากรณ์ 2) ติดต่อฐานข้อมูล 3) เลือกตารางข้อมูล 4) จับคู่แอตทริบิวของข้อมูลกับแอตทริบิวของแบบจำลอง 5) แปลงข้อมูลให้มีรูปแบบเดียวกันกับแบบจำลอง ดังรูปที่ 4.4

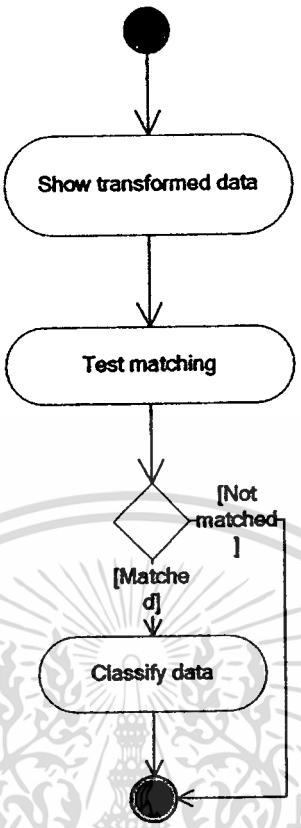


รูปที่ 4.4 Activity Diagram ของการแมพข้อมูลและแปลงข้อมูล

4) Activity Diagram ของการ จัดหมวดหมู่ข้อมูลโดยใช้แบบจำลอง (Use Case จัดหมวดหมู่ข้อมูลด้วยแบบจำลอง)

เมื่อข้อมูลใหม่ที่จะนำมาจัดหมวดหมู่ผ่านการแมพและจัดรูปแบบแล้ว โปรแกรมจะทำการตรวจสอบความเข้ากันได้ของข้อมูลใหม่กับแบบจำลอง

ถ้าข้อมูลใหม่สามารถเข้ากับแบบจำลองได้ โปรแกรมจะทำการจัดหมวดหมู่ข้อมูลและแสดงผล แต่ถ้าข้อมูลใหม่ไม่สามารถเข้ากับแบบจำลองได้ โปรแกรมจะทำการแจ้งเตือนและจบการทำงาน ดังรูปที่ 4.5

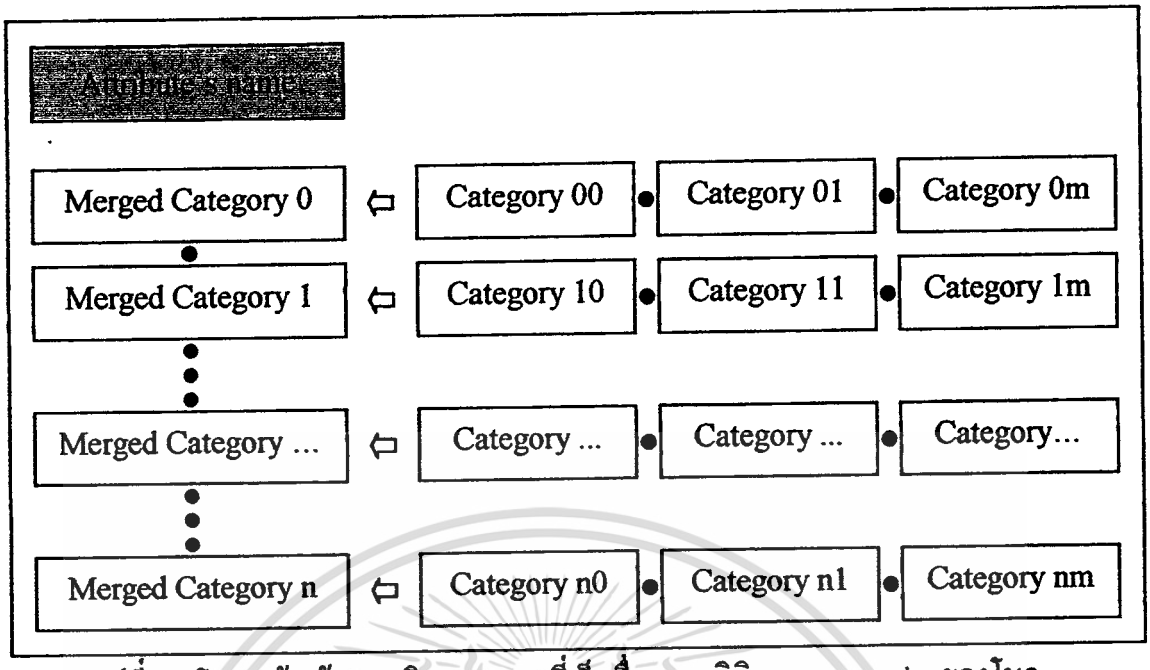


รูปที่ 4.5 Activity Diagram ของการการจัดหมวดหมู่ข้อมูล (Classification) โดยใช้แบบจำลองพยากรณ์

4.3 โครงสร้างข้อมูล Data Structure

4.3.1 โครงสร้างข้อมูลในการแตกโนดตาม Categories

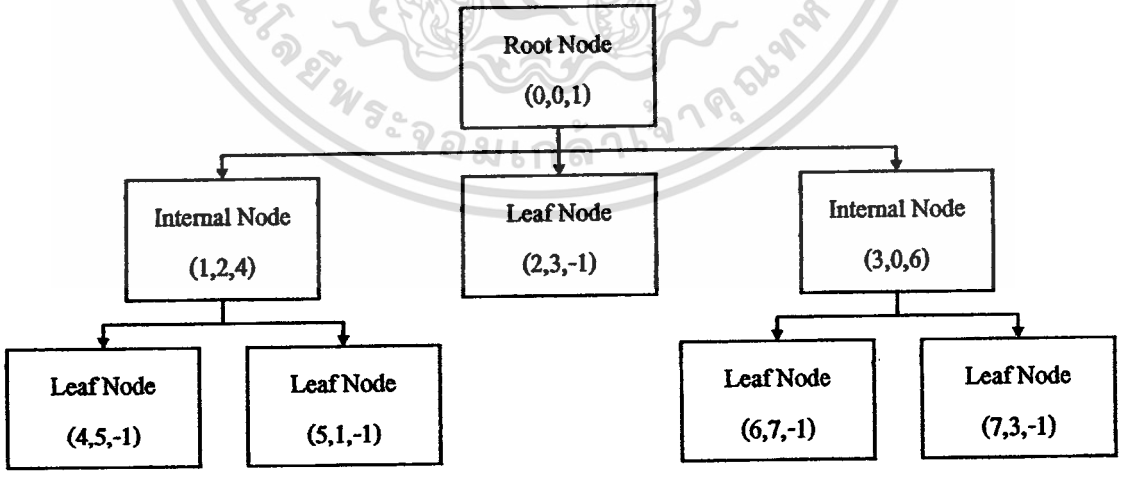
เนื่องจากอัลกอริทึม CHAID มีการยุบรวม categories ของแอตทริบิว การออกแบบโครงสร้างข้อมูลจำต้องเน้นความยืดหยุ่นเป็นหลัก จึงใช้โครงสร้างข้อมูลแบบ Structure ที่ประกอบด้วยตัวแปร String เก็บชื่อแอตทริบิวที่เป็นตัวคัดแยก และ ArrayList เก็บ categories ที่ถูก Merged สมาชิกของ ArrayList แต่ละตัวคือ ArrayList ที่เก็บชื่อ categories ย่อยหรือ categories ดั้งเดิมไว้ ดังรูปที่ 4.6



รูปที่ 4.6 โครงสร้างข้อมูลชนิด structure ที่เก็บชื่อแอตทริบิวและ categories ของโนด

4.3.2 โครงสร้างข้อมูลของแขนงการตัดสินใจ

เนื่องจากแขนงการตัดสินใจที่ถูกสร้างโดยอัลกอริทึม CHAID มีโครงสร้างเป็นแบบ Non-Binary Tree คือ โนดแม่ 1 โนด สามารถแตกโนดลูกได้มากกว่า 2 โนด จึงได้เลือกเทคนิคการสร้าง tree แบบ First Child-Next Sibling ซึ่งมีความยืดหยุ่นสูง สามารถรองรับ Non-Binary Tree ได้ ดังตัวอย่างในรูปที่ 4.7



รูปที่ 4.7 ตัวอย่างโครงสร้างของแขนงการตัดสินใจ

4.4 ฐานข้อมูลของโปรแกรม

การทำงานของโปรแกรมจัดหมวดหมู่ข้อมูลนี้ มีการบันทึกและเรียกใช้ข้อมูลในฐานข้อมูล MS. SQL Server 2005 ชื่อ TreeIndex ซึ่งประกอบด้วยตารางข้อมูลทั้งหมด 4 ตาราง ได้แก่

1) ตารางข้อมูล TreeIndex เก็บรายละเอียดของโนดต่างๆ ในแขนงการตัดสินใจ (ตารางที่ 4.5)

ตารางที่ 4.5 ชื่อและชนิดของแอตทริบิวในตารางข้อมูล TreeIndex

ชื่อแอตทริบิว	ชนิดข้อมูล	รายละเอียด	คีย์	อ้างถึงตาราง
ProjectID	int	รหัสแบบจำลอง	F.K.	ProjectIndex
NodeID	int	รหัสโนด	P.K.	
Node_Name	varchar(500)	ชื่อโนด		
Node_Parent	int	รหัสโนดแม่	F.K.	TreeIndex
Node_Command	varchar(2047)	คำสั่ง SQL Command		
Node_Friend	int	รหัสโนดเพื่อน(ตัวถัดไป)	F.K.	TreeIndex
Node_Next	int	รหัสโนดลูก(ตัวแรก)	F.K.	TreeIndex
Node_Status	int	สถานะของโนด		
Node_Label	text	ป้ายชื่อของโนด (ระบุคลาสของโนด)		
NodeUsedAttribute	nvarchar(2047)	ชื่อแอตทริบิวที่ถูกใช้ไปแล้ว		

2) ตารางข้อมูล ProjectIndex เก็บรหัส ชื่อ และข้อมูลต่างๆ ของแบบจำลอง (ตารางที่ 4.6)

ตาราง 4.6 ชื่อและชนิดของแอตทริบิวในตารางข้อมูล ProjectIndex

ชื่อแอตทริบิว	ชนิดข้อมูล	รายละเอียด	คีย์	อ้างถึงตาราง
ProjectID	nchar(10)	รหัสแบบจำลอง	P.K.	
ProjectName	text	ชื่อแบบจำลอง		
Error	float	ค่าความคลาดเคลื่อนของแบบจำลอง		
Detailtree	text	รายละเอียดของแบบจำลอง		

3) ตารางข้อมูล Field_datatree เก็บข้อมูลการแปลงข้อมูลก่อนการสร้างแบบจำลอง (ตารางที่ 4.7)

ตารางที่ 4.7 ชื่อและชนิดของแอตทริบิวในตารางข้อมูล Field_datatree

ชื่อแอตทริบิว	ชนิดข้อมูล	รายละเอียด	คีย์	อ้างถึงตาราง
ProjectID	int	รหัสแบบจำลอง	F.K.	ProjectIndex
Name_possible	nvarchar(255)	ชื่อแอตทริบิว		
Insert_value	nvarchar(5)	ตัวอักษรตัวแรกที่เติมไว้หน้าค่า categories เช่น H-M-L		
Num_division	float	จำนวนช่วงค่าที่มีการแปลงข้อมูล		
Val1	float	ตัวตัดค่า "ต่ำ"		
Val2	float	ตัวตัดค่า "สูง"		

4) ตารางข้อมูล refer_datatree เก็บชื่อและค่าของแอตทริบิวของข้อมูลที่ใช้สร้างแบบจำลอง (ตารางที่ 4.8)

ตารางที่ 4.8 ชื่อและชนิดของแอตทริบิวในตารางข้อมูล refer_datatree

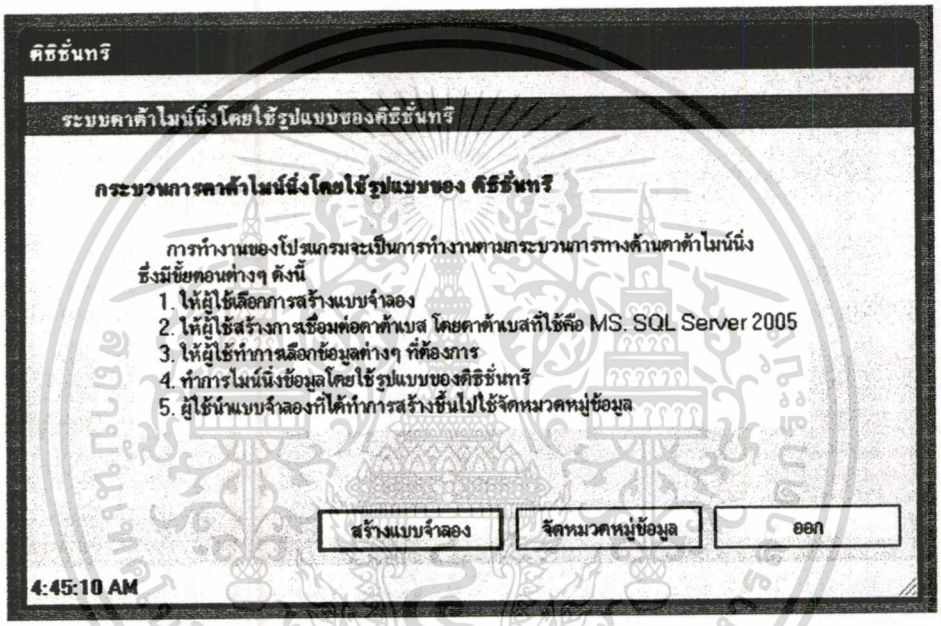
ชื่อแอตทริบิว	ชนิดข้อมูล	รายละเอียด	คีย์	อ้างถึงตาราง
ProjectID	int	รหัสแบบจำลอง	F.K.	ProjectIndex
Name_field_ref	nvarchar(255)	ชื่อแอตทริบิว (ข้อมูลที่นำมาสร้างแบบจำลอง)		
Value_ref	nvarchar(255)	ค่าของแอตทริบิว (ที่นำมาสร้างแบบจำลอง)		

บทที่ 5
การประยุกต์ใช้โปรแกรม

5.1 เปิดโปรแกรมและติดต่อกับฐานข้อมูล

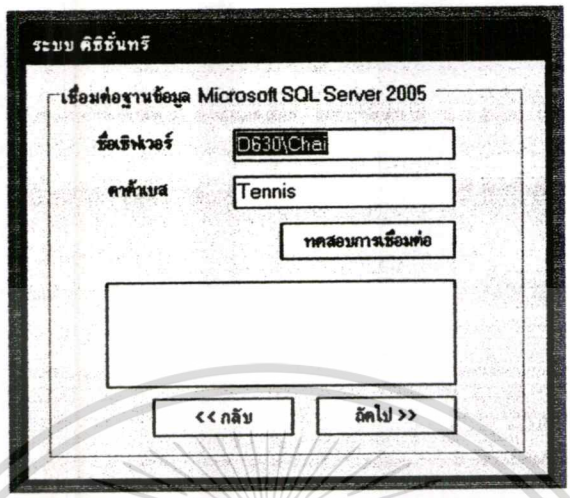
ขั้นตอนแรกก่อนที่จะเข้าสู่กระบวนการทำคาต้าไมนิ่งต้องทำการติดต่อกับฐานข้อมูล โดยฐานข้อมูลที่จะทำการติดต่อด้วยคือ Microsoft SQL Server 2005

1) ผู้ใช้เปิดโปรแกรม และเลือกการสร้างแบบจำลอง (รูปที่ 5.1)



รูปที่ 5.1 ผู้ใช้เลือกการสร้างแบบจำลอง

2) ผู้ใช้ทำการกรอกข้อมูลการติดต่อฐานข้อมูล ได้แก่ ชื่อเซิฟเวอร์ และ ชื่อดาต้าเบส (รูปที่ 5.2)



รูปที่ 5.2 หน้าต่างสำหรับผู้ใช้กรอกชื่อเซิฟเวอร์ และชื่อฐานข้อมูล

3) กดปุ่ม Test Connection เพื่อทดสอบการเชื่อมต่อกับฐานข้อมูล (รูปที่ 5.3)

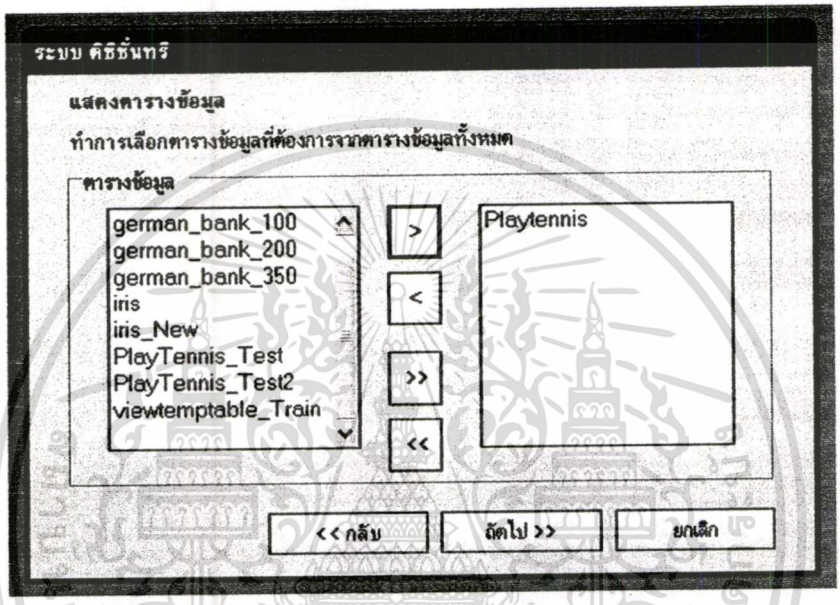
รูปที่ 5.3 ผลการทดสอบการเชื่อมต่อฐานข้อมูล

5.2 การเตรียมข้อมูล (Data Preparation)

ในการเตรียมชุดข้อมูลสำหรับใช้สร้างแบบจำลองพยากรณ์ ผู้ใช้จะทำการเลือกตารางข้อมูล เลือกแอ็ตทริบิว กำจัดค่าว่าง และทำการแปลงข้อมูล

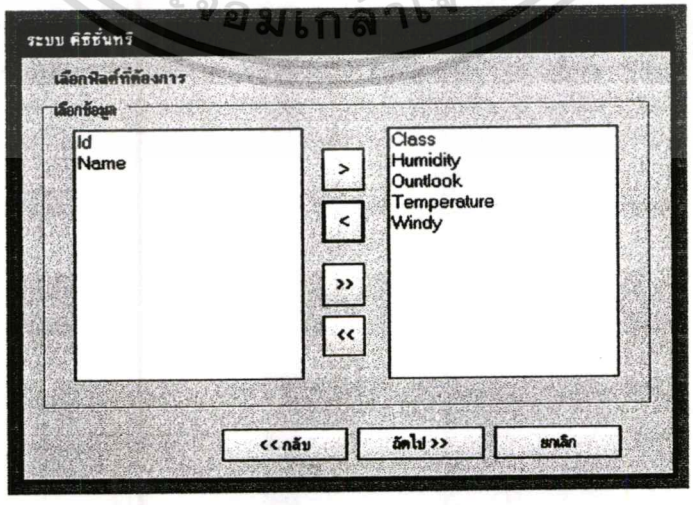
5.2.1 การเลือกข้อมูล

1) เมื่อระบบทำการติดต่อกับฐานข้อมูลแล้ว หน้าต่างของระบบจะแสดงรายชื่อตารางข้อมูลที่อยู่ในฐานข้อมูล ผู้ใช้จะทำการเลือกตารางข้อมูลสำหรับทำไมน์นิ่ง (รูปที่ 5.4)



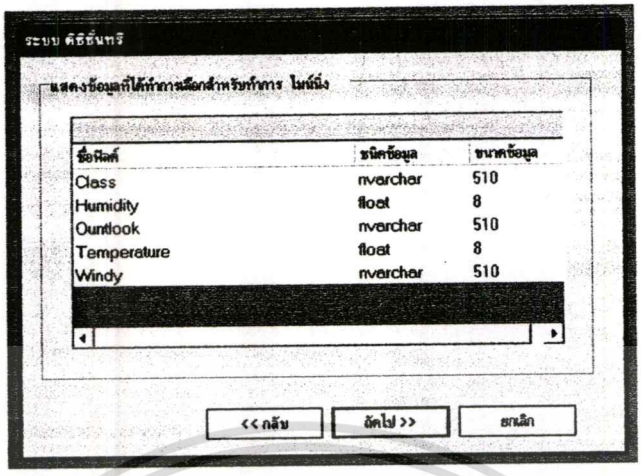
รูปที่ 5.4 หน้าต่างเลือกตารางข้อมูลสำหรับสร้างแบบจำลอง

2) ระบบจะแสดงรายชื่อแอตทริบิวในตารางข้อมูล ผู้ใช้ทำการคัดเลือกแอตทริบิวที่มีความหมาย (รูปที่ 5.5)



รูปที่ 5.5 หน้าต่างเลือกแอตทริบิวข้อมูลสำหรับสร้างแบบจำลอง

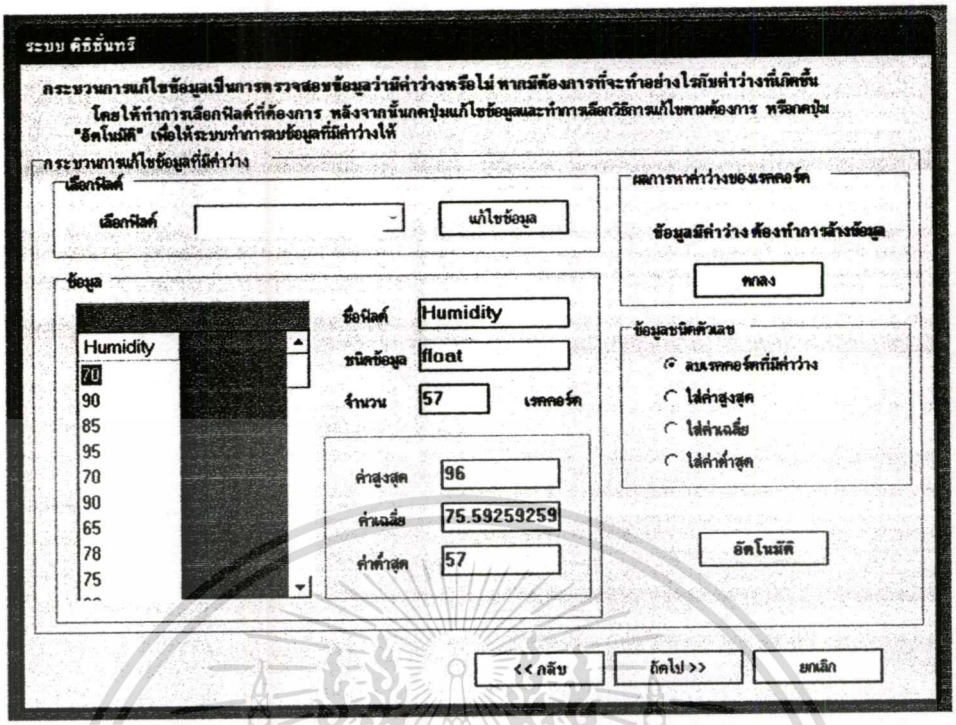
3) ระบบจะแสดงรายชื่อและคุณสมบัติของแอตทริบิวที่ถูกเลือก (รูปที่ 5.6)



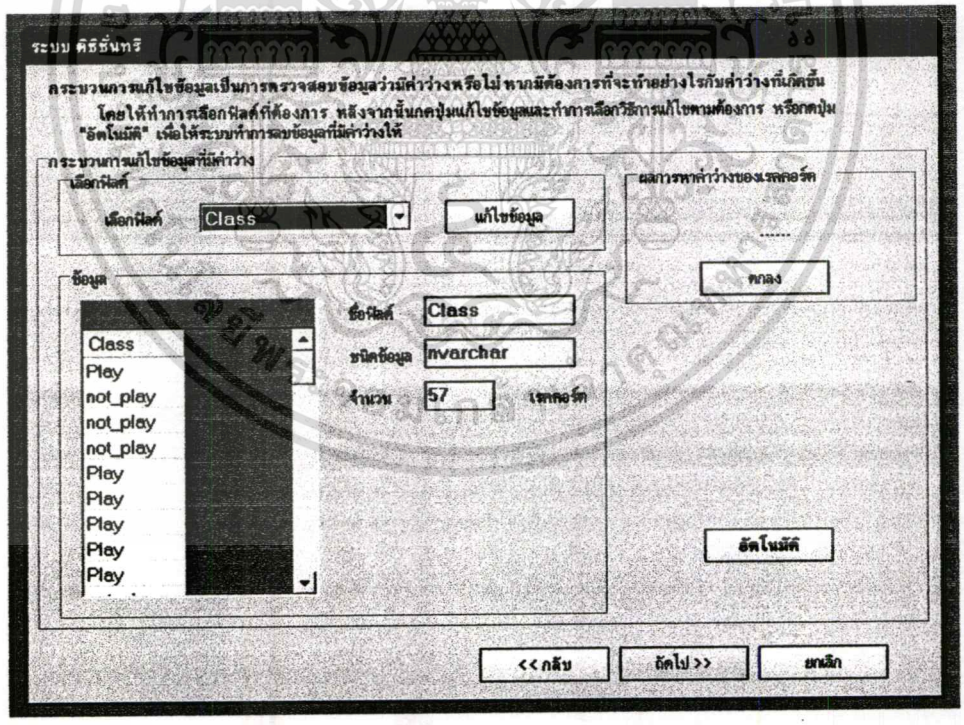
รูปที่ 5.6 หน้าต่างตารางรายชื่อและคุณสมบัติของแอตทริบิว

5.2.2 การกำจัดค่าว่าง (Data Cleaning)

1) เมื่อผู้ใช้เลือกการกำจัดค่าว่าง ระบบจะแสดงหน้าต่าง "การกำจัดค่าว่าง" โดยแสดงรายละเอียดดังนี้ (รูปที่ 5.7)
   - รายชื่อและข้อมูลของแอตทริบิว
   - วิธีในการกำจัดค่าว่าง ซึ่งมีด้วยกัน 3 วิธีคือ
     - เติมค่าเฉลี่ย (mean) ลงในข้อมูลที่ขาดไป (ข้อมูลเป็นตัวเลข)
     - เติมค่า "Unknow" ลงในข้อมูลที่ขาดไป (ข้อมูลเป็นตัวอักษร)
     - ลบระเบียนที่มีค่าว่างทิ้งไป
2) ผู้ใช้เลือกแอตทริบิวและวิธีในการกำจัดค่าว่าง (รูปที่ 5.8 การกำจัดค่าว่างแบบอัตโนมัติ)



รูปที่ 5.7 ผู้ใช้เลือกแอตทริบิวและวิธีการกำจัดค่าว่าง



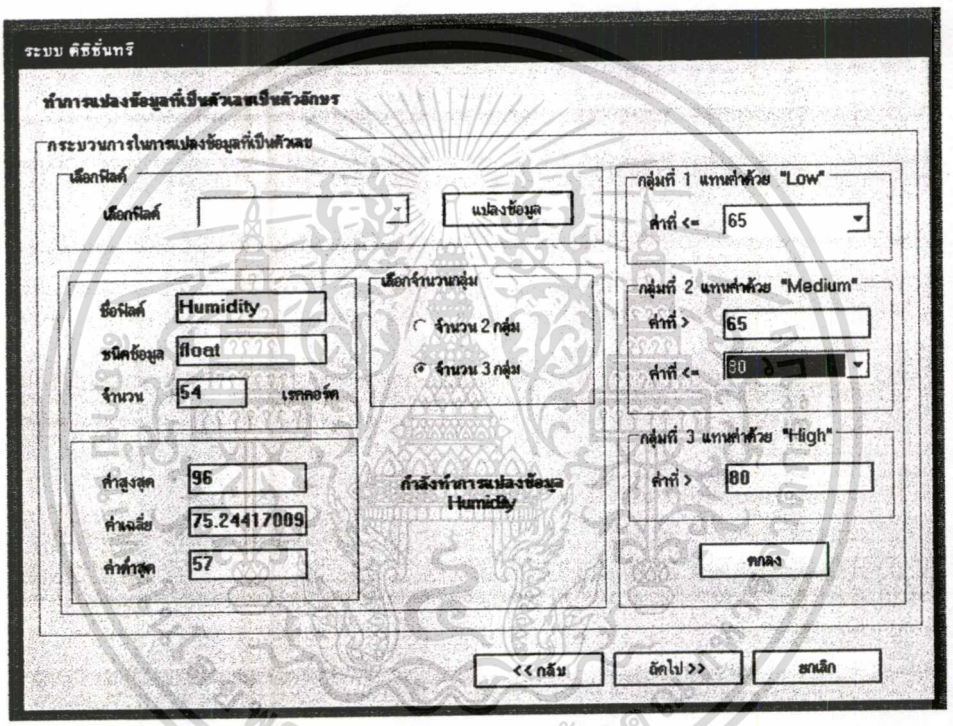
รูปที่ 5.8 ผู้ใช้เลือกให้ระบบกำจัดค่าว่างแบบอัตโนมัติ

5.2.3 การแปลงข้อมูล (Data Transformation)

เนื่องจากอัลกอริทึม CHAID รองรับรูปแบบข้อมูลที่เป็นลำดับขั้น (Category) จึงจำเป็นต้องทำการแปลงข้อมูลที่อยู่ในรูปแบบอื่นๆ ให้มีรูปแบบเป็น category

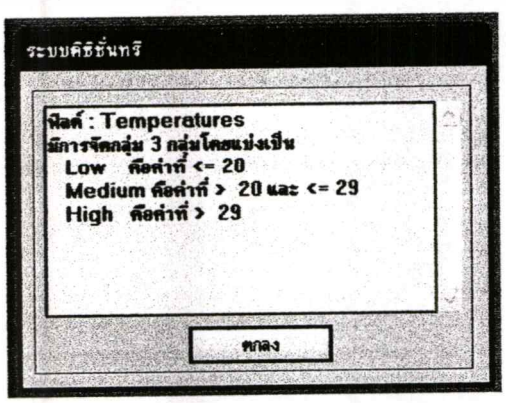
การแปลงข้อมูลที่เป็นตัวเลขหรือค่าต่อเนื่องให้เป็นข้อมูลที่เป็น category มีขั้นตอนการแปลงข้อมูล ดังนี้

1) ระบบแสดงรายชื่อแอตทริบิวและคุณสมบัติ ได้แก่ ชนิดข้อมูล ค่าสูงสุด-ต่ำสุด ค่าเฉลี่ย
2) ผู้ใช้เลือกแอ็ตทริบิวที่ต้องการแปลงข้อมูล และระบุจำนวน categories และช่วงที่ต้องการ แล้วคลิกปุ่ม "ตกลง" เพื่อให้ระบบแปลงข้อมูล (รูปที่ 5.9)



รูปที่ 5.9 ผู้ใช้เลือกแอตทริบิว ระบุจำนวน categories และช่วงค่าที่ต้องการแปลงข้อมูล

3) ระบบแสดงรายละเอียดการแปลงข้อมูลของแอตทริบิว (รูปที่ 5.10)



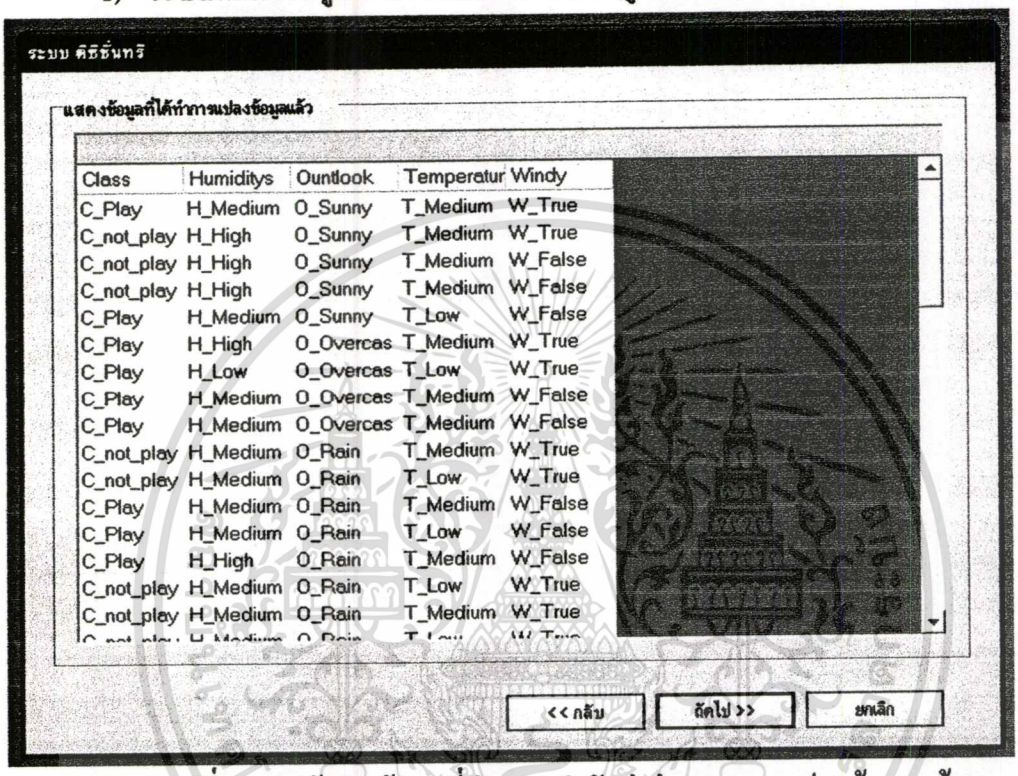
รูปที่ 5.10 หน้าต่างรายละเอียดการแปลงข้อมูล

5.3 การสร้างแบบจำลองพยากรณ์ (Decision Tree)

กระบวนการสร้างแบบจำลองพยากรณ์ แบ่งได้เป็น 2 ส่วนคือ 1) การสร้างแบบจำลอง และ 2) การตั้งชื่อและบันทึกแบบจำลอง

5.3.1 การสร้างแบบจำลอง

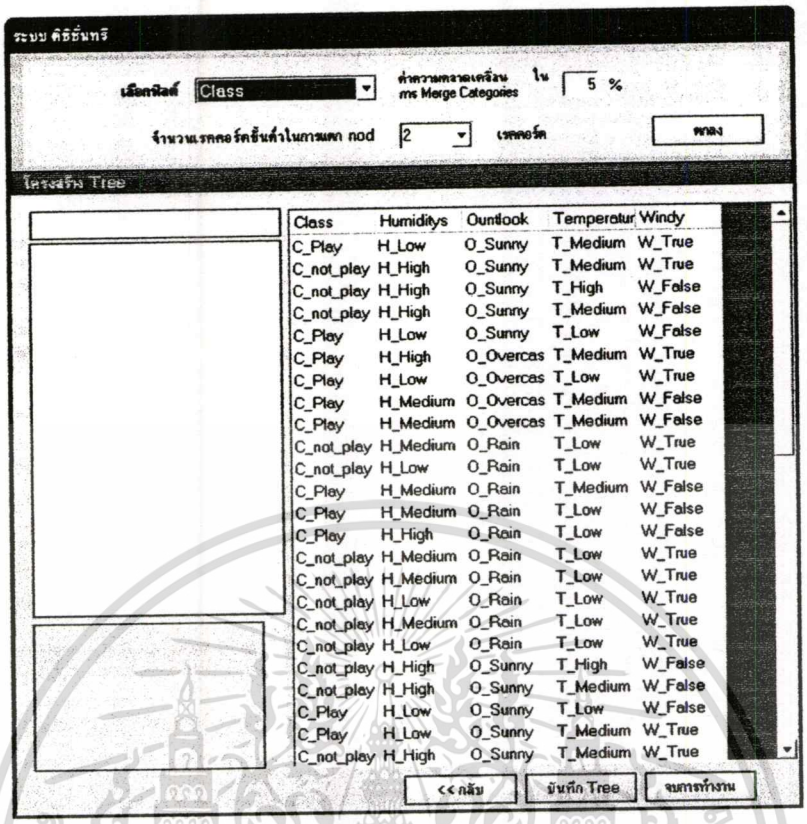
1) ระบบแสดงข้อมูลที่ผ่านการแปลงค่าแล้ว (รูปที่ 5.11)



รูปที่ 5.11 หน้าต่างข้อมูลที่ผ่านการกำจัดค่าว่างและการแปลงข้อมูลแล้ว

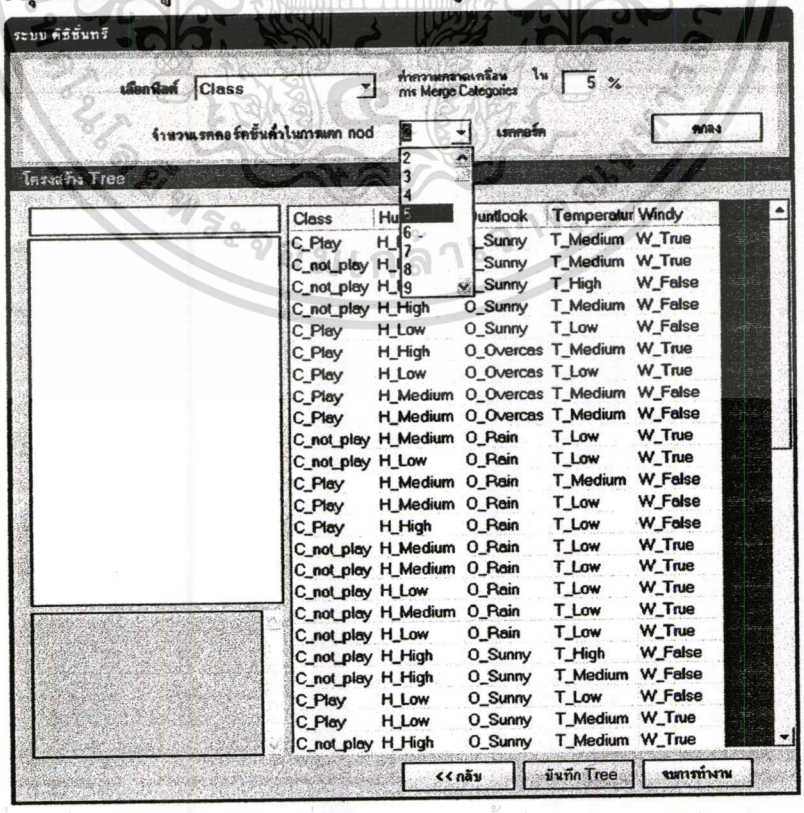
2) ระบบแสดงหน้าต่างการสร้างแบบจำลองจากข้อมูลที่ได้เตรียมไว้

3) ผู้ใช้เลือกแอตทริบิวที่เป็นแอตทริบิวเป้าหมาย (หมวดหมู่ของข้อมูล) และกำหนดค่าความคลาดเคลื่อนของการ Merge Categories หน่วยเป็นเปอร์เซ็น (ค่ามาตรฐาน = 5%) ดังรูปที่ 5.12



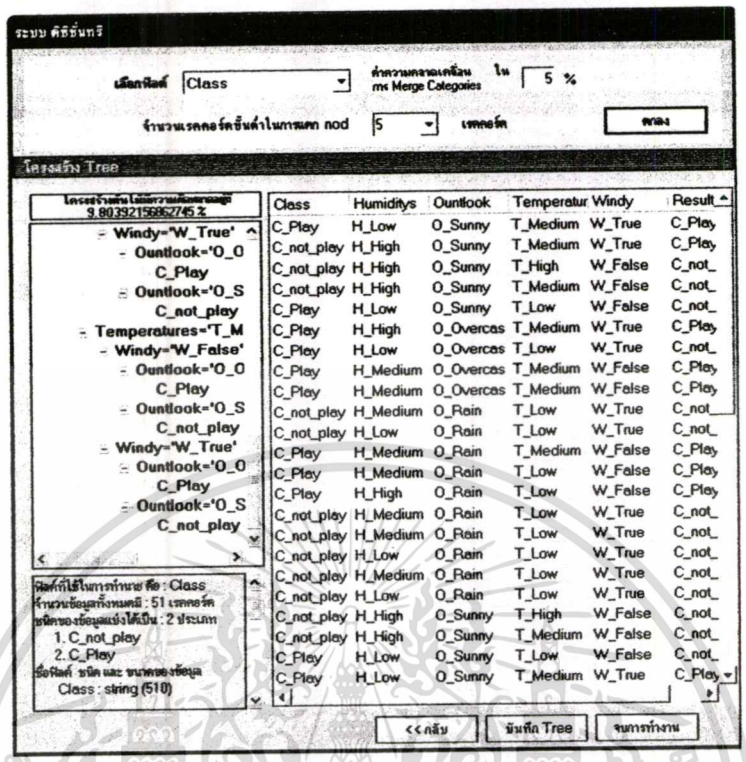
รูปที่ 5.12 ผู้ใช้แอตทริบิวเป้าหมายและกำหนดค่าความคลาดเคลื่อนของการ Merge Categories

4) ระบุจำนวนข้อมูลขั้นต่ำในการแตกโนด (รูปที่ 5.13)



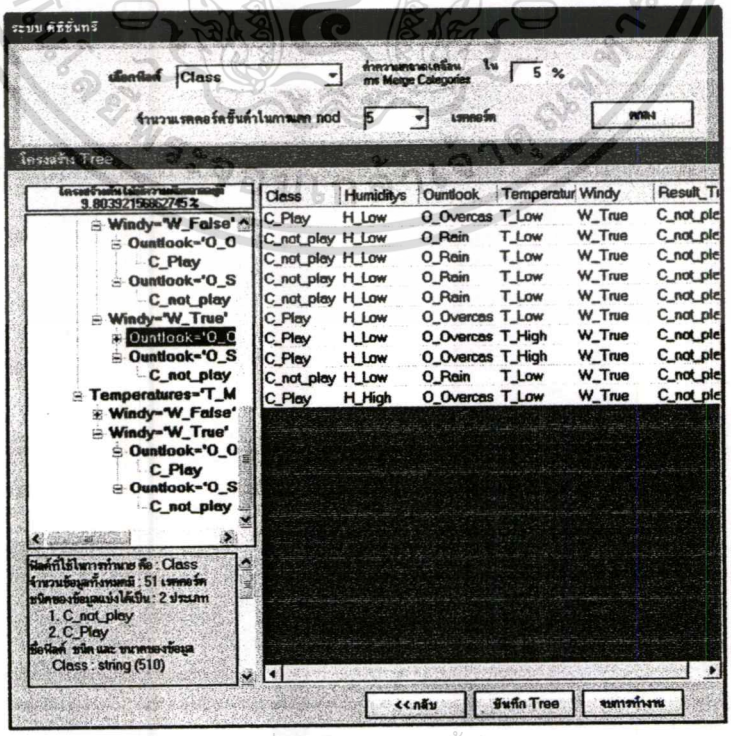
รูปที่ 5.13 ผู้ใช้ระบุจำนวนข้อมูลขั้นต่ำที่ใช้ในการแตกโนดของแขนงการตัดสินใจ

5) ระบบทำการสร้างแบบจำลองและแสดงโครงสร้าง Tree



รูปที่ 5.14 ระบบทำการสร้างและแสดงผลแบบจำลอง

6) ระบบแสดงข้อมูลของแบบจำลอง ได้แก่ โครงสร้างของแบบจำลอง ตัวคัดแยกในโนดต่างๆ และข้อมูลที่ผ่านเงื่อนไขของตัวคัดแยกนั้นๆ (รูปที่ 5.15)

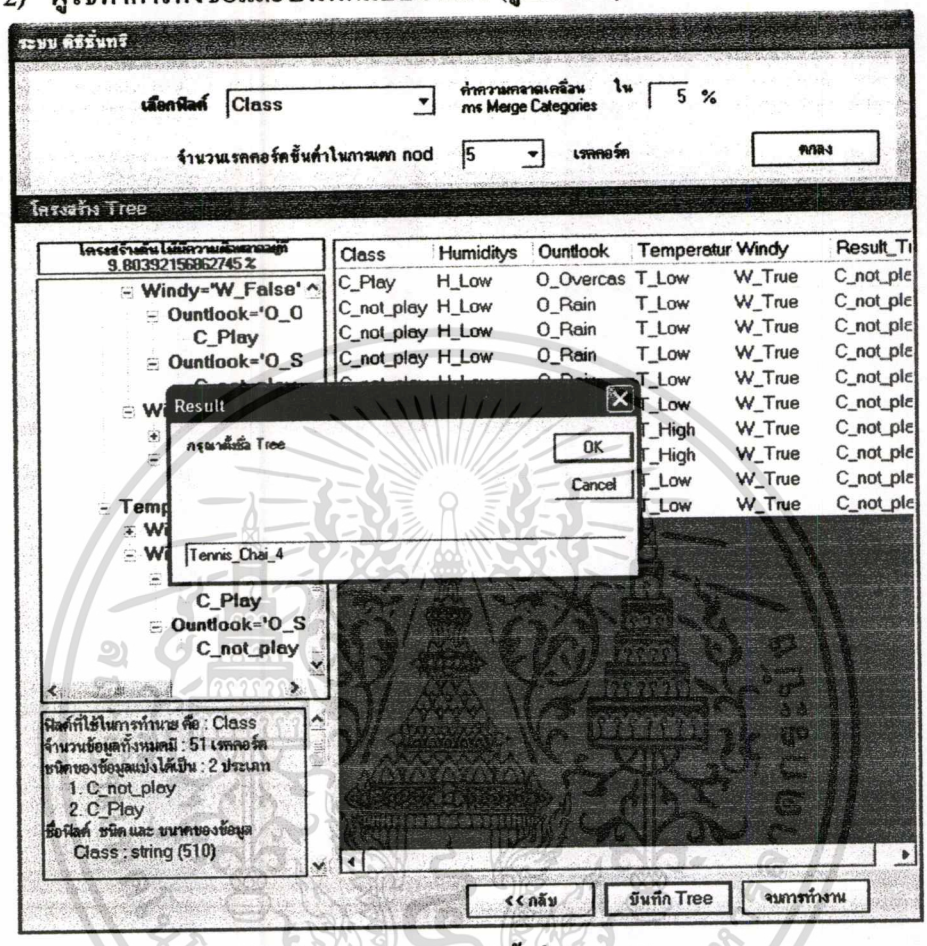


รูปที่ 5.15 โครงสร้างแขนงการตัดสินใจ ตัวคัดแยกในโนด และข้อมูลที่ผ่านเงื่อนไขของตัวคัดแยก

5.3.2 การตั้งชื่อและบันทึกแบบจำลอง

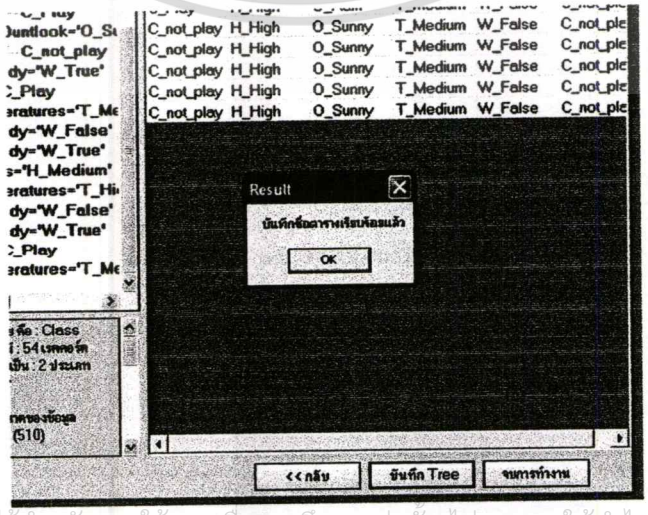
1) ระบบแสดงหน้าต่างการตั้งชื่อและบันทึกแบบจำลอง

2) ผู้ใช้ทำการตั้งชื่อและบันทึกแบบจำลอง (รูปที่ 5.16)



รูปที่ 5.16 หน้าต่างการตั้งชื่อแบบจำลอง

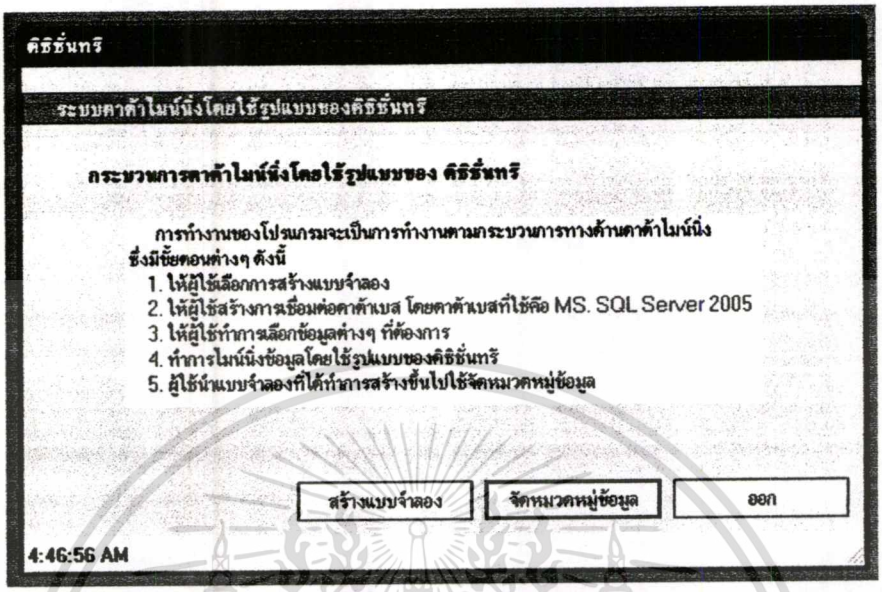
3) ระบบแสดงการยืนยันการบันทึกแบบจำลอง (รูปที่ 5.17)



รูปที่ 5.17 ผลการบันทึกแบบจำลอง

5.4 การจัดหมวดหมู่ข้อมูล (Classification) โดยใช้แบบจำลองพยากรณ์

1) ผู้ใช้เลือกการจัดหมวดหมู่ (รูปที่ 5.18)

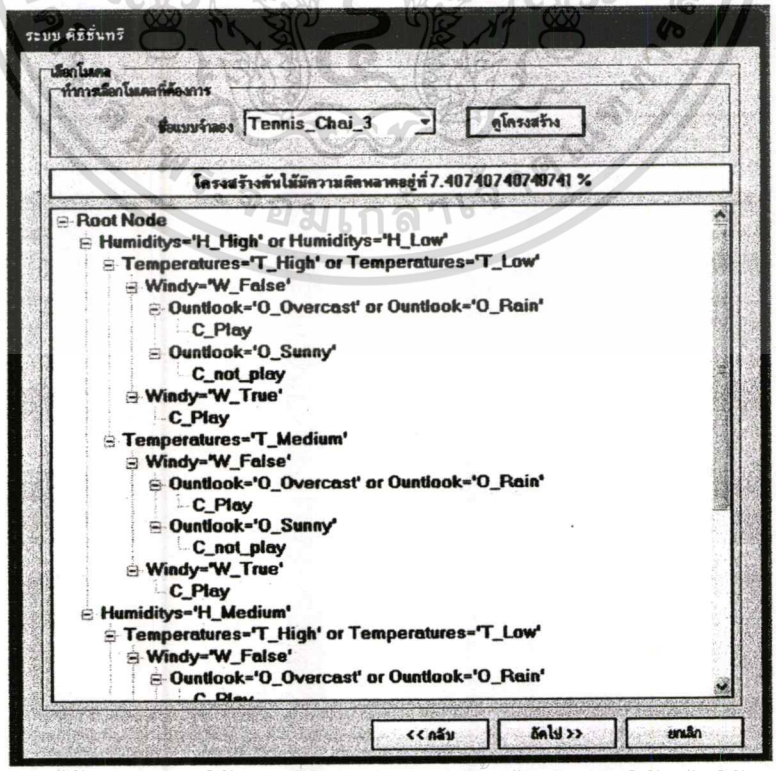


รูปที่ 5.18 ผู้ใช้เลือกการจัดหมวดหมู่ข้อมูล

2) ระบบทำการติดต่อฐานข้อมูลที่เก็บแบบจำลองที่ได้สร้างไว้แล้ว

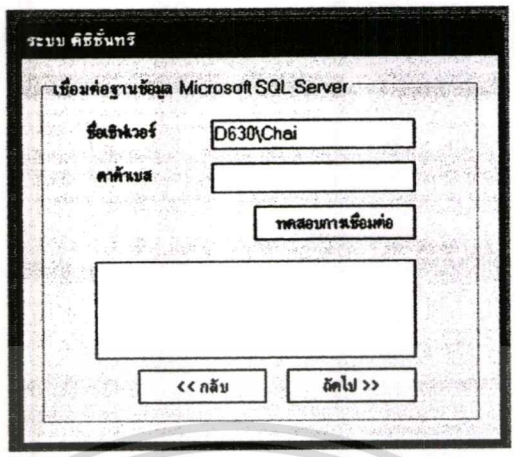
3) ผู้ใช้ทำการเลือกแบบจำลองและกดปุ่ม "ดูโครงสร้าง"

4) ระบบแสดงโครงสร้าง Tree และค่าความผิดพลาดของแบบจำลองที่ถูกเลือก(รูปที่ 5.19)



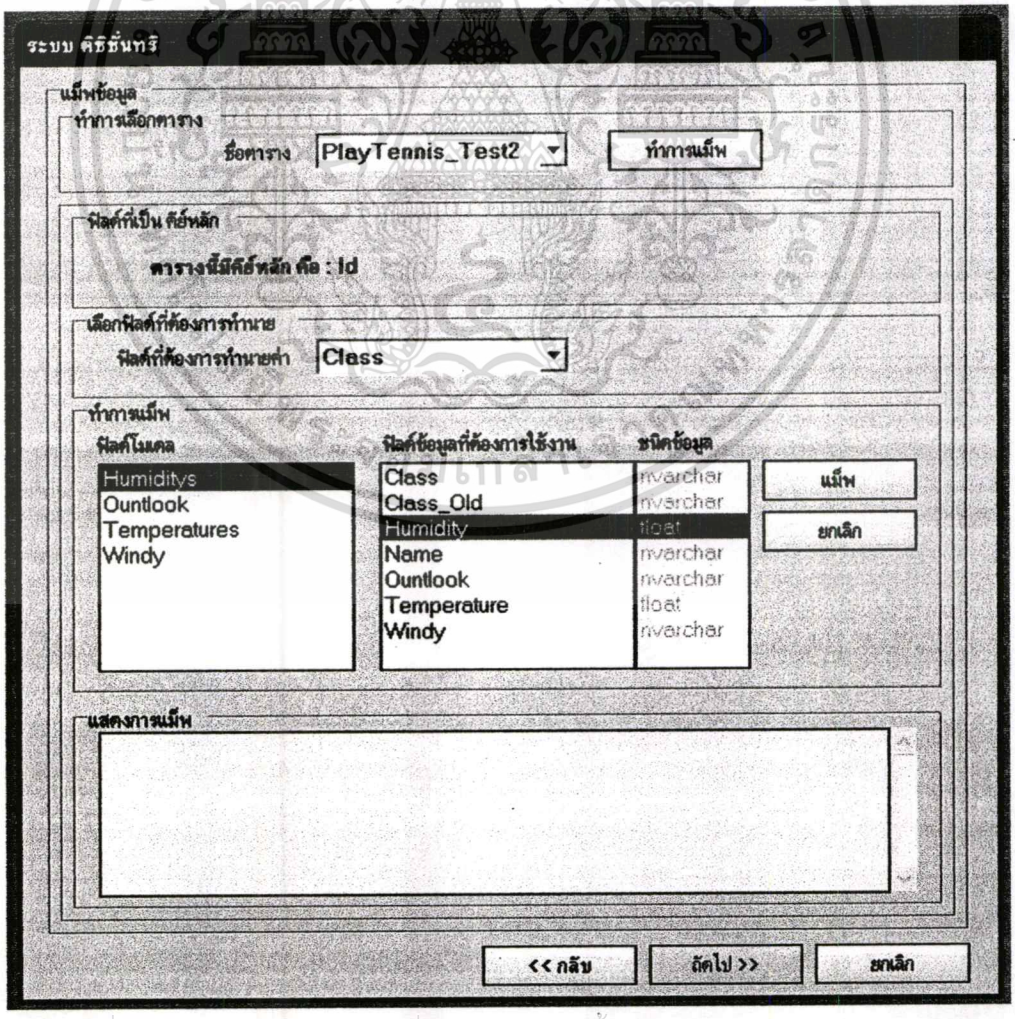
รูปที่ 5.19 ผู้ใช้เลือกแบบจำลองสำหรับจัดหมวดหมู่ข้อมูล

5) ระบบทำการติดต่อฐานข้อมูลใหม่ (รูปที่ 5.20)



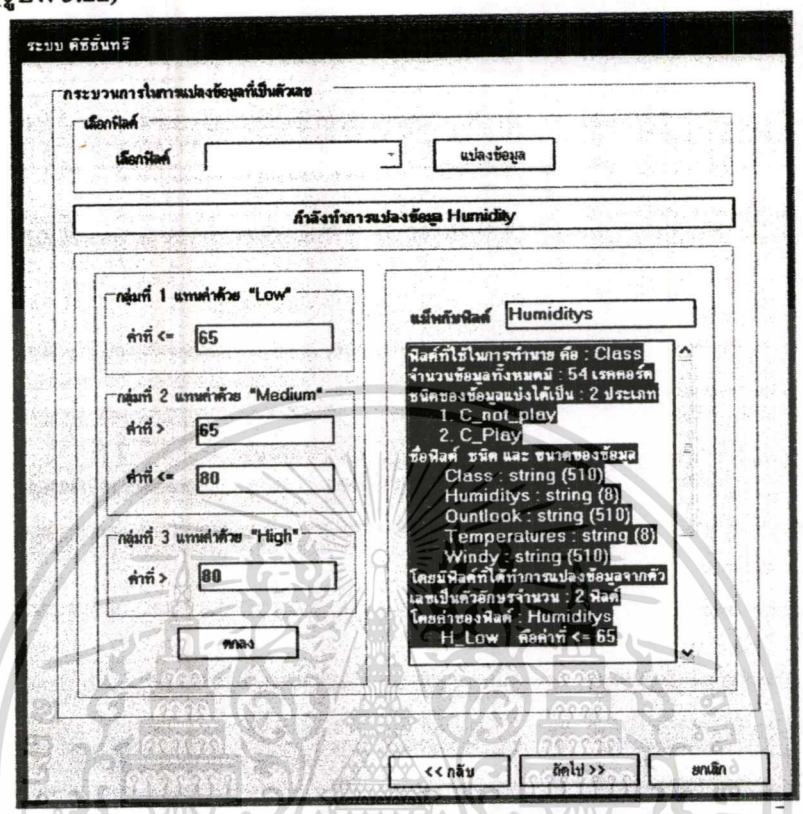
รูปที่ 5.20 ผู้ใช้กรอกชื่อเซิฟเวอร์ และชื่อฐานข้อมูลใหม่

6) ผู้ใช้ทำการเลือกตารางข้อมูล เลือกแอตทริบิวเป้าหมาย และ แมพข้อมูล (จับคู่แอตทริบิวของข้อมูลในแบบจำลองกับแอตทริบิวของข้อมูลใหม่) ดังรูปที่ 5.21



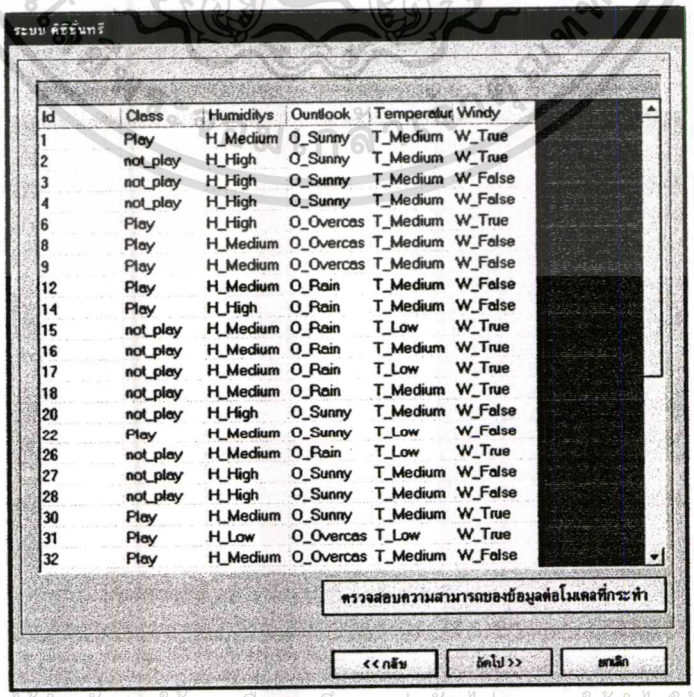
รูปที่ 5.21 ผู้ใช้เลือกตารางข้อมูลและแมพชื่อแอตทริบิว

7) ระบบทำการแปลงข้อมูลให้อยู่ในรูปแบบเดียวกับชุดข้อมูลที่ใช้สร้างแบบจำลอง (รูปที่ 5.22)



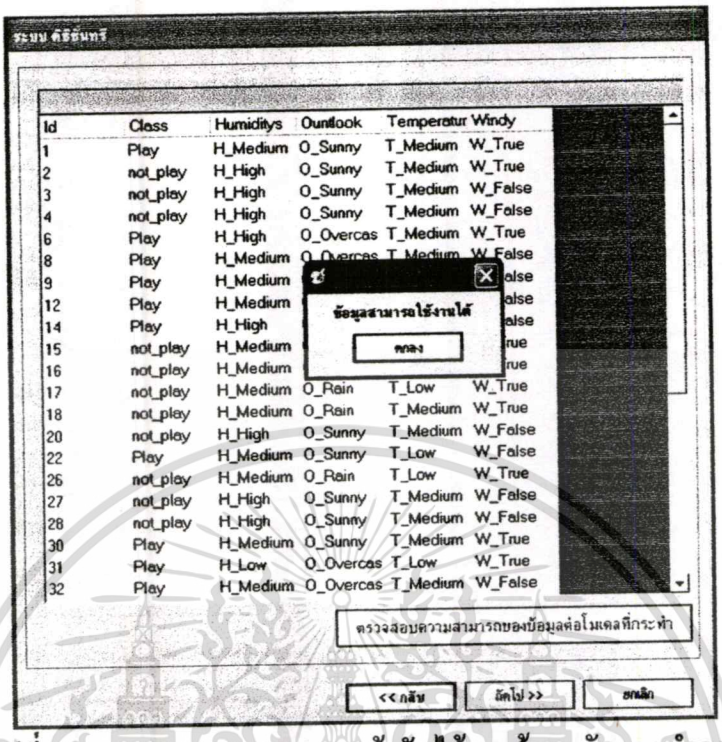
รูปที่ 5.22 ทำการแปลงข้อมูลให้ตรงกับชุดข้อมูลในแบบจำลอง

8) ผู้ใช้เลือกการตรวจสอบความเข้ากันได้ของข้อมูลกับแบบจำลอง (รูปที่ 5.23)



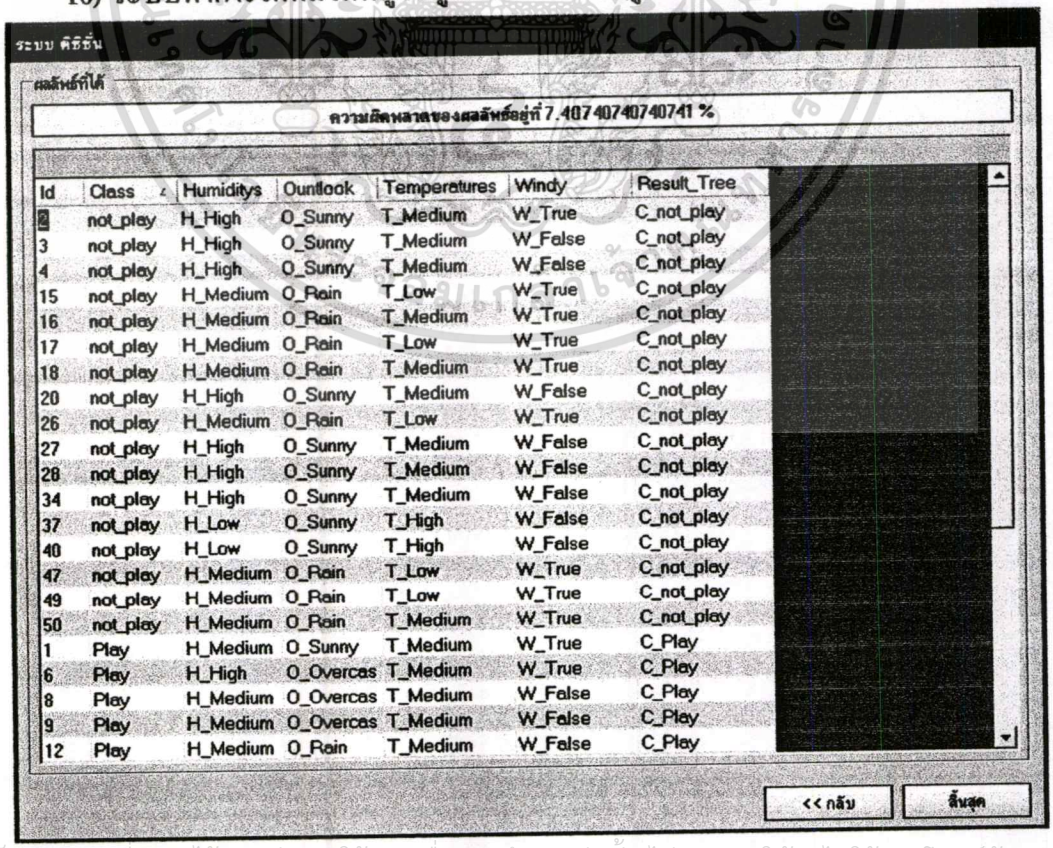
รูปที่ 5.23 หน้าต่างแสดงผลการแปลงข้อมูลและตัวเลือกการตรวจสอบข้อมูลกับแบบจำลอง

9) ระบบแสดงผลการตรวจสอบความเข้ากันได้ของข้อมูลกับแบบจำลอง (รูปที่ 5.24)



รูปที่ 5.24 ผลการตรวจสอบความเข้ากันได้ของข้อมูลกับแบบจำลอง

10) ระบบทำการจัดหมวดหมู่ข้อมูลและแสดงผล (รูปที่ 5.25)



รูปที่ 5.25 ตารางแสดงผลการจัดหมวดหมู่ข้อมูล

5.5 การวิเคราะห์ผล

ในการสร้างแบบจำลอง ผู้ใช้สามารถปรับแต่งค่าต่างๆ เพื่อให้เหมาะสมกับลักษณะของข้อมูลที่นำมาสร้างแบบจำลอง

แบบจำลองที่สร้างขึ้นมานี้ สามารถนำไปใช้ในการจัดหมวดหมู่ข้อมูลและการพยากรณ์ข้อมูลใหม่ ความถูกต้องของแบบจำลองนี้ขึ้นอยู่กับความถูกต้องและครอบคลุมของข้อมูลที่นำมาสร้างแบบจำลอง รวมถึงการเตรียมข้อมูล (Data Preparation) และปรับค่า Parameter ต่างๆ ในการสร้างแบบจำลอง ได้แก่ จำนวนระเบียนขั้นต่ำที่ยอมให้มีการแตกโนดและค่าความคลาดเคลื่อนในการ merge categories

นอกจากนี้เราสามารถนำแบบจำลองมาวิเคราะห์เพื่อค้นหาสารสนเทศหรือความรู้ใหม่ๆ แบบจำลองที่สร้างจากอัลกอริทึม CHAID นี้มีลักษณะเป็นแขนงการตัดสินใจ (Decision Tree) ตัวคัดแยกในโนดต่างๆ มีลักษณะเป็นเงื่อนไขซึ่งสามารถถอดออกมาเป็นความสัมพันธ์และกฎเงื่อนไข (If...Then) ผู้ใช้สามารถนำความสัมพันธ์และกฎเงื่อนไขเหล่านี้ไปตีความให้เป็นความรู้แล้วนำไปใช้ประโยชน์ต่อไปได้

บทที่ 6
สรุปผลและข้อเสนอแนะ

6.1 สรุปผลการศึกษา

จากการศึกษาหลักการของดาต้าไมน์นิ่ง ทำให้ได้เรียนรู้ว่าดาต้าไมน์นิ่งเป็นกระบวนการที่ค้นหาข้อมูลที่เป็นประโยชน์จากภายในฐานข้อมูลที่มีอยู่ ทำให้ได้รับสารสนเทศที่เป็นประโยชน์และสามารถนำสารสนเทศนั้นไปช่วยสนับสนุนการตัดสินใจและการประยุกต์นำไปใช้งานกับธุรกิจและงานวิจัยต่างๆได้ กระบวนการของดาต้าไมน์นิ่งเริ่มตั้งแต่การกำหนดวัตถุประสงค์ของการทำดาต้าไมนิ่ง จากนั้นก็มีขั้นตอนการเตรียมข้อมูลมาวิเคราะห์ ซึ่งจะประกอบไปด้วยการคัดเลือกข้อมูล การกำจัดค่าว่าง และการแปลงข้อมูลให้อยู่ในรูปแบบที่เหมาะสม หลังจากนั้นจะสร้างแบบจำลอง แล้วนำแบบจำลองไปใช้จัดหมวดหมู่ข้อมูลใหม่ หรือนำไปวิเคราะห์หาสารสนเทศและความรู้ ผลลัพธ์เหล่านี้สามารถนำไปใช้เพื่อให้เกิดประโยชน์ได้

ประสิทธิภาพของระบบดาต้าไมน์นิ่งนั้น เกิดจากปัจจัยหลายอย่างประกอบกัน ได้แก่ ข้อมูลที่นำมาสร้างแบบจำลอง การเตรียมข้อมูล (Data Preparation) ประเภทและชนิดของอัลกอริทึมที่ใช้ ความเข้ากันได้ของลักษณะข้อมูลกับอัลกอริทึมที่ใช้ ความสามารถของโปรแกรม และความสามารถของผู้วิเคราะห์ผล เป็นต้น

จากการพัฒนาโปรแกรมการจัดหมวดหมู่ข้อมูลนี้ พบว่าการโปรแกรมที่พัฒนาขึ้นมานี้มีข้อดีและข้อจำกัดดังนี้

<u>ข้อดี</u>

- ในการสร้างแบบจำลอง ผู้ใช้จะได้ประโยชน์สองอย่าง คือ 1.ได้แบบจำลองที่มีความสามารถในการจัดหมวดหมู่ข้อมูล 2.ได้โครงสร้าง Tree และกฎเงื่อนไข (If...Then) ที่แสดงความสัมพันธ์ของข้อมูล ซึ่งผู้ใช้สามารถนำไปวิเคราะห์เพื่อให้ได้สารสนเทศและความรู้ไปใช้ประโยชน์ต่อไป
- อัลกอริทึม CHAID มีความสามารถในการยุบรวม (Merge) Categories ในแอตทริบิว ให้เป็น categories ที่ใหญ่และมีนัยสำคัญสูง จึงทำให้ลดจำนวน node ของ Tree ลง แขนงการตัดสินใจจึงมีขนาดไม่ใหญ่เกินไป
- ผู้ใช้สามารถปรับแต่งการทำงานของโปรแกรมได้ เริ่มจากการกำหนดลักษณะการกำจัดค่าว่าง การแปลงข้อมูล และการปรับพารามิเตอร์ในการสร้างแขนงการตัดสินใจ ได้แก่ ค่าความคลาดเคลื่อนของการ merge categories จำนวนระเบียนที่ยอมให้มีการแตกโนด (ถ้ากำหนดค่าความคลาดเคลื่อนของการ merge categories ต่ำ โปรแกรมจะพยายาม merge categories เข้า เพื่อให้มีค่านัยสำคัญสูงเพียงพอ แต่

ถ้าค่าความคลาดเคลื่อนของการ merge categories สูง โปรแกรมจะไม่พยายาม merge categories มากนัก และจะแตกโนดจำนวนมากขึ้น)

- โปรแกรมสามารถรองรับฐานข้อมูลที่ชื่อแอตทริบิวมีความแตกต่างกันกับชื่อแอตทริบิวของแบบจำลองได้ แต่ categories ในแอตทริบิวนั้น จำเป็นต้องเหมือนกัน

ข้อจำกัด

- โปรแกรมการจัดหมวดหมู่ข้อมูลนี้ แบบจำลองที่ถูกสร้างขึ้นจะรู้จักรูปแบบของข้อมูลเฉพาะข้อมูลที่ได้นำมาสร้างแบบจำลองเท่านั้น ยังไม่สามารถรองรับรูปแบบของข้อมูลที่ไม่เคยมีมาก่อน

6.2 ข้อเสนอแนะ

1. การเพิ่มประสิทธิภาพของการแปลงข้อมูล(จากตัวเลขเป็น category) ซึ่งอยู่ในส่วนการเตรียมข้อมูล สามารถทำได้โดยเพิ่มการแบ่งกลุ่มข้อมูล (Clustering) เข้าไป เพื่อให้การแบ่งช่วงข้อมูลมีความเหมาะสมและมีความหมายยิ่งขึ้น
2. ในการเพิ่มความสามารถของโปรแกรมให้สามารถรองรับรูปแบบข้อมูลใหม่ๆ เช่น การรองรับข้อมูลที่มีค่าของแอตทริบิวบางตัวต่างจากข้อมูลเดิม โดยดูจากการทดสอบความเข้ากันได้ของข้อมูลใหม่กับแบบจำลอง การจัดหมวดหมู่ข้อมูลสามารถปรับเปลี่ยนรูปแบบ จากเดิมที่โปรแกรมนี้ใช้วิธีตรวจสอบเงื่อนไขว่าเป็นจริงหรือเท็จ โดยข้อมูลที่จะผ่านเงื่อนไขนั้นจะต้องได้ผลการทดสอบเป็นจริงทั้งหมด ปรับเป็น ถ้าข้อมูลใหม่มีแอตทริบิวใดที่ค่าไม่เข้ากันกับข้อมูลเดิม ให้ข้ามการตรวจสอบแอตทริบิวนั้น หรือให้ค่าการตรวจสอบแอตทริบิวนั้นเป็นจริงเสมอ แล้วดูว่าผลลัพธ์จะออกมาเป็นคลาสใดได้บ้าง แล้วเลือกเอาค่าผลลัพธ์ที่มีจำนวนมากที่สุดมาเป็นคำตอบ

ภาคผนวก ก.

โครงสร้าง Decision Tree

การสร้างโครงสร้าง Tree แบบ First Child-Next Sibling

ในการสร้างโครงสร้าง Tree เพื่อใช้เป็นแขนงการตัดสินใจ (Decision Tree) โดยใช้หลักการของ First Child-Next Sibling ใน node ของ Tree ประกอบด้วยส่วนสำคัญ 3 ส่วน คือ

1. **Node Id:** คือ Id ของตัวมันเอง
2. **Friend Node Id:** คือ Id ของ Node ที่อยู่ถัดไปใน Level เดียวกัน แต่ในกรณีที่ node นั้นเป็น node ตัวสุดท้าย ให้ใส่ Id ของโนดแม่(Parent Node) แทน
3. **First Child Node Id:** คือ Id ของ Node ลูกตัวแรก

การสร้าง Tree แบบ First Child-Next Sibling มีขั้นตอนดังนี้

1. สร้าง root node โดยให้ค่า Node Id = 0 , Friend Node Id = 0 และ First Child Node Id = -1
2. ตรวจสอบสถานภาพของ node ที่ชี้อยู่ (node ปัจจุบัน) ว่าทำการแตก node ลูกแล้วหรือยัง โดยดูจากค่า Friend Node Id ถ้าเป็น "-1" แสดงว่ายังไม่เคยแตก node ลูก แต่ถ้าไม่ใช่ "-1" แสดงว่า เคยแตก node ลูกมาแล้ว
    a. ถ้ายังไม่เคยแตก node ลูก (Friend Node Id = -1)ให้ตรวจสอบว่า สามารถแต่ node ลูกได้หรือไม่
        i. ถ้าสามารถแตก node ลูกได้ ให้ทำการแตก node ลูก (ข้อ3)
        ii. ถ้าไม่สามารถแตก node ลูกได้ ให้เคลื่อนไปยัง Friend Node (อ่านข้อมูลใน Friend Node Id แล้วเคลื่อนไปยัง node นั้น แล้วทำการตรวจสอบเช่นเดียวกันนี้ต่อไป)
    b. ถ้าเคยแตก node ลูกแล้ว (First Child Node Id ≠ -1) ให้เคลื่อนไปยัง Friend Node (อ่านข้อมูลใน Friend Node Id แล้วเคลื่อนไปยัง node นั้น แล้วทำการตรวจสอบเช่นเดียวกันนี้ต่อไป)
3. แตก node ลูกจาก node ที่ชี้อยู่ (node ปัจจุบัน) โดย
    a. คำนวณจำนวนและรายละเอียดของ node ลูกที่จะสร้าง (ตามอัลกอริทึมของแขนงการตัดสินใจ)
    b. หาค่า Node Id (เดิม) ที่มีค่ามากที่สุด แล้วเก็บไว้ในตัวแปร temporary ชื่อ MaxNodeId

c. วนลูปสร้าง node ใหม่ โดย

i. คำนวณค่า Node Id ใหม่จากตัวแปร MaxNodeId แล้วบวกเพิ่มไปเรื่อยๆ ตามลำดับของ node ลูกที่มีการแตก

ii. คำนวณค่า Friend Node Id โดยเพิ่มค่า Node Id อีก 1 (เพื่อให้เท่ากับ Node Id ของตัวถัดไป) ยกเว้น node ลูกตัวสุดท้าย ค่า Friend Node Id ต้องมีค่าเท่ากับ Node Id ของ node แม่ (node ปัจจุบัน)

iii. ให้ค่าของ First Child Node Id ของทุก node เท่ากับ "-1"

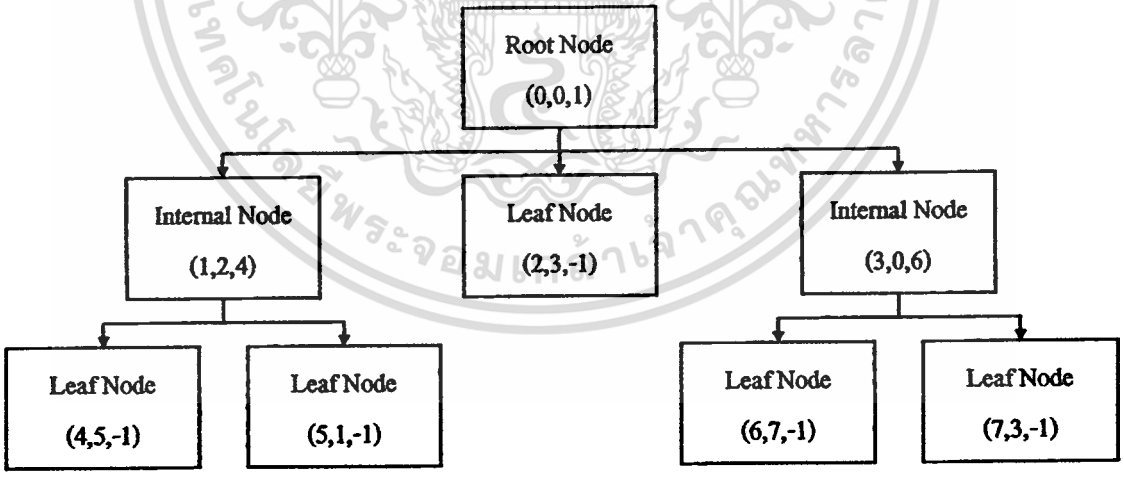
iv. ใส่ค่ารายละเอียดต่างๆ ของ node ตามอัลกอริทึมของแขนงการตัดสินใจ

d. นำ Node Id ของ node ลูกตัวแรกมาใส่ใน First Child Node Id ของ node ปัจจุบัน

e. เคลื่อน node จาก node ปัจจุบัน ไปยัง node ลูกตัวแรก (ตามค่า First Child Node Id ของ node ปัจจุบัน)

4. ทำเช่นนี้ไปเรื่อยๆ จนกระทั่งไม่สามารถแตก node ได้อีกต่อไป (โปรแกรม/Pointer ชี้กลับไปยัง root node)

ตัวอย่างการสร้างโครงสร้าง Tree

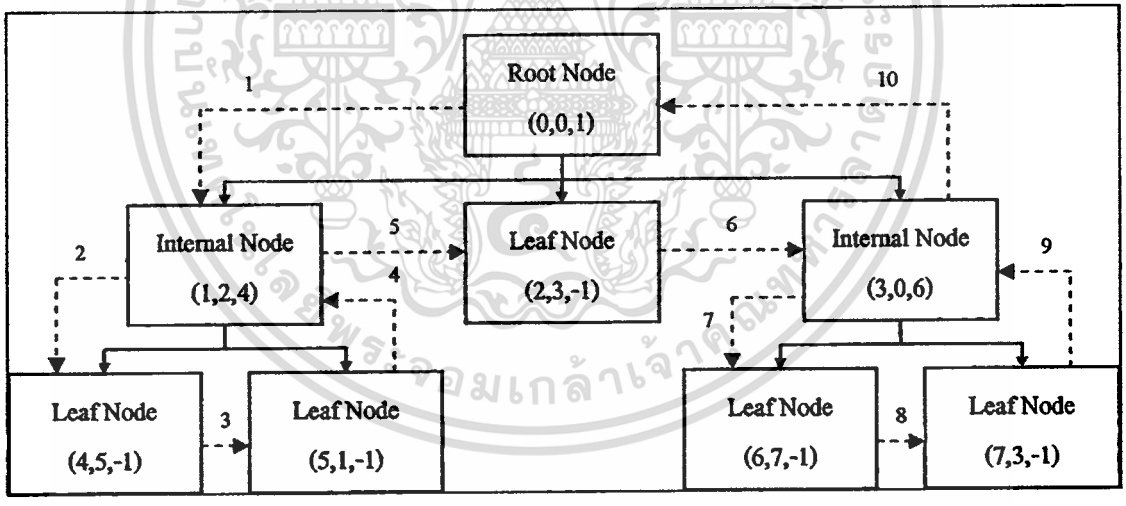


การเก็บข้อมูลในฐานข้อมูล

โครงสร้างของ Tree มีการเก็บข้อมูลในแต่ละ node ดังนี้

Node Type	Node Id	Friend Node Id	First Child Node Id	Data				
Root node	0	0	1					
Internal node	1	2	4					
Leaf node	2	3	-1					
Internal node	3	0	6					
Leaf node	4	1	-1					
Leaf node	5	1	-1					
Leaf node	6	7	-1					
Leaf node	7	3	-1					

ลำดับของการท่อง Tree แบบ First Child-Next Sibling



ภาคผนวก ข.

การทบสอบ Chi-square

1. การทดสอบไคสแควร์(Chi-square Test)[1]

การทดสอบไคสแควร์(Chi-square Test) เป็นการทดสอบความมีนัยสำคัญของข้อมูล โดยเปรียบเทียบจำนวนหรือความถี่ของข้อมูลที่เรานับได้จริง (O_i) กับจำนวนหรือความถี่ของข้อมูลที่คาดไว้ (E_i) นำไปคำนวณโดยใช้สูตร

$$\chi^2 = \sum_{i=1}^{k} \frac{(O_i - E_i)^2}{E_i}$$

เมื่อ χ^2 คือ ค่า Chi-square

O_i คือ จำนวนหรือความถี่ของข้อมูลที่นับได้จริง

E_i คือ จำนวนหรือความถี่ของข้อมูลที่คาดไว้

k คือ จำนวนลักษณะของข้อมูล

<u>ตัวอย่าง</u>

ข้อมูลชุดหนึ่งประกอบด้วยตัวแปร X และ ตัวแปร Y จำนวน 100 ระเบียน โดยตัวแปร X มีค่าอยู่ 4 ลักษณะ คือ x1, x2, x3, x4 และตัวแปร Y มีค่าอยู่ 4 ลักษณะ คือ y1, y2, y3, y4 สามารถเขียนเป็นตารางแจกแจงความถี่ได้ดังนี้

X \ Y	y1	y2	y3	y4	Sum of Row
x1	23	5	19	4	*51*
x2	12	2	15	13	*42*
x3	0	1	0	1	*2*
x4	0	0	1	4	*5*
Sum of Column	*35*	*8*	*35*	*22*	*100*

ตารางความถี่ที่นับได้ของข้อมูลตัวแปร X และ Y

เราสามารถคำนวณ "ค่าที่คาดไว้" แต่ละตัว โดยใช้สูตร $\frac{(\text{Sum of Row}) \times (\text{Sum of Column})}{Total}$

---

[1] วีนัสพีรวณิชย์ และสมจิต วัฒนาชยากูล. สถิติ สำหรับนักสังคมศาสตร์ สาขาวิชาคณิตศาสตร์-สถิติ คณะศิลปศาสตร์ มหาวิทยาลัยธรรมศาสตร์

X \ Y	Y1	Y2	Y3	Y4
X1	(51x35)/100	(51x8)/100	(51x35)/100	(51x22)/100
X2	(42x35)/100	(42x8)/100	(42x35)/100	(42x22)/100
X3	(2x35)/100	(2x8)/100	(2x35)/100	(2x22)/100
X4	(5x35)/100	(5x8)/100	(5x35)/100	(5x22)/100

ได้ตารางความถี่ที่คาดไว้ (E_i) ของข้อมูลตัวแปร X และ Y ดังนี้

X \ Y	Y1	Y2	Y3	Y4
X1	17.85	4.08	17.85	11.22
X2	14.7	3.36	14.7	9.24
X3	0.7	0.16	0.7	0.44
X4	1.75	0.4	1.75	1.1

ตารางความถี่ที่คาดไว้ของข้อมูลตัวแปร X และ Y

คำนวณค่า Chi-square โดยใช้สูตร $\chi^2 = \sum_{i=1}^{k} \frac{(O_i - E_i)^2}{E_i}$

$$\chi^2 = \frac{(23-17.85)^2}{17.85} + \frac{(5-4.08)^2}{4.08} + \frac{(19-17.85)^2}{17.85} + \ldots + \frac{(4-1.1)^2}{1.1}$$

$$\chi^2 = 25.63559$$

2. การหาค่าความเป็นอิสระของข้อมูล (Degrees of freedom หรือ d.f.)

ความเป็นอิสระของข้อมูล (Degrees of freedom หรือ d.f.) คือ ค่าของข้อมูลที่สามารถเปลี่ยนแปลงได้[2] เช่น ข้อมูลชุดหนึ่งมีจำนวน n ตัวแล้วเลือกมาเพียง 1 ตัว จำนวนที่เราสามารถเปลี่ยนค่าได้คือ n-1 (เพราะเราได้ดึงข้อมูลออกมาแล้ว 1 ตัว จึงเหลือข้อมูลที่สามารถเปลี่ยนค่าได้เพียง n-1 ตัว)

ค่า degrees of freedom ของข้อมูล 2 มิติ หาได้จากสูตร

$$d.f. = (row - 1) \times (column - 1)$$

เมื่อ

d.f. คือ degrees of freedom

row คือ จำนวนแถวของข้อมูล

column คือ จำนวนหลักของข้อมูล

[2] คณาจารย์ภาควิชาสถิติ. 2529. หลักสถิติ คณะพาณิชยศาสตร์และการบัญชี จุฬาลงกรณ์มหาวิทยาลัย

ตัวอย่าง

ตารางแจกแจงความถี่ของข้อมูล 100 ระเบียน ประกอบด้วยตัวแปร X และตัวแปร Y

X \ Y	y1	y2	y3	y4	Sum of Row
x1	23	5	19	4	*51*
x2	12	2	15	13	*42*
x3	0	1	0	1	*2*
x4	0	0	1	4	*5*
Sum of Column	*35*	*8*	*35*	*22*	*100*

row = 4, column = 4

d.f. = (row -1) x (column-1)

= 3 x 3

d.f. = 9

3. การคำนวณค่า P-Value

ค่า P-Value เป็นค่าที่แสดงความมีนัยสำคัญของตัวแปร โดยดูจากค่า Chi-square และค่า degrees of freedom แล้วนำไปเปรียบเทียบกับตารางการแจกแจง Chi-square

	P-values								
Degrees of Freedom	0.900	0.750	0.500	0.250	0.100	0.050	0.025	0.010	0.005
1	0.01579	0.10153	0.45494	1.32330	2.70554	3.84146	5.02389	6.63490	7.87944
2	0.21072	0.57536	1.38629	2.77259	4.60517	5.99146	7.37776	9.21034	10.59663
3	0.58437	1.21253	2.36597	4.10834	6.25139	7.81473	9.34840	11.34487	12.83816
4	1.06362	1.92256	3.35669	5.38527	7.77944	9.48773	11.14329	13.27670	14.86026
5	1.61031	2.67460	4.35146	6.62568	9.23636	11.07050	12.83250	15.08627	16.74960
6	2.20413	3.45460	5.34812	7.84080	10.64464	12.59159	14.44938	16.81189	18.54758
7	2.83311	4.25485	6.34581	9.03715	12.01704	14.06714	16.01276	18.47531	20.27774
8	3.48954	5.07064	7.34412	10.21885	13.36157	15.50731	17.53455	20.09024	21.95495
9	4.16816	5.89883	8.34283	11.38875	14.68366	16.91898	19.02277	21.66599	23.58935
10	4.86518	6.73720	9.34182	12.54886	15.98718	18.30704	20.48318	23.20925	25.18818

ตารางเทียบค่า Chi-square และ P-Value

ตัวอย่าง

ค่า d.f. = 9, ค่า Chi-square(χ^2) = 25.63559

เทียบบัญญัติไตรยางค์ จากตารางการแจกแจง Chi-square ที่ d.f. = 9

$$\frac{0.005-P}{0.005-0,001}=\frac{25.63559-23.58935}{38.43986-23.58935}$$

P = 0.004448843

∴ ค่า P-Value = 0.004448843

ประวัติผู้เขียน

ชื่อผู้เขียน	นายชัยธวัช ธีรานุสนธิ์
วันเกิด	13 ตุลาคม 2523
สถานที่เกิด	กรุงเทพมหานคร
วุฒิการศึกษาระดับปริญญาตรี	วิศวกรรมศาสตรบัณฑิต
สถานที่สำเร็จการศึกษา	คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีที่สำเร็จการศึกษา	2544