การพัฒนาระบบสนับสนุนการพิจารณาอนุมัติให้สินเชื่อบัตรเครดิต โดยใช้ดิซิชั่นทรี

Apply Decision tree to Credit card loan Approval Support System

โดย

วิชุดา ศรีศิริศุภโยค

รหัส 46066325

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน

หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

ภาคฤดูร้อน ปีการศึกษา 2547

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

หัวข้อ	การพัฒนาระบบสนับสนุนการพิจารณาอนุมัติให้สินเชื่อบัตรเครดิตโดยใช้ดิซิชั่นทรี
นักศึกษา	นางสาว วิชุดา ศรีศิริศุภโยค
อาจารย์ที่ปรึกษา	ผศ. ดร. วรพจน์ กรีสุระเดช
ระดับการศึกษา	วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2547

บทคัดย่อ

ปัจจุบันกระบวนการทางด้านดาต้าไมน์นิ่ง สามารถนำไปใช้ในกระบวนการทำงานทางธุรกิจได้มากมายหลายด้าน ทางด้านการเงินการธนาคาร การให้สินเชื่อต่างๆ ก็สามารถนำกระบวนการดาต้าไมน์นิ่งมาช่วยพิจารณาการอนุมัติวงเงินสินเชื่อต่างๆได้เช่นกัน โครงการพัฒนาระบบงานนี้มีวัตถุประสงค์เพื่อศึกษาและทำความเข้าใจขั้นตอนการทำงานตามแนวคิดของดาต้าไมน์นิ่ง ในรูปแบบของวิธีการ Classification โดยประยุกต์ใช้เพื่อศึกษาถึงแนวทางและความเป็นไปได้ของการใช้หลักการ Decision Tree กับกระบวนการพิจารณาอนุมัติสินเชื่อบัตรเครดิต

Title	Apply Decision tree to Credit card loan approval support system
Student	Miss. Vichuda Srisirisupayok
Advisor	Asst. Prof. Dr. Worapoj Kresuradej
Level of Study	Master of Science in information Technology
Major	Information Science
Academic Year	2004

ABSTRACT

Presently, data mining process can be utilized with many business aspects. In financial industry, it can be used for loan approval decision. This system development project has an objective to study and understand operation process and concept of data mining with classification methodology. This will be applied for concept and feasibility studying of decision tree and credit card loan approval process.

กิตติกรรมประกาศ

ในการพัฒนาระบบงานในครั้งนี้ทางผู้จัดทำขอขอบพระคุณ ผศ. ดร. วรพจน์ กรีสุระเดช เป็นอย่างสูง ที่เป็นผู้ให้ความรู้และให้คำแนะนำในการพัฒนาระบบงาน ขอขอบคุณพี่ๆ เพื่อนๆ ทุกคนใน TSS ที่ให้คำแนะนำและเป็นที่ปรึกษาที่ดีในการพัฒนาระบบ พร้อมทั้งให้ความช่วยเหลืออย่างมากมาย และท้ายที่สุดนี้ขอขอบคุณ คุณแม่ และบอล ที่เป็นพลังเกื้อหนุนที่ดีที่สุด ที่ทำให้ผู้จัดทำมีวันนี้

ขอบคุณมากค่ะ

วิชุดา ศรีศิริศุภโยค

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....I

บทคัดย่อภาษาอังกฤษ.....II

กิตติกรรมประกาศ.....III

สารบัญ.....IV

สารบัญตาราง.....VI

สารบัญภาพ.....VII

บทที่

1. บทนำ.....1

1.1 หลักการและเหตุผล.....1

1.2 วัตถุประสงค์/ เป้าหมาย.....2

1.3 ขอบเขตของการดำเนินงาน.....2

1.4 ขั้นตอนการดำเนินงาน.....3

1.5 แผนการดำเนินงาน/ ระยะเวลา.....4

1.6 ประโยชน์ที่คาดว่าจะได้รับ.....5

2. ทฤษฎี เครดิต สกอร์ริ่ง.....6

2.1 ประวัติเครดิต สกอร์ริ่ง.....6

2.2 ความหมายของเครดิต สกอร์ริ่ง.....6

2.3 โมเดล เครดิต สกอร์ริ่ง.....7

2.4 การพัฒนา โมเดล เครดิต สกอร์ริ่ง.....8

2.5 หลักการให้เครดิต.....11

2.6 ประโยชน์ที่ได้รับจาก เครดิต สกอร์ริ่ง.....12

3. ดาต้าไมน์นิ่ง.....14

3.1 ความเป็นมาของดาต้าไมน์นิ่ง.....14

3.2 การทำงานของดาต้าไมน์นิ่ง (Data Mining Process).....14

3.3 รูปแบบการเก็บข้อมูลที่สามารถทำดาต้าไมน์นิ่ง.....15

3.4 รูปแบบข้อมูลตัวแปรในการทำดาต้าไมน์นิ่ง.....15

3.5 ขั้นตอนในการทำดาต้าไมน์นิ่ง.....15

สารบัญ (ต่อ)

หน้า

บทที่

3.6 ประเภทงานทางด้านดาต้าไมน์นิ่ง.....16

3.7 ประเภทของแบบจำลองสำหรับการทำดาต้าไมน์นิ่ง (Data mining Model).....16

3.8 SPRINT Algorithm.....18

3.9 ประโยชน์ของดาต้าไมน์นิ่ง.....34

4. ขั้นตอนการพัฒนาระบบการพิจารณาอนุมัติให้สินเชื่อบัตรเครดิตโดยใช้ดิซิชั่นทรี.....35

4.1 การศึกษาความต้องการของระบบ.....35

4.2 การวิเคราะห์และออกแบบระบบ.....36

5. สรุปผลการทดสอบ/ ข้อเสนอแนะ.....48

5.1 สรุปผลการทดสอบ.....48

5.2 ข้อเสนอแนะ.....49

บรรณานุกรม..... 50

ประวัติผู้เขียน.....51

สารบัญตาราง

หน้า

ตารางที่

1. MINIG_TABLE.....37
2. ตัวอย่าง Text File ที่ใช้เป็น Input ในการ Mining.....41
3. โครงสร้างและตัวอย่างแบบจำลองต้นไม้ที่ทำการบันทึกเป็นไฟล์.....44

สารบัญภาพ

หน้า

รูปที่

1. แสดงขั้นตอนที่ 1 นิยามปัญหาและจัดเตรียมข้อมูล.....9
2. แสดงขั้นตอนของกระบวนการสร้างโมเดลเครดิต สกอร์ริ่ง.....11
3. ค่า Histograms ของ continuous attribute.....21
4. ค่า Histograms ของ categorical attribute.....22
5. Decision Tree ผ่านการ Split ครั้งที่ 1.....29
6. Decision Tree ผ่านการ Split ครั้งที่ 2.....32
7. Decision Tree ของข้อมูลตัวอย่างด้วย SPRINT.....33
8. โครงสร้างความสัมพันธ์ของตารางต่างๆ ในระบบ Credit card System.....36
9. Logon Screen.....39
10. Invalid User name.....39
11. Invalid Password.....39
12. Building Panel.....40
13. Testing Tree Panel.....42
14. Tree Model Panel.....43
15. Classification Rules Panel.....45
16. Predict Scoring Panel.....46
17. Event Log.....47
18. Classification Rules จากการ Training Tree.....48
19. Accuracy จากการ Testing Tree.....49

บทที่ 1
บทนำ

1.1 หลักการและเหตุผล

ในปัจจุบันดาต้าไมน์นิ่ง (Data Mining) เป็นเทคโนโลยีที่กำลังได้รับความสนใจและถูกประยุกต์ใช้ประโยชน์ในหลายๆ ธุรกิจ เนื่องจากสามารถค้นหาองค์ความรู้ (Knowledge) ที่ซ่อนอยู่ในข้อมูลที่รวบรวมไว้จำนวนมากมาประกอบการดำเนินงานทางธุรกิจ ทั้งนี้เนื่องจากเทคโนโลยีดาต้าไมน์นิ่งเป็นเทคโนโลยีที่ไม่เพียงแต่ประมวลผลข้อมูลในอดีตจนถึงปัจจุบันเท่านั้น แต่ยังสามารถพยากรณ์ (Prediction) ไปสู่ข้อมูลในอนาคต โดยการสร้างแบบจำลอง (Model) ข้อมูลขนาดใหญ่ สำหรับค้นหารูปแบบ (Pattern) การเกิดข้อมูลและความสัมพันธ์ของข้อมูลเพื่อนำผลที่ได้จากการวิเคราะห์ข้อมูลไปช่วยสนับสนุนการตัดสินใจซึ่งในการดำเนินธุรกิจองค์กรที่สามารถคาดการณ์ล่วงหน้าเพื่อใช้ในการวางแผนงานต่างๆ ในอนาคต จะเป็นผู้ที่ได้เปรียบและสร้างโอกาสทางธุรกิจเหนือคู่แข่ง

ปัจจุบันธุรกิจทางด้านการเงินการธนาคาร รวมทั้งบริษัทที่ดำเนินธุรกิจทางด้านการให้สินเชื่อมีการนำวิธีการให้คะแนนผู้สมัครขออนุมัติสินเชื่อมาใช้ในการประเมินความเสี่ยงที่จะเกิดขึ้น เพื่อประกอบการพิจารณาการให้สินเชื่อแก่ลูกค้าแต่ละราย วิธีการนี้เรียกว่า "เครดิต สกอร์ริ่ง (Credit Scoring)" ซึ่งเป็นวิธีการที่ช่วยเพิ่มประสิทธิภาพของการพิจารณาการให้สินเชื่อให้มีมากขึ้นกว่าในอดีต ที่ใช้วิธีการพิจารณาคุณสมบัติและประวัติของผู้สมัคร โดยพนักงานสินเชื่อจะพิจารณาตามหลักเกณฑ์ที่วางไว้ล่วงหน้า ประกอบกับรายงานเครดิตของผู้สมัครขอสินเชื่อ ซึ่งวิธีการแบบเดิมนี้มักใช้เวลาในการทำรายงานเป็นเวลานาน และมักเกิดความคลาดเคลื่อนในการพิจารณา และเป็นสาเหตุหนึ่งที่ทำให้เกิดหนี้เสียในระบบเศรษฐกิจเพิ่มมากขึ้น ด้วยเหตุนี้จึงได้มีการนำวิธีการเครดิตสกอร์ริ่ง มาใช้เพื่อให้การพิจารณาอนุมัติสินเชื่อเป็นวิธีการที่สามารถจัดการได้ และมีหลักการชัดเจน ยุติธรรม อีกทั้งยังเป็นการนำเสนอข้อมูลที่ใช้ช่วยในการตัดสินใจได้อย่างรวดเร็ว และทันสมัย

ในโครงงานนี้จะกล่าวถึงการพัฒนาระบบช่วยวิเคราะห์การอนุมัติสินเชื่อบัตรเครดิต โดยใช้ดิซิชั่นทรีโดยนำอัลกอริทึมที่ชื่อว่า SPRINT เป็นหลักในการเขียนโปรแกรม ซึ่งระบบนี้จะช่วยสนับสนุนข้อมูลในด้านการตัดสินใจให้กับผู้มีหน้าที่พิจารณาอนุมัติสินเชื่อ โดยผลการวิเคราะห์จะแสดงออกมารูปของแบบจำลองต้นไม้ (Tree Model)

1.2 วัตถุประสงค์/ เป้าหมาย

โครงการพัฒนาระบบงานนี้มีวัตถุประสงค์และเป้าหมายดังต่อไปนี้

1. เพื่อศึกษาและทำความเข้าใจเกี่ยวกับแนวคิดและขั้นตอนการทำคาต้าไมนิ่ง
2. เพื่อศึกษาและทำความเข้าใจในหลักการเบื้องต้นของเดรดิต สกอร์ริ่ง
3. เพื่อศึกษาและทำความเข้าใจเกี่ยวกับการใช้แนวคิดClassificationโดยนำหลักการดิซิชั่นทรีและอัลกอริทึมที่ชื่อว่า SPRINT (Scalable PaRallelizable INduction of decision Trees) เข้ามาใช้
4. เพื่อศึกษาถึงแนวทางและความเป็นไปได้ในการนำแนวคิดดิซิชั่นทรีมาใช้ในการวิเคราะห์ข้อมูลสินเชื่อบัตรเครดิต เพื่อสนับสนุนกระบวนการพิจารณาอนุมัติสินเชื่อบัตรเครดิต ว่าสามารถนำมาวิเคราะห์ข้อมูลได้จริงและมีประสิทธิภาพหรือไม่
5. เพื่อเป็นแนวทางในการประยุกต์ใช้เทคนิคคาต้าไมน์นิ่งในการสนับสนุนการตัดสินใจให้สถาบันการเงิน หรือผู้ให้สินเชื่อต่างๆ สามารถนำข้อมูลที่ได้วิเคราะห์เพื่อประกอบการพิจารณาต่อไปอย่างมีประสิทธิภาพ

1.3 ขอบเขตของการดำเนินงาน

โครงการพัฒนาระบบงานนี้ เป็นการศึกษาหลักการในแนวทางของเครดิต สกอร์ริ่งและขั้นตอนการพัฒนาระบบช่วยสนับสนุนการอนุมัติสินเชื่อบัตรเครดิต ด้วยกระบวนการคาต้า ไมน์นิ่งแบบ Classification โดยอาศัยหลักการของ Decision Tree ตามแนวคิดของ SPRINT อัลกอริทึม

1.3.1 ข้อมูลที่นำมาวิเคราะห์

สำหรับแหล่งของข้อมูล (Data) ที่ใช้ในการวิเคราะห์นั้น ได้รับการเอื้อเฟื้อจาก ศูนย์ข้อมูลเครดิต ABC (ชื่อสมมุติที่ตั้งขึ้นสำหรับใช้อ้างอิงถึงศูนย์ข้อมูลเครดิตแห่งหนึ่งในประเทศไทย) ได้ก่อตั้งขึ้นมาเมื่อปี 2001 มีหน้าที่รวบรวมข้อมูลสินเชื่อบัตรเครดิต ประเภทบุคคลจากธนาคารพาณิชย์ต่าง ๆ ในประเทศไทย ที่ส่งเข้ามาที่ศูนย์ข้อมูลเครดิต ABC เพื่อรวบรวมข้อมูลและจัดเก็บเป็นระบบสอบถามข้อมูลและเปิดให้บริการแก่เจ้าหน้าที่อนุมัติสินเชื่อ ของธนาคารสมาชิกในการสอบถามข้อมูลสินเชื่อบัตรเครดิต สำหรับใช้เป็นข้อมูลประกอบการพิจารณาอนุมัติสินเชื่อบัตรเครดิตของเจ้าหน้าที่สินเชื่อ

ข้อมูลที่นำมาวิเคราะห์จะนำมาทั้งหมดโดยประมาณจำนวน 5,000 เรคคอร์ด เป็นข้อมูลย้อนหลังช่วงปี คริสต์ศักราช 2004 โดยเป็นข้อมูลเกี่ยวกับพฤติกรรมการจ่ายหนี้บัตรเครดิตประเภทบุคคล เนื่องจากเป็นข้อมูลส่วนบุคคล และเพื่อผลประโยชน์ขององค์กรในการ

ดำเนินการ ข้อมูลดังกล่าวจึงเป็นความลับ ดังนั้นจึงมีข้อมูลบางส่วนที่ไม่สามารถเปิดเผยออกไปได้

1.3.2 หน้าที่การทำงานของระบบ

ระบบทำการวิเคราะห์ข้อมูลในอดีตเพื่อสร้างโมเดลแบบจำลองต้นไม้ ด้วยหลักการของดาต้าไมน์นิ่ง แบบ Classification ที่สามารถใช้ในการพยากรณ์ข้อมูลนำเข้าใหม่เพื่อจัดชั้นของผู้ขออนุมัติสินเชื่อบัตรเครดิต ขอบเขตการทำงานหลักดังต่อไปนี้

- ระบบจะเปิดให้ผู้ใช้สามารถนำข้อมูลที่จะนำมาวิเคราะห์ได้ 2 ทางคือ วิเคราะห์จากฐานข้อมูล RDBMS ปัจจุบันเท่านั้น ซึ่งข้อมูลถูกเก็บอยู่บน Oracle Database 8i Server และการโหลดข้อมูลจาก Text File ซึ่งมีรูปแบบตามที่ผู้พัฒนาระบบกำหนดไว้เท่านั้น
- ระบบจะทำการวิเคราะห์ข้อมูลโดยใช้แนวคิด Classification โดยนำหลักการดิซิชั่นทรีและอัลกอริทึมที่ชื่อว่า SPRINT (Scalable PaRallelizable INduction of decision Trees) และแสดงผลลัพธ์ออกมาในรูปของแบบจำลองต้นไม้ และ Classification Rules เท่านั้น
- ในส่วนของการทดสอบผลของการวิเคราะห์แบบจำลองต้นไม้ นั้นจะใช้แนวคิดดังนี้คือ แบ่งข้อมูลออกเป็น 2 ส่วน ข้อมูลส่วนแรก (Training Data)ใช้ในการสร้างแบบจำลองต้นไม้ ข้อมูลส่วนที่สองจะถูกแบ่งไว้ใช้ในการทดสอบ (Testing Data) แบบจำลองต้นไม้ที่ระบบสร้างขึ้น

1.4 ขั้นตอนการดำเนินงาน

การพัฒนาระบบงานในการศึกษานี้สามารถจำแนกขั้นตอนการทำงานออกเป็นข้อๆ ได้ดังนี้

1. ศึกษาหลักการและกระบวนการทำงานของดาต้า ไมน์นิ่ง
2. ศึกษาหลักการและกระบวนการทำงานของเครดิต สกอร์ริ่ง
3. ศึกษาการนำดาต้า ไมน์นิ่งมาประยุกต์ใช้ในการพัฒนาระบบอนุมัติสินเชื่อบัตรเครดิตโดยใช้แนวทางของเครดิตสกอร์ริ่ง
4. ศึกษาอัลกอริทึม SPRINT เพื่อนำมาประยุกต์ใช้กับระบบ
5. รวบรวมและเตรียมข้อมูล รวมทั้งกำหนดเครื่องมือที่จะนำมาใช้ในการพัฒนาระบบ

5.1 การจัดเตรียมข้อมูล : จัดเตรียมข้อมูลให้เหมาะสมกับเป้าหมายการดำเนินงาน ทั้งปริมาณของข้อมูลก็ควรให้มีความพอเพียง โดยขั้นตอนของการจัดเตรียมข้อมูลมีดังนี้

5.1.1 การรวบรวมข้อมูล รวบรวมข้อมูลที่ระบบทดสอบ จากธนาคารต่าง ๆ ที่ส่งเข้ามาที่ระบบทดสอบ

5.1.2 การคัดเลือกข้อมูล การคัดเลือกข้อมูลแบ่งเป็น 2 ระดับคือ

- การคัดเลือกข้อมูลในระดับคอลัมน์ คือ พิจารณาดูว่าข้อมูลคอลัมน์ใดที่มีผลกับการวิเคราะห์
- การคัดเลือกข้อมูลในระดับแถว คือ การตรวจสอบความสมบูรณ์ของ Record ที่จะนำเข้ามาทดสอบว่ามีข้อมูลในส่วนใดที่ไม่สมบูรณ์ ต้องทำการปรับแก้ หรือคัดเลือกออก

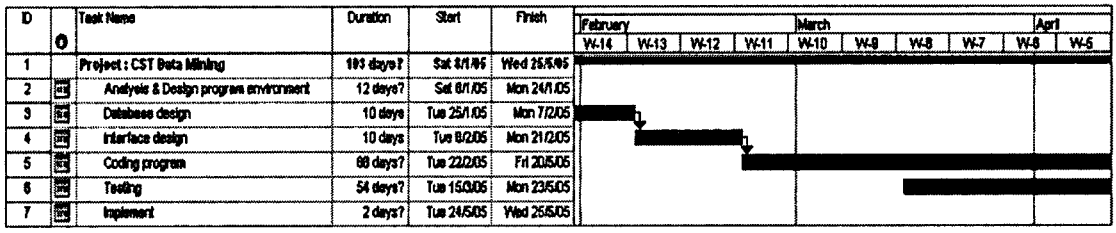
5.2 เลือกเครื่องมือที่ใช้ในการพัฒนาระบบงาน : ปัจจุบันการพัฒนาระบบงานหนึ่งๆ มีวิธีการและเครื่องมือมากมายให้เลือกใช้ ตามแต่ความเหมาะสม สำหรับโครงการพัฒนาระบบงานนี้ได้เลือกใช้เครื่องมือดังนี้

- เครื่องมือที่ใช้ในการพัฒนาโปรแกรม คือ J2SE
- ระบบฐานข้อมูลคือ Oracle 8i Server
- PL/SQL เป็นเครื่องมือที่ช่วยในการจัดเตรียมและทดสอบข้อมูลระหว่างการพัฒนา
- EditPlus2 สำหรับเปิดอ่านไฟล์
- เครื่องคอมพิวเตอร์ที่นำมาใช้ในการพัฒนา คือ Microsoft Windows XP Professional 2002 Pentium 4 CPU 3 GHz 512 MB

6. ออกแบบหน้าจอและฐานข้อมูล
7. พัฒนาระบบงานเพื่อใช้ในการทำเครดิต สกอร์ริ่ง
8. สรุปผลการศึกษา

1.5 แผนการดำเนินงาน/ ระยะเวลา

ระยะเวลาในการดำเนินการ โครงงาน เริ่มประมาณ 01/04/2005 - 20/05/05 โดยรายละเอียดของแต่ละช่วงเวลาของการดำเนินการมีดังต่อไปนี้



ID	Task Name	Duration	Start	Finish
1	Project : CST Beta Mining	183 days?	Sat 8/1/05	Wed 25/5/05
2	Analysis & Design program environment	12 days?	Sat 8/1/05	Mon 24/1/05
3	Database design	10 days	Tue 25/1/05	Mon 7/2/05
4	Interface design	10 days	Tue 8/2/05	Mon 21/2/05
5	Coding program	88 days?	Tue 22/2/05	Fri 20/5/05
6	Testing	54 days?	Tue 15/3/05	Mon 23/5/05
7	Implement	2 days?	Tue 24/5/05	Wed 25/5/05

1.6 ประโยชน์ที่คาดว่าจะได้รับ

การเก็บรวบรวมฐานข้อมูลสินเชื่อของลูกค้านำมาวิเคราะห์หาพฤติกรรมของข้อมูลสินเชื่อบัตรเครดิตรวมทั้งการพัฒนาระบบงานสนับสนุนการอนุมัติสินเชื่อบัตรเครดิตในโครงงานนี้ ก็เพื่อประโยชน์ดังต่อไปนี้

1. สามารถเข้าใจถึงขั้นตอนการทำดาต้า ไมน์นิ่ง ในรูปแบบของ Classification รวมทั้งหลักการของเครดิต สกอร์ริ่ง, การประยุกต์ดาต้า ไมน์นิ่งในการพัฒนาระบบงานทางด้านการอนุมัติสินเชื่อบัตรเครดิต
2. ระบบที่พัฒนาขึ้น สามารถช่วยในการตัดสินใจให้กับผู้ใช้งานในการพิจารณาอนุมัติสินเชื่อบัตรเครดิต
3. ระบบที่พัฒนาขึ้น สามารถแบ่งประเภทลูกค้าว่ามีความเสี่ยงในเรื่องของสินเชื่อ ปานกลาง หรือ สูง
4. ระบบที่พัฒนาขึ้น สามารถช่วยแบ่งกลุ่ม และวิเคราะห์ลูกค้าเพื่อที่จะผลิตและเสนอสินค้าตรงตามกลุ่มเป้าหมายแต่ละกลุ่ม
5. ระบบที่พัฒนาขึ้น สามารถช่วยทำนายว่าลูกค้าคนใดมีโอกาสมีความเสี่ยงสูงต่อการอนุมัติสินเชื่อบัตรเครดิต
6. ช่วยเพิ่มประสิทธิภาพในการทำงานของเจ้าหน้าที่สินเชื่อ และลดข้อผิดพลาดที่อาจเกิดขึ้นอันเนื่องมาจากเนื่องจาก
   - ขาดมาตรฐาน หรือหลักเกณฑ์ที่มาช่วยสนับสนุนการดำเนินงาน
   - ความหลากหลายของแหล่งที่มาของข้อมูล ที่มาจากหลายที่ด้วยกัน
7. การศึกษาและพัฒนานี้สามารถเป็นแนวทางในการประยุกต์ใช้ดาต้า ไมน์นิ่ง กับงานทางด้านอื่นๆ ได้ต่อไป

บทที่ 2
ทฤษฎีเครดิต สกอร์ริ่ง

ยุคที่อำนาจตัดสินใจอนุมัติวงเงินสินเชื่อที่สถาบันการเงินต่างๆ เริ่มนำระบบ "เครดิต สกอร์ริ่ง" มาใช้เป็นเครื่องมือสำคัญในการวิเคราะห์และพิจารณาว่า ลูกค้ารายใดควรได้รับสินเชื่อหรือไม่ ในวงเงินเท่าใด เครดิต สกอร์ริ่ง จึงทำหน้าที่เป็นเสมือนอีกหน่วยงานของสถาบันการเงิน เพื่อคำนวณ "แต้ม" ของลูกค้าทุกรายที่ยื่นคำร้องขอสินเชื่อเข้ามา แล้ววิเคราะห์ว่า ลูกค้าแต่ละรายมีระดับความเสี่ยงมากน้อยเท่าไร และสูงเกินกว่ามาตรฐานที่ตั้งไว้หรือไม่เป็นการช่วยสถาบันการเงินบรรเท่าความเสี่ยงจาก "หนี้เสีย" ที่อาจจะเกิดขึ้นได้ในอนาคต ขณะที่ชะตากรรมทางการเงินของลูกค้าในอนาคตก็จะถูกชี้สั่งด้วยระบบประมวลผลจากเครื่องคอมพิวเตอร์ รูปแบบการใช้บัตรเครดิตแต่ละครั้ง สถาบันการเงินผู้ออกบัตรจะรู้ทันทีว่า ลูกค้านิยมใช้บัตรเครดิตเพื่อซื้อสินค้าประเภทใดและบริการอะไร ฉะนั้นแล้ว การรวบรวมข้อมูลดังกล่าวเมื่อมาวัดเรทติ้ง หรือการคิดคะแนนด้วยระบบ "เครดิต สกอร์ริ่ง" ที่สถาบันการเงินทุกแห่งพยายามนำมาเป็นเครื่องมือ "บริหารความเสี่ยง" ระบบจะสามารถบอกได้ทันทีว่า ลูกค้ารายนั้นรายนี้จะมีศักยภาพในการผ่อนชำระค่างวดได้นานแค่ไหน หรือจะกู้ยืมได้ในวงเงินเท่าไร

ในส่วนของบทที่ 2 นี้จะกล่าวถึงความหมายและทฤษฎีของการพัฒนาเครดิต สกอร์ริ่ง เพื่อให้เข้าใจถึงแนวคิด วิธีการ และประโยชน์ที่จะได้รับจากการพัฒนาเครดิต สกอร์ริ่ง

2.1 ประวัติเครดิต สกอร์ริ่ง

เครดิต สกอร์ริ่ง ถูกพัฒนาขึ้นครั้งแรกในทศวรรษที่ 1950s แต่เพิ่งได้รับความนิยมใน 2 ทศวรรษที่ผ่านมา

2.2 ความหมายของเครดิต สกอร์ริ่ง

เครดิต สกอร์ริ่ง คือ เครื่องมือที่ช่วยให้ผู้มีหน้าที่อนุมัติสินเชื่อ (Creditor) สามารถตัดสินจะว่าผู้สมัครขอสินเชื่อสมควรที่จะได้รับการอนุมัติหรือไม่ และเป็นวิธีการที่ทำให้การอนุมัติทำได้รวดเร็วและยังลดความผิดพลาดได้อีกด้วย ในหลายปีที่ผ่านมาวิธีการในการพิจารณาอนุมัติสินเชื่อได้นำระบบเครดิต สกอร์ริ่งมาใช้ช่วยในการประกอบการตัดสินว่าผู้สมัครรายใดที่จะเป็นความเสี่ยงที่ดีในการให้สินเชื่อ ไม่ว่าจะเป็นการให้สินเชื่อ บัตรเครดิต การกู้ซื้อรถยนต์ ยิ่งไปกว่านั้นเครดิต

สกอร์ริ่งยังสามารถช่วยในการประเมินความสามารถและศักยภาพในการชำระการกู้จากการนำที่อยู่อาศัยมาจำนองอีกด้วย

เครดิต สกอร์ริ่งเป็นการนำข้อมูล (Information) และประวัติทางการเงิน (Credit experiences) ของผู้สมัคร ตัวอย่างเช่น การชำระค่าสาธารณูปโภค หรือ การชำระค่าสินค้าและบริการต่างๆ (bill-paying history) จำนวนบัญชีและประเภทบัญชีที่ผู้สมัครมีอยู่ การชำระล่าช้า หนี้คงค้าง และระยะเวลาของบัญชีที่ผู้สมัครมีอยู่ เป็นต้น ซึ่งข้อมูลต่างๆ เหล่านี้ได้มาจากแบบฟอร์มการสมัคร (Credit application) และ รายงานทางเครดิต (Credit report) จากองค์กรอ้างอิงข้อมูลทางเครดิต (Credit reference agencies) จากนั้นจะใช้โปรแกรมทางสถิติ ที่ผู้ที่ทำหน้าที่ในการอนุมัติสินเชื่อสามารถเปรียบเทียบข้อมูลเหล่านี้กับพฤติกรรมทางด้านสินเชื่อของลูกค้าที่ได้รับอนุมัติแล้วที่มีประวัติคล้ายคลึงกัน โดยระบบเครดิต สกอร์ริ่งจะให้ความสำคัญสำหรับแต่ละปัจจัย (factor) ที่ช่วยในการพยากรณ์ (predict) ว่าผู้สมัครรายใดที่จะมีความสามารถในการชำระหนี้ชำระอย่างตรงเวลา โดยผลลัพธ์ที่ได้จะอยู่ในรูปของคะแนน (Credit score)

วิธีการของเครดิต สกอร์ริ่งจะอยู่บนพื้นฐานข้อมูลจริงและวิธีการทางสถิติ ซึ่งเป็นวิธีการที่มีความน่าเชื่อถือมากกว่าวิธีการที่ใช้การตัดสินของผู้มีอำนาจในการอนุมัติ ซึ่งไม่มีระเบียบแบบแผนและกรอบในการพิจารณาก็อาจแตกต่างกันไปในในการพิจารณาผู้สมัครแต่ละราย ตามแต่ความเห็นของผู้พิจารณา ทำให้อาจเกิดความไม่เป็นกลางในการพิจารณาอนุมัติ เป็นเหตุให้ขาดความน่าเชื่อถือในวิธีการแบบเดิม

2.3 โมเดลเครดิต สกอร์ริ่ง

วิธีการโดยทั่วไปที่ใช้ในการพัฒนาโมเดลเครดิต สกอร์ริ่งนั้น จะใช้วิธีการวิเคราะห์ร่วมกับกระบวนการทางสถิติ ที่จะเปรียบเทียบรายละเอียดของผู้สมัครรายใหม่กับข้อมูลของผู้สมัครรายเก่าที่อนุมัติเรียบร้อยแล้ว ที่มีพฤติกรรมการชำระหนี้ตรงเวลา เพื่อดูว่าปัจจัยใดที่มีความเกี่ยวข้องต่อพฤติกรรมทางด้านสินเชื่อ โดยแต่ละสถาบันการเงิน หรือบริษัทที่ให้สินเชื่อมักมีโมเดลเครดิต สกอร์ริ่งในรูปแบบของตนเอง และโมเดลเครดิต สกอร์ริ่งก็จะแตกต่างกันในแต่ละประเภทของสินเชื่อ และขึ้นอยู่กับนโยบายขององค์กรหรือสถาบันนั้นๆ ซึ่งปัจจุบันมีบริษัทที่รับพัฒนาโมเดลเครดิต สกอร์ริ่ง เกิดขึ้นด้วยสาเหตุที่มีการพบว่า เครดิต สกอร์ริ่ง ที่ถูกสร้างขึ้นและมีประสิทธิภาพในองค์กรหนึ่ง อาจไม่มีประสิทธิภาพเพียงพอเมื่อนำไปใช้กับอีกองค์กรหนึ่ง ขึ้นอยู่กับปัจจัยแวดล้อมของแต่ละองค์กร นั่นแสดงให้เห็นว่า กระบวนการเครดิต สกอร์ริ่งไม่มีมาตรฐานที่ชัดเจน ปัญหาที่มักเกิดขึ้นจากการที่ไม่มีมาตรฐานที่ชัดเจนของเครดิต สกอร์ริ่งนี้ทำให้ไม่มีเป้าหมายที่ชัดเจน ต้องทำซ้ำๆ และ เกิดค่าใช้จ่าย ต้นทุนต่างๆ สูงแต่ก็ไม่สามารถรับประกันว่าจะได้แนวทางที่

ให้ประโยชน์สูงสุด ทำให้สถาบันการเงินมักไม่ลงทุนทำการพัฒนาเองแต่ว่าจ้างให้บริษัทที่มีความรู้ความสามารถโดยตรงรับผิดชอบดูแล

โดยทั่วไปเครดิต สกอร์ริ่งจะถูกประเมินจากข้อมูลต่างๆ เหล่านี้

1. ประวัติการชำระเงิน
2. มูลค่าหนี้ปัจจุบันเปรียบเทียบกับวงเงินสินเชื่อ
3. ประวัติทางด้านรายรับ
4. ระยะห่างในการสมัครขอสินเชื่อจากครั้งล่าสุด
5. จำนวนและประเภทของบัญชีสินเชื่อที่มีอยู่

ซึ่งโมเดลเครดิต สกอร์ริ่งยังสามารถสร้างขึ้นจากข้อมูลอื่นๆ นอกเหนือจาก 5 ข้อข้างต้น เช่น อาชีพ ระยะเวลาที่ว่าจ้างจนถึงปัจจุบัน หรือแม้แต่สถานภาพของตนเองต่อที่อยู่อาศัย เป็นต้น

โดยทั่วไปความหมายของคะแนนจะขึ้นอยู่กับโมเดลที่ใช้ในการแปลงความหมาย แต่โดยส่วนใหญ่ เมื่อคะแนนมากขึ้น จะหมายความถึงความเสี่ยงในการผิดนัดชำระหนี้จะลดลง

โมเดลเครดิต สกอร์ริ่งนั้น นอกจากจะพัฒนาได้ด้วยกระบวนการทางสถิติ ตามที่กล่าวข้างต้นแล้ว ยังสามารถนำเทคนิคดาต้า ไมน์นิ่งมาใช้ในการพัฒนาได้อีกด้วย ซึ่งหลักการ Classification เป็นเทคนิคหนึ่งของดาต้า ไมน์นิ่งที่นำมาใช้ในการพัฒนาโมเดลเครดิต สกอร์ริ่ง มากที่สุด ในการที่จะพยากรณ์ความเป็นไปได้ (probabilities) ในการผิดนัดชำระหนี้ของผู้ขอสินเชื่อ สำหรับวิธีการที่สามารถนำมาใช้ได้นั้นมีมากมายหลายวิธีการด้วยกัน ตามหลักการของ Classification ไม่ว่าจะเป็น linear และ logistic regression, decision trees, neural networks วิธีต่างๆ เหล่านี้มักถูกนำมาใช้ในการสร้างและพัฒนาโมเดลเครดิต สกอร์ริ่ง ซึ่งแนวโน้มในปัจจุบันและอนาคต การพัฒนาโมเดลเครดิต สกอร์ริ่งเพื่อให้ได้ผลลัพธ์ที่น่าเชื่อถือ นั้นกำลังได้รับความสนใจในการนำเทคนิคดาต้า ไมน์นิ่งเข้ามาช่วยในการพัฒนาเป็นอย่างมาก

2.4 การพัฒนาโมเดลเครดิต สกอร์ริ่ง

ขั้นตอนในการพัฒนาโมเดลเครดิต สกอร์ริ่งแบ่งออกได้เป็น 3 ขั้นตอนดังนี้

1. การนิยามปัญหาและจัดเตรียมข้อมูล

1.1 นิยามปัญหาทางธุรกิจ (Define the business problem)

1.1.1 ค้นหาปัญหาจากวัตถุประสงค์ทางธุรกิจ (Business Objective) ในการสร้างโมเดลสกอร์ริ่ง ตัวอย่างเช่น เพื่อคัดเลือกผู้สมัครของสินเชื่อที่มีคุณภาพ หรือแม้แต่ใช้ในการประเมินลูกค้าเดิมที่มีอยู่ว่ามีพฤติกรรมจัดอยู่ในประเภทใด

1.1.2 ใช้งานโมเดลเดรดิต สกอร์ริ่งในทางปฏิบัติจริงนั้น (Practical used)สามารถใช้ได้ด้วยวิธีการที่แตกต่างกัน ตัวอย่างเช่น โมเดลสกอร์ริ่ง ถูกออกแบบมาเพื่อทำการตัดสินใจให้โดยอัตโนมัติ หรือ โมเดลสกอร์ริ่ง ทำหน้าที่เป็นเพียงเครื่องมือช่วยในการสนับสนุนการตัดสินใจของผู้เชี่ยวชาญเท่านั้น

1.2 รวบรวมข้อมูลที่เกี่ยวข้อง (Collect relevant data)

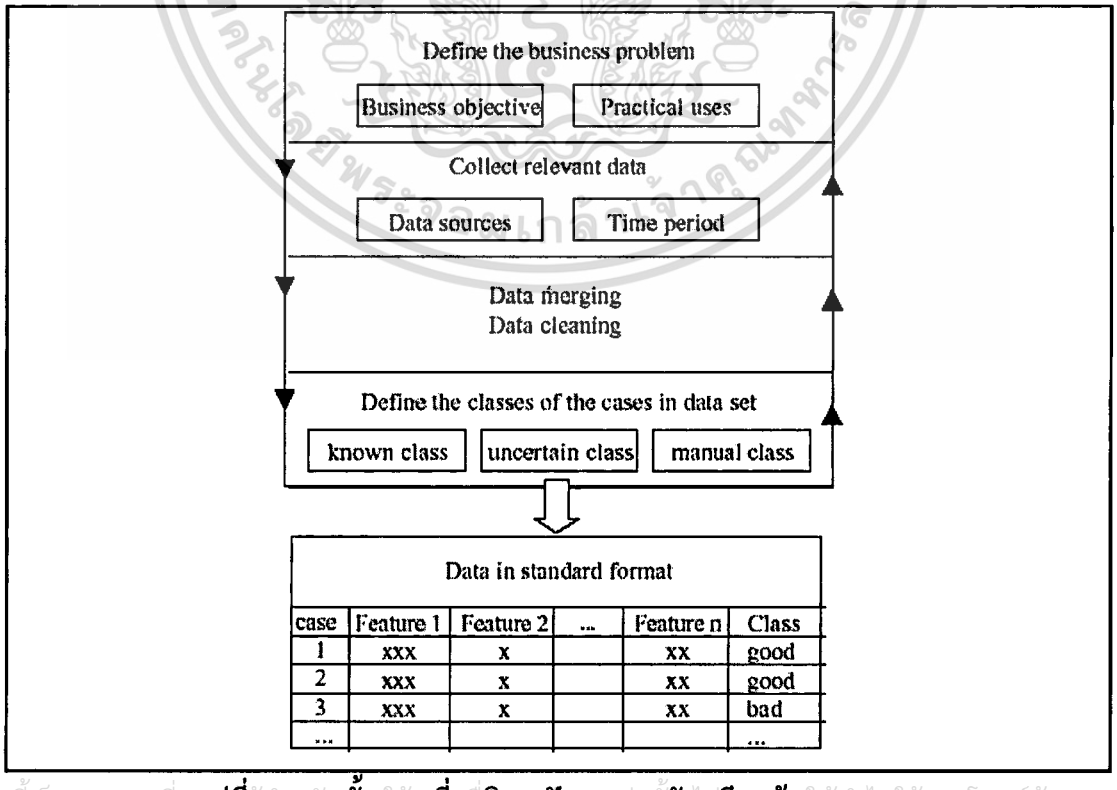
เป็นการรวบรวมข้อมูลที่เกี่ยวข้องกับปัญหา โดยที่โมเดลสกอร์ริ่ง จะเกี่ยวข้องกับข้อมูลในอดีต

1.3 รวมข้อมูลและทำความสะอาดข้อมูล (Merge and clean data)

ข้อมูลอาจมาจากแหล่งที่แตกต่างกันจึงต้องทำการจัดรูปแบบให้เป็นแบบเดียวกัน ซึ่งเป็นขั้นตอนหนึ่งของการทำดาต้าไมน์นิ่ง

1.4 กำหนดคลาส (Define the classes)

เป็นการกำหรดกลุ่มหรือคลาสให้กับข้อมูลที่นำมาใช้ในการสร้างโมเดล ตัวอย่างเช่น กำหนดให้ลูกค้าที่มีประวัติการชำระหนี้ล่าช้ากว่า 90 วันหรือ 30 วัน จัดอยู่ในกลุ่ม “ไม่ดี” เป็นต้น ในบางครั้งข้อมูลไม่สามารถระบุคลาสได้เนื่องจากไม่มีประวัติการชำระเงินประกอบการพิจารณา จึงต้องมีการวิเคราะห์โดยผู้เชี่ยวชาญเพื่อแบ่งแยกคลาสต่อไป



รูปที่ 1 แสดงขั้นตอนที่ 1 นิยามปัญหาและจัดเตรียมข้อมูล

2. การวิเคราะห์ข้อมูลและสร้างโมเดล

วัตถุประสงค์ของขั้นตอนนี้คือการสร้างโมเดลจากข้อมูลที่มีอยู่ โดยสามารถนำเทคนิคดาต้า ไมน์นิ่งมาใช้ได้ในขั้นตอนนี้ ซึ่งจะกล่าวในรายละเอียดทางด้านเทคนิคการทำงานของดาต้า ไมน์นิ่งต่อไป ในบทที่ 3

3. การนำโมเดลไปใช้และการตรวจสอบความน่าเชื่อถือของโมเดล

ในการนำโมเดลเครดิต สกอร์ริ่งไปใช้งานนั้นมีความแตกต่างกันในการปฏิบัติงานจริง แนวทางการนำไปใช้ในรูปแบบที่แตกต่างกัน มีดังต่อไปนี้

3.1 ใช้โมเดลเครดิต สกอร์ริ่งเป็นวิธีการวิเคราะห์สำหรับการตัดสินใจอนุมัติสินเชื่ออัตโนมัติ ซึ่งระบบประเภทนี้ จะเป็นระบบที่ใช้ในรูปแบบการให้สินเชื่อแก่ลูกค้าจำนวนมากและจำนวนเงินที่กู้ไม่มากนัก ตัวอย่างเช่น บัตรเครดิต การที่คนจะเข้ามาร่วมในการตัดสินใจการให้สินเชื่อนั้นจะจำเป็นเมื่อเป็นกรณีที่มีปัญหาเป็นรายบุคคลและเป็นกรณีสำคัญเท่านั้น

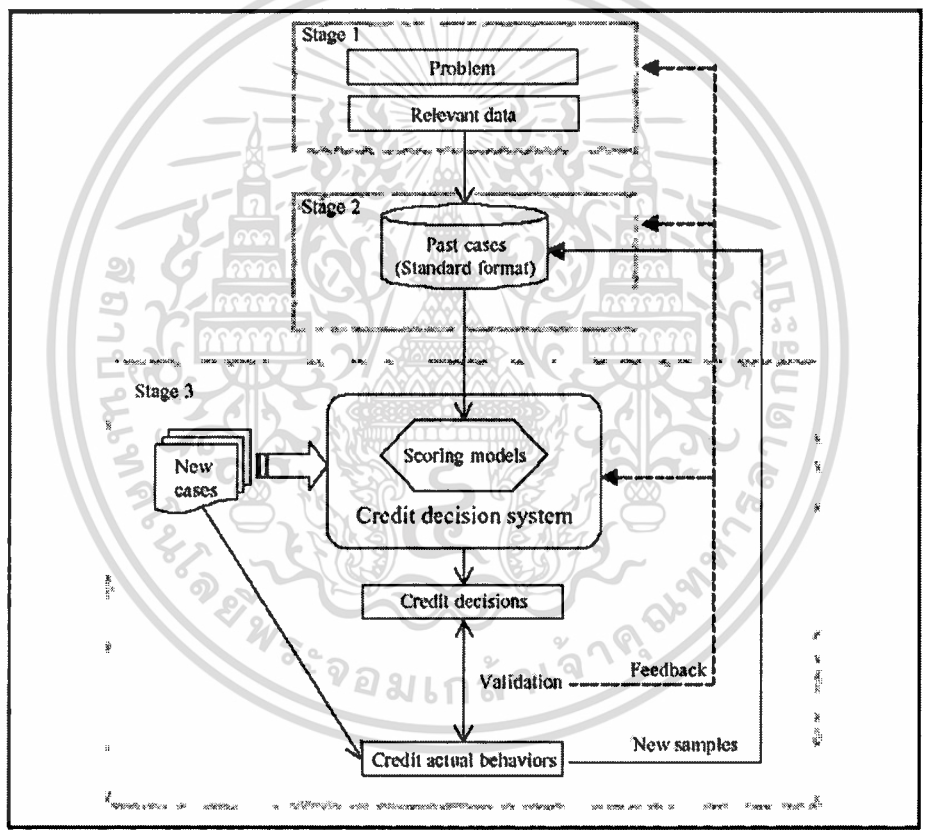
3.2 ใช้โมเดลเครดิต สกอร์ริ่งเป็นเครื่องมือในการวิเคราะห์และใช้ร่วมกับกระบวนการตัดสินใจหรือระบบการพิจารณาอนุมัติให้สินเชื่อ โดยระบบการพิจารณาอนุมัติการให้สินเชื่อนั้นอาจจะประกอบด้วยโมเดลสกอร์ริ่งและเครื่องมือวิเคราะห์อื่นๆ เช่น ระบบผู้เชี่ยวชาญ หรือ แม้แต่บุคคลากรที่มีประสบการณ์สูง ตัวอย่างเช่น กระบวนการอนุมัติที่ประกอบด้วย 2 ขั้นตอนในการให้สินเชื่อ โดยในขั้นแรก ผู้สมัครขอสินเชื่อจะถูกประเมินก่อนในขั้นแรกนี้ด้วยโมเดลสกอร์ริ่ง ถ้าผู้สมัครถูกจัดอยู่ในกลุ่ม “ดี” (“good”) ก็จะถูกยอมรับ (accept) แต่หากผู้สมัครถูกจัดอยู่ในกลุ่ม “ไม่ดี” (“bad”) จะถูกพิจารณาเป็นรายบุคลจากนักวิเคราะห์เครดิตที่มีประสบการณ์สูง ซึ่งถือว่าเป็นขั้นตอนที่สอง

3.3 ผลลัพธ์ที่ได้จากการทำโมเดลเครดิต สกอร์ริ่งจะถูกนำไปใช้เป็นข้อมูลนำเข้า (input) ของระบบการให้เรตติ้ง (rating system) ตัวอย่างเช่น มีการนำผลลัพธ์ที่ได้ตากโมเดลสกอร์ริ่งไปใช้ในการสร้างเรตติ้งสำหรับความเสี่ยงในการให้สินเชื่อ โดยจะถูกแบ่งออกเป็นหลายระดับด้วยกันขึ้นอยู่กับคะแนนที่ได้

ประเด็นที่สำคัญอีกประเด็นหนึ่งของการพัฒนาโมเดลไปสู่ระบบงาน (application) คือการสร้างโมเดลใหม่ (rebuilding) เนื่องจากโมเดลเสกอร์ริ่งที่สร้างจากข้อมูลที่คงที่ จะไม่ทันต่อเหตุการณ์ที่เปลี่ยนแปลงอยู่เสมอไม่ว่าจะเป็นทางด้านเศรษฐกิจ หรือ ด้านเงื่อนไขทางการตลาด เพื่อที่จะสร้างโมเดลให้มีความคงที่สำหรับประชากร (population) ที่ม่แน่นอนนั้น จำเป็นอย่างยิ่งที่จะต้องให้โมเดลสามารถสร้างขึ้นมาใหม่ได้โดยอัตโนมัติหลังจากเวลาผ่านไประยะหนึ่ง ซึ่งความถี่

ในการสร้างโมเดลใหม่นั้น ขึ้นอยู่กับปัญหาที่พบเกี่ยวกับผู้สมัคร เช่น อาจมีเครดิตทางการบริโภค หรือธุรกิจที่เปลี่ยนแปลงไป และความผันผวนของสภาวะเศรษฐกิจ เป็นต้น

ในการใช้งานโมเดลนั้น พฤติกรรมการชำระหนี้จะถูกบันทึกคดยอัตโนมัต ซึ่งข้อมูลเหล่านี้สามารถนำไปตรวจสอบโมเดลสกอร์ริ่ง ที่ใช้ว่ามีความถูกต้องน่าเชื่อถือมากน้อยแค่ไหนอีกด้วย ละมีปัจจัยอะไรเพิ่มเติมที่น่าจะนำเข้าสู่โมเดลจากผลลัพธ์ที่เกิดขึ้นจริง เพื่อนำมาปรับปรุงโมเดลให้มีความถูกต้องและทันสมัยยิ่งขึ้น จึงจะเห็นว่ากระบวนการพัฒนาระบบเครดิต สกอร์ริ่ง เป็นกระบวนการวนซ้ำ (iterative process)



รูปที่ 2 แสดงขั้นตอนของกระบวนการสร้างโมเดลเครดิต สกอร์ริ่ง

2.5 หลักการให้เครดิต

หลักการให้เครดิตต้องอาศัยปัจจัยต่าง ๆ ดังต่อไปนี้

2.5.1 ระยะเวลา – การทำสัญญาเครดิตจะเกิดขึ้นได้ต้องมีกำหนดเวลา วัน เดือน ปี ที่จะชำระแน่นอน โดยที่เจ้าหน้าที่และลูกหนี้จะต้องยินยอมทั้ง 2 ฝ่าย

2.5.2 ความมั่นใจของเจ้าหนี้ - เป็นสิ่งสำคัญที่เจ้าหนี้หรือผู้ให้เครดิตจะนำมาพิจารณาตัดสินใจว่าจะมีการทำสัญญาเครดิต หรือจะให้เครดิตแก่ลูกหนี้หรือไม่ ความมั่นใจพิจารณาจากปัจจัยต่าง ๆ 4 ประการ

1) อุปนิสัยใจคอลูกหนี้ (Character) - เป็นการวัดความรับผิดชอบของลูกหนี้ว่า ยินดีจะชำระหนี้เพียงใด
2) ความรู้ความสามารถที่จะหาเงินมาชำระคืนได้ (Capacity to pay) เป็นการวัดความสามารถของลูกหนี้ว่า สามารถหารายได้มาชำระคืนได้หรือไม่ ใช้เวลานานเท่าไร โดยพิจารณาจากวัตถุประสงค์ในการกู้ยืม โครงการและวิธีการใช้จ่ายเงิน ฐานการเงินในปัจจุบัน แหล่งเงินทุนหรือแหล่งรายได้จากที่อื่น และถ้าเป็นหนี้ส่วนบุคคล ย่อมพิจารณาถึงพื้นฐานการศึกษา หน้าที่การงาน อายุการทำงาน ตลอดจนความสามารถพิเศษอื่น ๆ
3) เงินทุน (Capital) เป็นการวัดความสามารถในการชำระหนี้ของลูกหนี้ได้อย่างหนึ่ง เป็นหลักประกันได้ว่า หากลูกหนี้ไม่มีความสามารถจะหารายได้มาชำระคืนได้ เจ้าหนี้ยังมีโอกาสรับการชำระคืนจาก เงินทุน หรือทรัพย์สิน ที่ลูกหนี้มีในปัจจุบัน
4) หลักทรัพย์ค้ำประกัน (Collateral) เป็นการสร้างความมั่นใจแก่เจ้าหนี้ นับเป็นสิ่งสำคัญในการให้กู้

ปัจจัยทั้งหมดนี้ เป็นสิ่งที่จะสร้างความมั่นใจแก่เจ้าหนี้ได้ว่า ควรจะให้เครดิตแก่ลูกหนี้หรือไม่ ซึ่งเรียกว่าเป็น "หลักในการพิจารณาให้เครดิต"

และในการพิจารณาทั้ง 4 ประการนี้ จำเป็นจะต้องพิจารณาประกอบไปพร้อม ๆ กัน ไม่ใช่พิจารณาแต่เพียงปัจจัยใดปัจจัยหนึ่ง และเนื่องจากปัจจัยทั้ง 4 ใช้ภาษาอังกฤษ ขึ้นต้นด้วย C ทุกตัว จึงเรียก C's of credit

จากที่กล่าวมาข้างต้นจึงต้องสร้างระบบที่มาช่วยในการวิเคราะห์ข้อมูล และสนับสนุนระบบการทำงานโดยสามารถสร้าง Model ที่สามารถนำมาวิเคราะห์ และจัดกลุ่มลูกค้า โดยอาศัยข้อมูลจากTransaction ต่าง ๆ ที่เข้ามา และสามารถจัดกลุ่มลูกค้าออกมาได้ดังนี้

Class G คือ Good Customer

Class B คือ Bad Customer

2.6 ประโยชน์ที่ได้รับจากเครดิต สกอร์ริ่ง

การนำเครดิต สกอร์ริ่งมาใช้ในกระบวนการอนุมัติสินเชื่อนั้นมีประโยชน์หลักๆ ดังนี้

1. เพิ่มประสิทธิภาพของกระบวนการประเมินพิจารณาให้กู้หรือการอนุมัติสินเชื่อ และลดภาวะเงินค้างชำระจำนวนมากจากลูกหนี้ อีกทั้งยังเพิ่มจำนวนลูกค้าที่ดีให้กับองค์กรอีกด้วย
2. เพิ่มเวลาให้กับพนักงานไปทำงานในส่วนอื่นได้มากขึ้น
3. ช่วยให้สามารถจัดโปรโมชั่นได้ตรงตามกลุ่มของผู้สมัครได้

บทที่ 3
ดาต้าไมน์นิ่ง

ดาต้าไมน์นิ่ง เป็นเทคโนโลยีที่ขยายมาจากเทคนิคทางสถิติร่วมกับเทคโนโลยีปัญญาประดิษฐ์ (Artificial Intelligence) และ Machine learning เพื่อสร้างรูปแบบ (Model) สำหรับช่วยตัดสินใจปัญหาทางธุรกิจ ดาต้าไมน์นิ่งมีอีกชื่อว่า "Knowledge Discovery in Database (KDD)" ดาต้าไมน์นิ่งเป็นกระบวนการค้นหาความรู้จากฐานข้อมูลที่มีอยู่ ต้องอาศัยข้อมูลจำนวนมากและพิจารณาความสัมพันธ์ระหว่างข้อมูลเหล่านั้น โดยนำแนวความคิดและเทคนิคต่างๆ คือ การนำหลักสถิติมาประยุกต์ใช้กับปัญญาประดิษฐ์, ฐานข้อมูล และหลักทางการตลาด หรือแนวความคิดอื่นเพื่อนำมาสร้างกฎและรูปแบบเพื่อนำไปวิเคราะห์ให้เกิดประโยชน์ต่อไป

3.1 ความเป็นมาของดาต้าไมน์นิ่ง

ในอดีตเมื่อเริ่มมีการเก็บข้อมูลด้วยฐานข้อมูล (Database) ในทศวรรษที่ 60 (1960s) จากนั้นเทคโนโลยีการเก็บข้อมูลด้วยฐานข้อมูลได้ถูกพัฒนาขึ้นมาอย่างต่อเนื่อง จนมาสู่ยุคข้อมูลข่าวสาร (1990s – 2000s) ข้อมูลมีจำนวนมาก แต่ไม่สามารถนำข้อมูลเหล่านั้นมาใช้ให้เกิดประโยชน์ (Data Explosion) ซึ่งข้อมูลมากมายที่มีอยู่อาจมีข้อมูลที่มีประโยชน์เพียงบางส่วน แต่เป็นข้อมูลส่วนที่มีประโยชน์อย่างมาก ดาต้าไมน์นิ่งจึงได้ถูกพัฒนาขึ้นเพื่อช่วยจัดการปัญหาดังกล่าว และได้รับความสนใจเป็นอย่างมาก

3.2 การทำงานของดาต้าไมน์นิ่ง (Data Mining Process)

ดาต้าไมน์นิ่งเป็นกระบวนการในการค้นหาแนวโน้ม และรูปแบบของข้อมูลที่ซ่อนอยู่ เพื่อสร้างความรู้ใหม่เกี่ยวกับข้อมูลนั้นๆ โดยใช้การวิเคราะห์ทางสถิติ และเทคนิคในการสร้างแบบจำลอง

ดาต้าไมน์นิ่งนำเอาวิธีการสร้างแบบจำลอง (Model) มาช่วยในการค้นหารูปแบบและความสัมพันธ์ของข้อมูล แบบจำลองเป็นเสมือนแบบจำลองของสถานการณ์จริง ซึ่งแบบจำลองที่ดีจะมีประโยชน์ในการทำความเข้าใจกับธุรกิจ และบอกได้ถึงสิ่งที่ควรปฏิบัติเพื่อทำให้เกิดความสำเร็จในธุรกิจ

3.3 รูปแบบการเก็บข้อมูลที่สามารถทำดาต้าไมน์นิ่ง

1. Relational Databases
2. Data Warehouses
3. Transactional Databases
4. Advanced Database Systems and Advanced Database Applications
5. Object-Oriented Databases
6. Object-Relational Databases
7. Spatial Databases
8. Temporal Databases and Time-Series Databases
9. Text Databases and Multimedia Databases
10. Heterogeneous Databases and Legacy Databases
11. The World Wide Web

3.4 รูปแบบข้อมูลตัวแปรในการทำดาต้าไมน์นิ่ง (Type of Attributes)

1. ตัวแปรแบบ Categorical จำแนกออกเป็น
   - 1.1 Nominal เป็นตัวแปรที่กำหนดความเป็นไปได้อย่างชัดเจน เช่น yes, no เป็นต้น
   - 1.2 Ordinal เป็นตัวแปรที่มีการจัดลำดับ เช่น อุณหภูมิ hot, mild, cool โดยที่ hot > mild > cool หรือ hot < mild < cool
   - 1.3 Interval เป็นตัวแปรที่ไม่เพียงแต่สามารถจัดลำดับเท่านั้นแต่สามารถวัดได้ในหน่วยที่เหมือนกัน เช่น อุณหภูมิ มักจะวัดกันในหน่วยของ องศาเซลเซียส มากกว่า hot, mild and cool จึงสามารถเปรียบเทียบได้เช่น 20 องศาเซลเซียส มีอุณหภูมิต่ำกว่า 22 องศาเซลเซียสอยู่ 2 องศาเซลเซียส
   - 1.4 Ratio ตัวแปรที่อยู่ในรูปแบบของสัดส่วน
2. ตัวแปรแบบ Quantitative จำแนกออกเป็น
   - 2.1 Continuous ค่าที่เก็บเป็นตัวเลขจำนวนจริง (Real Number) หรือค่าที่ต่อเนื่อง เช่น รายได้ เป็นต้น
   - 2.2 Discrete ค่าที่เก็บเป็นเลขจำนวนเต็ม (Integer) เช่นข้อมูลจำนวนพนักงานเป็นต้น

3.5 ขั้นตอนในการทำดาต้าไมน์นิ่ง

1. **Data cleaning** เป็นขั้นตอนในการกำจัด noise และ inconsistency data

2. **Data integration** ทำเมื่อมีการรวมข้อมูลจากหลายๆแหล่งข้อมูล
3. **Data selection** เมื่อมีข้อมูลที่เกี่ยวข้องกับสิ่งที่กำลังวิเคราะห์จะดึงข้อมูลเหล่านั้นขึ้นมาจากฐานข้อมูล
4. **Data transformation** เป็นขั้นตอนที่ทำการจัดรูปแบบข้อมูลให้อยู่ในรูปแบบที่เหมาะสมในการทำไมน์นิ่งตามอัลกอริทึมของดาต้าไมน์นิ่งที่เลือกใช้ เช่น การแปลงข้อมูลจากตัวแปรแบบ Quantitative ให้เป็นแบบ Categorical
5. **Data mining** เป็นขั้นตอนที่ทำการไมน์นิ่งข้อมูลเพื่อให้ได้รูปแบบ (Pattern) ของข้อมูลที่ซ่อนอยู่
6. **Pattern Evaluation** เป็นขั้นตอนที่ทำการเลือกและวิเคราะห์รูปแบบที่สนใจ
7. **Knowledge Representation** เป็นขั้นตอนที่ใช้เทคนิคของการนำเสนอความรู้ที่ได้จากการไมน์นิ่งให้ไปสู่ผู้ที่ใช้งานข้อมูลเหล่านั้น

3.6 ประเภทงานทางด้านดาต้าไมน์นิ่ง

เราสามารถจำแนกงานทางด้านดาต้าไมน์นิ่งออกเป็น 2 ประเภทคือ

1. Descriptive mining เป็นการหาลักษณะคุณสมบัติของข้อมูลในฐานข้อมูล อธิบายเป็นรูปแบบ (Pattern) เพื่อใช้เป็นรูปแบบในการตัดสินใจหรือวางแผนงานในอนาคตได้
2. Predictive mining เป็นการอ้างอิงจากข้อมูลเดิมที่มีอยู่ไปสู่การพยากรณ์ (prediction) ข้อมูลที่แตกต่างออกไป เช่น การนำข้อมูลประวัติการชำระลูกหนี้ของลูกหนี้มาสร้างแบบจำลอง (Model) เพื่อระบุลักษณะของลูกหนี้ที่อาจมีปัญหา

ความแตกต่างโดยพื้นฐานของงานทางด้านดาต้าไมน์นิ่งทั้ง 2 ประเภทข้างต้น คือ Predictive mining แสดงผลการพยากรณ์อย่างชัดเจน ในขณะที่ Descriptive mining สามารถนำมาใช้ในการทำ Predictive mining ได้อีกทีหนึ่ง

3.7 ประเภทของแบบจำลองสำหรับการทำดาต้าไมน์นิ่ง (Data mining Model)

3.7.1 Classification Model

Classification เป็นกระบวนการที่ใช้ในการค้นหาแบบจำลองที่จะใช้ในการพยากรณ์ (Prediction) โดยจะเป็นการจัดประเภทของสิ่งที่สนใจให้อยู่ในคลาส หรือ กลุ่ม ที่ได้กำหนดไว้ล่วงหน้า ซึ่งคลาสต้องเป็นเซ็ตของความเป็นไปได้ที่มีค่าแน่นอน Classification Model เป็นแบบจำลองที่สร้างขึ้นมาจากข้อมูลในอดีต โดยมีเฟสในการสร้าง 2 เฟส คือ

1. Training Phase เป็นการนำข้อมูลที่มีอยู่มาค้นหารูปแบบจำลองโดยผ่านอัลกอริทึมในเทคนิคของดาต้าไมน์นิ่งที่เลือกไว้ โดยข้อมูลที่ใช้เรียกว่า Training data
2. Test Phase เป็นการตรวจสอบความถูกต้องของรูปแบบที่ได้จาก Training phase

กระบวนการของการทำ Classification นี้ถูกจัดให้เป็น การเรียนรู้แบบมีเป้าหมาย (Supervised Learning) คือ มีการกำหนดรูปแบบ Input และ Output มาก่อนจากข้อมูลตัวอย่าง แบบจำลองที่ใช้ในการทำ Classification มีอยู่หลายรูปแบบ ที่รู้จักกันทั่วไป ได้แก่

1. Tree induction (Decision trees)

เป็นลักษณะของ Flow-Chart ที่มีโครงสร้างเหมือนต้นไม้ แต่ละโหนด (Node) จะเป็นตัวแทนของแอททริบิวท์โดยจะทดสอบค่าของแอททริบิวท์ในแต่ละโหนดและในแต่ละกิ่ง (branch) จะเป็นตัวแทนผลที่ได้จากการทดสอบ ส่วนใบ (leaves) ซึ่งเป็นโหนดในขั้นล่างสุด จะเป็นตัวแทนของคลาส ซึ่ง Decision trees เปลี่ยนเป็นแบบ Classification rules ได้ง่าย ตัวอย่างอัลกอริทึม ได้แก่ C4.5, CHAID, ID3

2. Mathematical formula

เป็นการใช้รูปแบบทางสถิติ (Statistical Model) หรือกระบวนการเชิงเส้นทางคณิตศาสตร์ (Linear Model) มักใช้กับคลาสที่เป็น numeric ตัวอย่างอัลกอริทึมประเภทนี้ เช่นLinear regression, Non-Linear regression, Logistic regression เป็นต้น

3. Neural Networks

เป็นรูปแบบที่เลียนแบบจากการทำงานของสมองมนุษย์ในการเรียนรู้และจัดการ โดยมีแนวคิดหลักคือ ปรับค่าถ่วงน้ำหนัก (Weight)

3.7.2 Clustering model

Clustering เป็นวิธีการในการจัดกลุ่มข้อมูลที่มีลักษณะคล้ายกันเข้าไว้ด้วยกัน ซึ่งสามารถเรียกอีกอย่างได้ว่าเป็นการทำ Segmentation โดยการจัดกลุ่มไม่ได้มีการกำหนดกลุ่มหรือคลาสไว้ล่วงหน้าแบบ Classification Model

3.7.3 Association Model

Association เป็นวิธีการที่ทำการวิเคราะห์หาความสัมพันธ์ของข้อมูลในรายการ (Transaction) เดียวกัน เป็นการค้นการความสัมพันธ์ที่ถูกซ่อนอยู่ ไม่เฉพาะความสัมพันธ์ของแอททริบิวท์ที่มีต่อคลาส เท่านั้น ยังหาความสัมพันธ์ระหว่างแอททริบิวท์ด้วยกันเองด้วย เช่น การวิเคราะห์จากรายการ

การซื้อสินค้าของลูกค้าในซูเปอร์มาร์เก็ต พบว่าส่วนใหญ่เมื่อลูกค้าซื้อผ้าอ้อม จะซื้อเบียร์ ด้วย เป็นต้น ตัวอย่างรูปแบบของกฎ Classification จะเป็นดังนี้

"IF condition1 THEN condition2" หรือ

"WHEN condition1 THEN condition2"

โดยที่ condition1 และ condition2 เกิดขึ้นพร้อมกันในรายการเดียวกันเรียก condition1 ว่า Rule Body หรือ Left-hand side หรือ Antecedent และเรียก condition2 ว่า Rule Head หรือ Right-hand side หรือ Consequent ซึ่งสามารถเรียก condition1 ว่า "เหตุ" และ condition2 ว่า "ผล" อย่างไรก็ดี Association rules จะใช้กับแอททริบิวท์ที่เป็น non-numeric attribute

3.8 SPRINT Algorithm

SPRINT [5] ย่อมาจากคำว่า Scalable PaRallelizable INduction of decision Trees เป็นอัลกอริทึมที่สามารถใช้ได้กับข้อมูลแบบตัวเลข (numeric attribute) และข้อมูลที่จัดเป็นหมวดหมู่ (categorical attribute) มีขั้นตอนหลักๆ เหมือนอัลกอริทึมของ Decision Tree ส่วนใหญ่คือ

1. Growth phase เป็นขั้นตอนการสร้าง tree จาก training data
2. Prune phase เป็นขั้นตอนการ prune เพื่อเพิ่มความถูกต้องให้แก่ tree สำหรับขั้นตอน test data

SPRINT เป็นอัลกอริทึมที่ถูกพัฒนาขึ้นมาช่วยแก้ปัญหาข้อจำกัดของหน่วยความจำที่ใช้ในการทำ Training phase และมีความเร็วและรองรับข้อมูลขนาดใหญ่ได้มากขึ้น โดยทั่วไปอัลกอริทึมของ Decision Tree มักมีข้อจำกัดในเรื่องของจำนวนข้อมูลที่สามารถทำงานได้ เนื่องจากจะเก็บข้อมูลไว้ในหน่วยความจำทั้งหมดขณะทำการสร้าง tree ซึ่ง SPRINT ถูกออกแบบมาให้สามารถทำงานในลักษณะคู่ขนานได้ (parallelization) ซึ่งทำให้สามารถรองรับการทำงานกับข้อมูลจำนวนมากได้ดี

หลักการของอัลกอริทึมใน Decision Tree คือจะทำการแบ่ง training data ออกเป็นส่วนๆ ของ tree โดยการแตกกิ่ง (split branch) เพื่อนำไปสู่ class ซึ่งเป็นแอททริบิวท์หนึ่งของ training set เรียกว่า classifying attribute ในที่สุด จุดมุ่งหมายคือ สร้างโมเดล tree ของ classifying attribute โดยขึ้นอยู่กับแอททริบิวท์ อื่นๆ ซึ่งการแบ่งแตกกิ่งก้านของ tree ออกไปนี้จะทำไปเรื่อยๆ จนกว่าจะได้ leaf node เป็น class ซึ่ง non-leaf node หรือ internal node คือ split point เป็นแอททริบิวท์ของข้อมูล training data ที่ผ่านวิธีการเลือกแล้ว โมเดลที่ได้สามารถนำไปใช้ในการหา class ของข้อมูลที่ยังไม่จัด class ได้ในอนาคต SPRINT ใช้วิธีการของ GINI index

3.8.1 Growth Phase

Tree ถูกสร้างขึ้นโดยวิธีการ recursive partitioning คือแบ่ง tree (split) ออกเป็นส่วนๆ (partition) ที่ node test โดยทำซ้ำไปเป็นรอบๆ จนกว่าแต่ละส่วนจะมีข้อมูลอยู่ในคลาสเดียวกัน โดยมีขั้นตอนการทำงานดังนี้

Partition(Data *S*)

```
if (all points in S are of the same class) then
        return;
for each attribute A do
        evaluate splits on attribute A;
Use best split found to partition S into S1 and S2;
Partition(S1)
Partition(S2)
```

ซึ่งมีหลักที่สำคัญอยู่ 2 ประการที่จะมีผลต่อประสิทธิภาพของ tree-growth phase คือ

1. วิธีการในการหาจุดแบ่ง (split point) ของ node test
2. วิธีการเลือก split point และจะแบ่งข้อมูลอย่างไร

สำหรับ SPRINT มีโครงสร้างและการทำงานแตกต่างจากอัลกอริทึมของ Decision Tree อย่าง CART หรือ C4.5 นั่นคือ ทำการเรียงลำดับข้อมูลเพียงครั้งเดียวเมื่อเริ่มต้น growth phase เท่านั้น (one-time sort)

3.8.1.1 Data Structure

โครงสร้างข้อมูลในรูปแบบของ SPRINT จะประกอบไปด้วย Attribute lists และ Histograms

Attribute lists

SPRINT เมื่อเริ่มต้นจะทำการสร้าง attribute list สำหรับแต่ละแอททริบิวท์ในข้อมูล แต่ละรายการใน Attribute lists เรียกว่า Attribute records ประกอบด้วย Attribute value, Class label และ Index of Record (rid) ในตอนเริ่มต้นเมื่อสร้าง List นั้นแอททริบิวท์ที่เป็น continuous attribute จะถูกเรียงลำดับด้วย Attribute value ซึ่งจะเป็นการเรียงลำดับเพียงครั้งเดียวเท่านั้นเมื่อเริ่มสร้าง list

Training data

rid	Age	Status	Class
0	23	Single	Good
1	17	Marriage	Good
2	43	Marriage	Good
3	68	Single	Bad
4	32	Divorce	Bad
5	20	Single	Good

Age Attribute List

Age	Class	rid
17	Good	1
20	Good	5
23	Good	0
32	Bad	4
43	Good	2
68	Bad	3

Status Attribute List

Status	Class	rid
Single	Good	0
Marriage	Good	1
Marriage	Good	2
Single	Bad	3
Divorce	Bad	4
Single	Good	5

ข้างต้นเป็นตัวอย่างการสร้าง Attribute list โดยมีแอททริบิวท์ 2 แอททริบิวท์ คือ Age กับ Status

3.8.1.2 Histograms

สำหรับ continuous attributes จะมี 2 histogram สำหรับแต่ละโหนดของ decision tree ที่กำลังถูกพิจารณาการ splitting โดยจะมี C_{above} และ C_{below} ที่ใช้ในการแสดงจำนวน class ของ Attribute record ในแต่ละโหนด โดยที่ C_{below} Attribute record ที่ได้ถูกประมวลผลไปแล้ว ในขณะที่ C_{above} คือ Attribute record ที่ยังไม่ถูกประมวลผล ส่วน categorical attributes จะมี histograms ที่เรียกว่า Count Matrix เราจะใช้ histograms ในการหา split point ด้วยวิธีการของ GINI index

Age	Class	rid	
17	Good	1	← Position 0
20	Good	5	
23	Good	0	
32	Bad	4	← Position 3
43	Good	2	
68	Bad	3	← Position 6

Cursor position 0 :

	G	B
C_{below}	0	0
C_{above}	4	2

Cursor position 3 :

	G	B
C_{below}	3	0
C_{above}	1	2

Cursor position 6 :

	G	B
C_{below}	4	2
C_{above}	0	0

รูปที่ 3 ค่า Histograms ของ continuous attribute

ข้างต้นเป็นการหา histograms ของ continuous attribute สำหรับ categorical attribute จะหา histograms ด้วยวิธี count matrix ดังต่อไปนี้

Status	Class	rid
Single	Good	0
Marriage	Good	1
Marriage	Good	2
Single	Bad	3
Divorce	Bad	4
Single	Good	5

Count matrix	G	B
Single	2	1
Marriage	2	0
Divorce	0	1

รูปที่ 4 ค่า Histograms ของ categorical attribute

การหาค่า histograms นี้สามารถนำไปใช้ในการทำ Gini index ได้

3.8.1.3 การหา Split points

ขณะที่ทำการสร้าง tree จุดหมายของแต่ละโหนดคือการการหา split point ที่ดีที่สุดที่จะเป็นตัวแบ่ง training record ออกเป็นแต่ละ leaf ค่าที่ได้ของ split point จะเป็นเครื่องบ่งชี้ถึงประสิทธิภาพในการแบ่ง class โดยในวิธีการของ SPRINT จะใช้ GINI index ในการหาค่า Split point

GINI index

สูตร

$$gini(S) = 1 - \sum p_j^2$$

เมื่อ

S เป็น data set ที่มีตัวอย่าง n คลาส

p_j เป็นค่าความถี่ของ class j ใน S

ตัวอย่างเช่น

มี 2 classes คือ Pos และ Neg และ p (Pos-elements), n (Neg-elements)

$p_{pos} = p/(p+n)$ $p_{neg} = n/(n+p)$

$gini(S) = 1- p_{pos}^2 - p_{neg}^2$

dataset S เมื่อ split ออกเป็น S_1 และ S_2

$gini_{SPLIT}(S) = (p_1+ n_1) / (p+n) * gini(S_1) + (p_2+ n_2) / (p+n) * gini(S_2)$

Split point ที่ดีที่สุดคือ มีค่า $gini_{SPLIT}$ น้อยที่สุด

ขั้นตอนในการหา split point ในการสร้าง tree มี 2 ขั้นตอนดังนี้

1. หาแอททริบิวท์ที่จะเป็น test node คือ เหมาะสมในการที่จะใช้เป็น split point ในการวิเคราะห์ข้อมูล โดยใช้สูตรของ Gini ที่กล่าวมาแล้วข้างต้น โดยเลือกแอททริบิวท์ที่มีค่า $gini_{SPLIT}$ น้อยที่สุด
2. สร้างจุดแบ่งโดยดูจากค่า $gini_{SPLIT}$ ที่ต่ำที่สุดมาเป็นตัวกำหนดจุดแบ่ง โดยวิธีการในการกำหนดจุดแบ่งนั้นขึ้นอยู่กับประเภทของแอททริบิวท์ดังนี้

   - การแบ่งข้อมูลแอททริบิวท์ที่เป็นประเภทตัวเลข (numeric / continuous attribute) เป็นการแบ่งลักษณะ 2 ทาง (binary split) จากที่กล่าวมาข้างต้น $A \leq v$ โดย v เป็นตัวเลขที่เป็นไปได้ของแอททริบิวท์ A ที่ถูกเรียงลำดับแล้ว ซึ่งจะอยู่ในรูปแบบ $v_1, v_2, \ldots, v_n$ เมื่อได้ค่า best split point ที่ v_1 ให้นำค่า $v_i + v_{1+1} / 2$ เป็น best split point

   - การแบ่งข้อมูลแอททริบิวท์ที่เป็นประเภทจัดหมวดหมู่ (categorical attribute) โดยให้ *S(A)* เป็นเซตของค่าที่เป็นไปได้ของแอททริบิวท์ A เมื่อ $X \subset$ domain(A) ดังนั้นจำนวนซับเซตที่เป็นไปได้เท่ากับ 2^n

3.8.1.4 การสร้าง Tree ด้วย SPRINT ในการทำ เครดิต สกอร์ริ่ง

ข้อมูลที่นำมาใช้เป็นข้อมูลตัวอย่างของลูกค้าที่เป็นสมาชิกบัตรเครดิต โดยประกอบด้วยแอททริบิวท์ดังนี้

- อายุ (Age)
- สถานภาพสมรส (Status)
- ชั้นหนี้ (Loan class) ซึ่งกำหนดค่าดังนี้

Class	Score
Good	> 201
Bad	≤ 200

1. เริ่มต้นมี training data ดังนี้

rid	Age	Status	Class
0	23	Single	Good
1	17	Marriage	Good
2	43	Marriage	Good
3	68	Single	Bad
4	32	Divorce	Bad
5	20	Single	Good

2. Initial attribute list สำหรับ node 0 โดยทำการเรียงลำดับค่าของแอททริบิวท์สำหรับ continuous attribute และเรียงตาม rid สำหรับ categorical attribute

Attribute list ใน node 0

Age Attribute List

Age	Class	rid
17	Good	1
20	Good	5
23	Good	0
32	Bad	4
43	Good	2
68	Bad	3

Status Attribute List

Status	Class	rid
Single	Good	0
Marriage	Good	1
Marriage	Good	2
Single	Bad	3
Divorce	Bad	4
Single	Good	5

1. เลือกแอททริบิวท์ที่จะหา best split point โดยใช้ Gini index โดยทำทุกๆ แอททริบิวท์

1.1 เริ่มจาก Age attribute ค่า histograms ที่เป็นไปได้ทั้งหมดคือ Age $\leq$ 17, Age $\leq$ 20, Age $\leq$ 23, Age $\leq$ 32, Age $\leq$ 43, Age $\leq$ 68

Age $\leq$ 17

	G	B
C_{below} Age $\leq$ 17	1	0
C_{above} Age $>$ 17	3	2

$G(\text{Age} \leq 17) = 1 - ((1/1)^2 + (0/1)^2) = 0$

$G(\text{Age} > 17) = 1 - ((3/5)^2 + (2/5)^2) = 0.48$

$G_{SPLIT} = (1/6)*0 + (5/6)*(0.48) = 0.4$

Age $\leq$ 20

	G	B
C_{below} Age $\leq$ 20	2	0
C_{above} Age $>$ 20	2	2

$G(\text{Age} \leq 20) = 1 - ((2/2)^2 + (0/2)^2) = 0$

$G(\text{Age} > 20) = 1 - ((2/4)^2 + (2/4)^2) = 0.5$

$G_{SPLIT} = (2/6) * (0) + (4/6) * (0.5) = 0.333$

Age $\leq$ 23

	G	B
C_{below} Age $\leq$ 23	3	0
C_{above} Age $>$ 23	1	2

$G(\text{Age} \leq 23) = 1 - ((3/3)^2 + (0/3)^2) = 0$

$G(\text{Age} > 23) = 1 - ((1/3)^2 + (2/3)^2) = 0.444$

$G_{SPLIT} = (3/6) * (0) + (3/6) * (0.444) = 0.222$

Age $\leq$ 32

	G	B
C_{below} Age $\leq$ 32	3	1
C_{above} Age $>$ 32	1	1

$G(\text{Age} \leq 32) = 1 - ((3/4)^2 + (1/4)^2) = 0.375$

$G(\text{Age} > 32) = 1 - ((1/2)^2 + (1/2)^2) = 0.5$

$G_{SPLIT} = (4/6)*(0.375) + (2/6)*(0.5) = 0.416$

Age $\leq$ 43

	G	B
C_{below} Age $\leq$ 43	4	1
C_{above} Age $>$ 43	0	1

$G(\text{Age} \leq 43) = 1 - ((4/5)^2 + (1/5)^2) = 0.32$

$G(\text{Age} > 43) = 1 - ((0/1)^2 + (1/1)^2) = 0$

$G_{SPLIT} = (5/6) * (0.32) + (1/6) * (0) = 0.266$

Age $\leq$ 68

	G	B
C_{below} Age $\leq$ 68	4	2
C_{above} Age $>$ 68	0	0

$G(Age \leq 68) = 1 - ((4/6)^2+(2/6)^2) = 0.444$

$G(Age > 68) = 1$

$G_{SPLIT} = (6/6)*(0.444)+(0/6)*(1) = 0.444$

1.2 หา Gini index จาก Status attribute ค่า histograms ที่เป็นไปได้ทั้งหมดเท่ากับ 2^n เมื่อ n คือ จำนวนค่าใน แอททริบิวท์

n = 3 (single, marriage, divorce)

Count matrix	G	B
Single	2	1
Marriage	2	0
Divorce	0	1

$G(Status \in \{Single\}) = 1-((2/3)^2+(1/3)^2) = 0.444$

$G(Status \in \{Marriage\}) = 1-((2/2)^2+(0/2)^2) = 0$

$G(Status \in \{Divorce\}) = 1-((0/1)^2+(1/1)^2) = 0$

$G(Status \in \{Single, Marriage\}) = 1-((4/5)^2+(1/5)^2) = 0.32$

$G(Status \in \{Single, Divorce\}) = 1-((2/4)^2+(2/4)^2) = 0.5$

$G(Status \in \{Marriage, Divorce\}) = 1-((2/3)^2+(1/3)^2) = 0.445$

$G_{SPLIT}(Status \in \{Single\}) = (3/6)*(0.444)+(3/6)*(0.445) = 0.4445$

$G_{SPLIT}(Status \in \{Marriage\}) = (2/6)*(0)+(4/6)*(0.5) = 0.333$

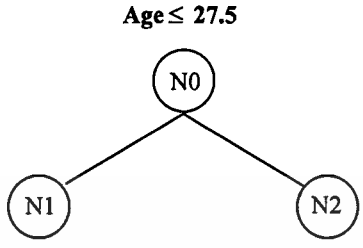
$G_{SPLIT}(Status \in \{Divorce\}) = (1/6)*(0)+(5/6)*(0.32) = 0.266$

$G_{SPLIT}(Status \in \{Single, Marriage\}) = (5/6)*(0.32)+(1/6)*(0) = 0.266$

$G_{SPLIT}(Status \in \{Single, Divorce\}) = (4/6)*(0.5)+(2/6)*(0) = 0.333$

$G_{SPLIT}(Status \in \{Marriage, Divorce\}) = (3/6)*(0.445)+(3/6)* (0.444) = 0.4445$

จะได้ค่า gini index ที่น้อยที่สุดคือ Age $\leq$ 23 ดังนั้น Best split point คือ (23+32)/2 = 27.5 คือ Age $\leq$ 27.5



รูปที่ 5 Decision Tree ผ่านการ Split ครั้งที่ 1

Attribute list ใน node 1

Age	Class	rid
17	Good	1
20	Good	5
23	Good	0

Status	Class	rid
Single	Good	0
Marriage	Good	1
Single	Good	5

Attribute list ใน node 2

Age	Class	rid
32	Bad	4
43	Good	2
68	Bad	3

Status	Class	rid
Marriage	Good	2
Single	Bad	3
Divorce	Bad	4

Status attribute แยกออกตาม rid ของ Age attribute ในแต่ละโหนด

จะเห็นว่าเรคคอร์ดใน node 1 เป็น Good ทั้งหมดแล้วจึงไม่ต้องแบ่งต่อ แต่เรคคอร์ดใน node 2 มีทั้ง Good และ Bad จึงทำการแบ่งต่อด้วยขั้นตอนต่อไปนี้

Age $\leq$ 32

	G	B
C_{below} Age $\leq$ 32	0	1
C_{above} Age $>$ 32	1	1

$G(\text{Age} \leq 32) = 1 - ((0/1)^2 + (1/1)^2) = 0$

$G(\text{Age} > 32) = 1 - ((1/2)^2 + (1/2)^2) = 0.5$

$G_{SPLIT} = (1/3)*0 + (2/3)*(0.5) = 0.333$

Age $\leq$ 43

	G	B
C_{below} Age $\leq$ 43	1	1
C_{above} Age $>$ 43	0	1

$G(\text{Age} \leq 43) = 1 - ((1/2)^2 + (1/2)^2) = 0.5$

$G(\text{Age} > 43) = 1 - ((0/1)^2 + (1/1)^2) = 0$

$G_{SPLIT} = (2/3)*(0.5) + (1/3)*(0) = 0.333$

Age $\leq$ 68

	G	B
C_{below} Age $\leq$ 68	1	2
C_{above} Age $>$ 68	0	0

$G(\text{Age} \leq 68) = 1 - ((1/3)^2+(2/3)^2) = 0.444$

$G(\text{Age} > 68) = 1$

$G_{SPLIT} = (3/3)*(0.444)+(0/3)*1 = 0.444$

Count matrix	G	B
Single	0	1
Marriage	1	0
Divorce	0	1

$G(\text{Status} \in \{\text{Single}\}) = 1-((0/1)^2+(1/1)^2) = 0$

$G(\text{Status} \in \{\text{Marriage}\}) = 1-((1/1)^2+(0/1)^2) = 0$

$G(\text{Status} \in \{\text{Divorce}\}) = 1-((0/1)^2+(1/1)^2) = 0$

$G(\text{Status} \in \{\text{Single, Marriage}\}) = 1-((1/2)^2+(1/2)^2) = 0.5$

$G(\text{Status} \in \{\text{Single, Divorce}\}) = 1-((0/2)^2+(2/2)^2) = 0$

$G(\text{Status} \in \{\text{Marriage, Divorce}\}) = 1-((1/2)^2+(1/2)^2) = 0.5$

$G_{SPLIT}(\text{Status} \in \{\text{Single}\}) = (1/3)*(0)+(2/3)*(0.5) = 0.333$

$G_{SPLIT}(\text{Status} \in \{\text{Marriage}\}) = (1/3)*(0)+(2/3)*(0) = 0$

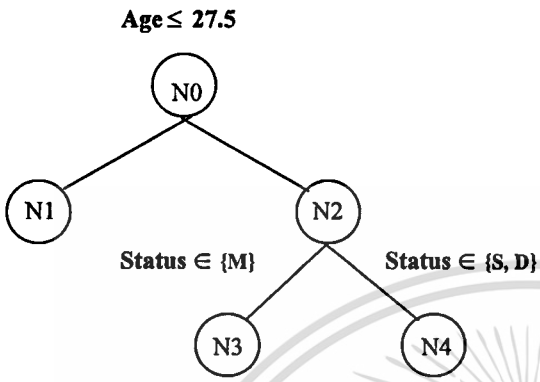
$G_{SPLIT}(\text{Status} \in \{\text{Divorce}\}) = (1/3)*(0)+ (2/3)*(0.5) = 0.333$

$G_{SPLIT}(\text{Status} \in \{\text{Single, Marriage}\}) = (2/3)*(0.5)+(1/3)*(0) = 0.333$

$G_{SPLIT}(\text{Status} \in \{\text{Single, Divorce}\}) = (2/3)*(0)+(1/3)*(0) = 0$

$G_{SPLIT}(\text{Status} \in \{\text{Marriage, Divorce}\}) = (2/3)*(0.5)+(1/3)*(0.444) = 0.481$

จะได้ค่า gini index ที่น้อยที่สุดคือ Status ∈ {Marriage} ดังนั้นจะได้ tree ดังรูป



รูปที่ 6 Decision Tree ผ่านการ Split ครั้งที่ 2

Attribute list ใน node 3

Status	Class	Rid
Marriage	Good	2

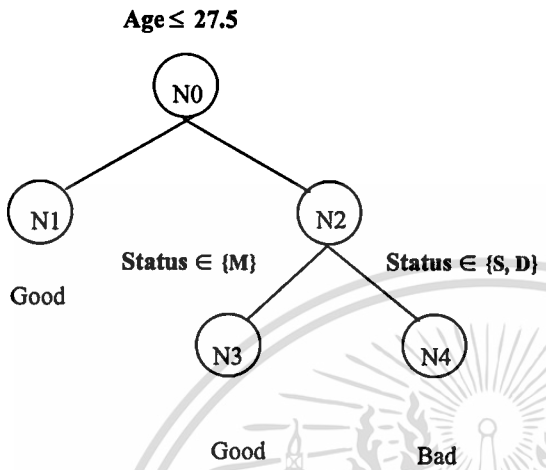
Age	Class	Rid
43	Good	2

Attribute list ใน node 4

Status	Class	Rid
Single	Bad	3
Divorce	Bad	4

Age	Class	rid
32	Bad	4
68	Bad	3

จะเห็นว่าใน node 3 มี class เป็น good เท่านั้นและใน node 4 มี class เป็น bad เท่านั้น จึงไม่ต้องทำการ split ต่อ ได้ tree ดังรูป



รูปที่ 7 Decision Tree ของข้อมูลตัวอย่างด้วย SPRINT

3.8.2 Prune Phase

เป็นขั้นตอนปรับแต่งกำจัดกิ่งของ tree ที่มีความผิดปกติ เนื่องจาก noise คือ ข้อมูลที่ไม่เกี่ยวข้องกับการวิเคราะห์หรือ outlier คือข้อมูลที่กระจายออกจากกลุ่มออกไปจาก training data โดยจะทำการตรวจสอบ Tree ที่ได้สร้างไปและเลือก Subtree ที่มีค่าความผิดพลาดน้อยที่สุด โดยใช้กระบวนการทางสถิติในการคำนวณหาค่าความผิดพลาด วิธีการทำ tree pruning มี 2 วิธีคือ

1. **Prepruning approach (stopping) :** เป็นวิธีที่ทำการ prune ในขณะที่ทำการสร้าง tree ในขั้นของ training phase หรือ growth phase
2. **Postpruning approach (pruning) :** เป็นวิธีที่ทำการ prune เมื่อได้โมเดล tree จาก training phase หรือ growth phase เรียบร้อยแล้วโดยทำการ prune ในลักษณะของการตัดกิ่ง tree ออก เหลือไว้เฉพาะ subtree ที่มีค่า estimated error rate ต่ำสุด โหนดที่อยู่ต่ำสุดที่ไม่ถูก prune จะกลายเป็น leaf โหนด และเป็น class ตามจำนวนความถี่ของข้อมูลที่มากที่สุดที่ตกอยู่ในโหนดนั้น

วิธีการหาค่าความผิดพลาดมีอยู่ 2 วิธี คือ

1. วิธีที่ใช้ training data set ที่ใช้ในการสร้าง tree ที่ต้องการ prune เรียกวิธีการ Cross Validation โดยใช้ training data มาสร้าง tree ย่อยๆ แล้วนำมาประเมินหาค่าความผิดพลาด

กับ tree ที่เราได้สร้างขึ้น วิธีการนี้มีความถูกต้องสูงแต่ใช้ทรัพยากรมากในการทำ จึงไม่เหมาะกับข้อมูลขนาดใหญ่

2. วิธีที่ใช้ training data set ใหม่ โดยจะแบ่ง training data ออกเป็น 2 ส่วน คือส่วนที่ใช้ในการสร้าง tree และส่วนที่ใช้ในการ prune โดยการ prune จะใช้ข้อมูลที่กระจายตามความเป็นจริง วิธีการนี้ถ้าเลือดข้อมูลไม่เหมาะสมจะทำให้ tree อาจลดขนาดลงอย่างผิดพลาดได้ สำหรับ SPRINT ใช้วิธีการ prune แบบ Prepruning approach ด้วยวิธีการ Minimum Description Length principle (MDL) ซึ่งเป็นวิธีการที่หา Subtree ที่ใช้จำนวน bit ในการ encode น้อยที่สุด

3.9 ประโยชน์ของดาต้าไมน์นิ่ง

กระบวนการดาต้าไมน์นิ่ง มีด้วยกันหลากหลายวิธีการ ขึ้นอยู่กับความเหมาะสมและของข้อมูลและผลลัพธ์ที่ต้องการ แต่ทุกวิธีการล้วนมีประโยชน์ในทำนองเดียวกันดังนี้

1. ตัวแบบที่ได้ง่ายต่อการเข้าใจ ผู้ที่ไม่มีความรู้พื้นฐานทางสถิติก็สามารถแปรความจากตัวแบบได้ สามารถนำสารสนเทศที่ได้ไปใช้ในกระบวนการทางธุรกิจได้
2. สามารถวิเคราะห์ข้อมูลจำนวนมากได้
3. ดาต้าไมน์นิ่งจะค้นพบในสิ่งที่เราคาดไม่ถึง เนื่องจากการที่มีตัวแบบและตรวจสอบที่หลากหลาย จะพบว่าการค้นหาจากตัวแปรที่นำมารวมกัน ทำให้ได้สารสนเทศที่เกี่ยวข้องกับธุรกิจ
4. สามารถนำดาต้าไมน์นิ่งมาใช้ในการค้นหารูปแบบและคำตอบที่ค้นหาจากข้อมูลที่มีอยู่ปัจจุบัน
5. สามารถนำดาต้าไมน์นิ่งมาใช้ในการพยากรณ์แนวโน้มในอนาคตจากข้อมูลที่มีอยู่ปัจจุบัน

บทที่ 4
ขั้นตอนการพัฒนาระบบการพิจารณาอนุมัติให้สินเชื่อบัตรเครดิต โดยใช้ดิซิชั่นทรี

หลังจากที่ได้ทำการศึกษาการทำงานของ SPRINT อัลกอริทึม ซึ่งเป็นอัลกอริทึมที่จะนำมาใช้ในการพัฒนาระบบพิจารณาอนุมัติให้สินเชื่อบัตรเครดิต ในบทที่ 3 แล้วนั้น ขั้นตอนต่อมาคือการทำความเข้าใจในหลักการของการให้เครดิตและการรวบรวมข้อมูล รวมทั้งการออกแบบหน้าจอต่างๆ การเลือกใช้เครื่องมือในการพัฒนา ก่อนที่จะทำการเขียนโปรแกรมเพื่อพัฒนาระบบต่อไป

4.1 การศึกษาความต้องการของระบบ

จากการศึกษาความต้องการของระบบนั้น พบว่าระบบนั้นต้องมีหน้าที่หลักในการทำงาน 6 หน้าที่ดังต่อไปนี้

1. ระบบต้องสามารถนำข้อมูลเข้าได้ 2 ทาง ดังนี้
   - เท็กซ์ไฟล์ (Text File) ซึ่งเท็กซ์ไฟล์นั้นจะต้องมีรูปแบบตามที่กำหนดดังนี้ ชื่อตัวแปร (Attribute) จะอยู่ในแถวแรกและข้อมูลแต่ละตัวจะขั้นด้วยเซมิโคลอน (;)
   - ข้อมูลจากฐานข้อมูล ซึ่งข้อมูลจะต้องมาจากฐานข้อมูลที่เป็น Oracle 8i Server เท่านั้น
2. ระบบต้องสามารถแบ่งข้อมูลออกเป็น 2 ส่วนจากแหล่งข้อมูลเดียวกัน
   - ส่วนที่ 1 คือ ส่วนของข้อมูลที่นำมาใช้ในการสร้างแบบจำลองต้นไม้ (Training Data)
   - ส่วนที่ 2 คือ ส่วนของข้อมูลที่นำมาใช้ทดสอบแบบจำลองต้นไม้ว่าที่เราได้สร้างขึ้น (Testing Data)
3. ผู้ใช้ระบบสามารถกำหนดเงื่อนไขในการสร้างแบบจำลองต้นไม้ได้ดังนี้
   - ผู้ใช้สามารถกำหนดระดับของแบบจำลองต้นไม้ที่จะแตกได้
   - ผู้ใช้สามารถกำหนดจำนวนข้อมูลที่น้อยที่สุดในแต่ละโหนดที่นำมาวิเคราะห์ได้
4. ระบบสามารถคำนวณผลการวิเคราะห์ได้ถูกต้องน่าเชื่อถือ ตามแนวคิดClassification โดยนำหลักการดิซิชั่นทรีและอัลกอริทึมที่ชื่อว่า SPRINT (Scalable PaRallelizable INduction of decision Trees) เข้ามาใช้
5. ระบบแสดงค่าความเชื่อมั่นเพื่อให้ผู้ใช้ระบบใช้ประกอบการตัดสินใจ โดยค่าความเชื่อมั่นมีวิธีในการคิดดังนี้

ค่าความเชื่อมั่น = $\frac{\text{จำนวนข้อมูลที่อยู่ในคลาสนั้น}}{\text{จำนวนข้อมูลในโหนดทั้งหมด}}$ x 100%

6. ระบบแสดงผลลัพธ์ให้ผู้ใช้ระบบสามารถเข้าใจได้ง่าย ซึ่งในที่นี้จะแสดงในรูปแบบจำลองต้นไม้ (Tree Model)

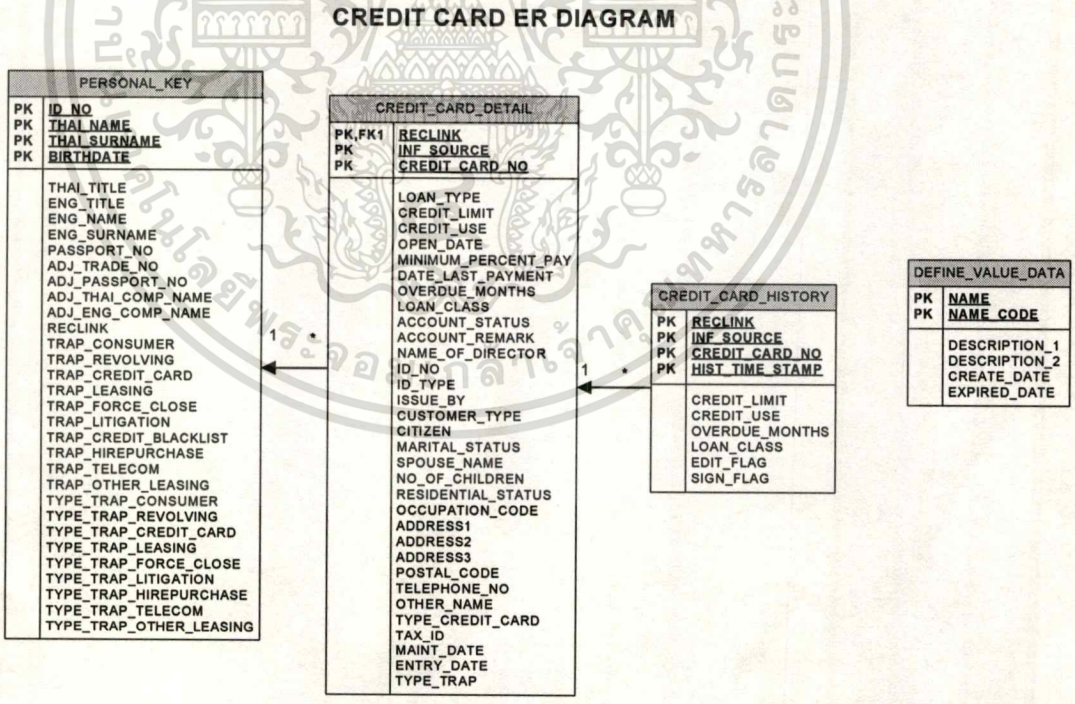
4.2 การวิเคราะห์และออกแบบระบบ

ในการวิเคราะห์และออกแบบระบบนั้นแบ่งเป็น 2 ส่วนดังนี้

- การวิเคราะห์และออกแบบข้อมูลที่ใช้ในการวิเคราะห์ในส่วนของฐานข้อมูล
- การวิเคราะห์และออกแบบส่วนของหน้าจอ

4.2.1 การวิเคราะห์และออกแบบข้อมูลที่ใช้ในการวิเคราะห์ในส่วนของฐานข้อมูล

โครงสร้างตารางต่าง ๆ ของระบบ Credit Card System ในฐานข้อมูลของบริษัท ABC แสดงได้ดังรูป



รูปที่ 8 : โครงสร้างความสัมพันธ์ของตารางต่าง ๆ ในระบบ Credit Card System

- **ตาราง personal_key** เป็นตารางหลักเก็บข้อมูลรายละเอียดของผู้ถือบัตรเครดิต เกี่ยวกับชื่อ นามสกุล หมายเลขบัตรประชาชน วันเดือนปีเกิดเป็นต้น ข้อมูลส่วนบุคคลหนึ่งคนจะ

เชื่อมกับตาราง credit_card_detail หลายเรคคอร์ด คือมีความสัมพันธ์เป็นแบบ one-to-many เนื่องจากคนหนึ่ง ๆ อาจถือบัตรเครดิตมากกว่าหนึ่งใบ

- **ตาราง credit_card_detail** เป็นตารางรองเก็บข้อมูลรายละเอียดเกี่ยวกับบัตรเครดิตแต่ละใบของผู้ถือบัตร ซึ่งมี Field Reclink,Inf_source,credit_card_no เชื่อมข้อมูลกับตารางหลักคือตาราง personal_key
- **ตาราง credit_card_history** เป็นตารางที่เก็บประวัติการชำระหนี้ของบัตรเครดิตแต่ละใบ โดยมี Field Reclink,inf_source,credit_card_no,hist_time_stamp ทำหน้าที่เชื่อมกับตาราง credit_card_detail
- **ตาราง define_value_data** เป็นตารางเก็บรายละเอียดคำอธิบายความหมายของข้อมูลแต่ละ Field

ข้อมูลที่ใช้ในการวิเคราะห์ ได้เลือกข้อมูลที่สำคัญและสามารถนำมาใช้ในการวิเคราะห์ได้ โดยให้ชื่อว่า MINING_TABLE มีโครงสร้างดังนี้

FIELD NAME	DATA TYPE	FIELD DESCRIPTION
ID_KEY	CHAR (24)	คีย์
AGE	NUMBER(2)	อายุ
CREDIT_LIMIT	NUMBER(14,2)	วงเงินที่ได้รับ
CREDIT USE	NUMBER(14,2)	จำนวนเงินที่ใช้ไป
OVERDUE MONTHS	CHAR (1)	จำนวนงวดหนี้(เดือน) ที่ลูกค้าค้างชำระ ณ. ปัจจุบันที่สมาชิกส่งข้อมูล
CLASS LABEL	CHAR (2)	การจัดชั้นหนี้ตามเกณฑ์ของ ธนาคารแห่งประเทศไทย
ACCOUNT STATUS	CHAR (3)	สถานะของ Account ในปัจจุบัน
MARITAL_STATUS	CHAR (1)	สถานภาพสมรส
NO_OF_CHILDREN	NUMBER(2)	จำนวนบุตร
RESIDENTIAL STATUS	CHAR (1)	สถานภาพของที่อยู่อาศัย
OCCUPATION CODE	CHAR (1)	อาชีพ
RID	NUMBER	RID สำหรับ SPRINT อัลกอริทึม

ตารางที่ 1: MINING_TABLE

- **AGE** (อายุผู้ถือบัตร) คำนวณจาก BIRTHDATE(วันเดือนปีเกิด)
- **CREDIT-LIMIT** (วงเงินที่ได้รับ) กฎในการสมัครบัตรเครดิตของธนาคารแห่งชาติระบุไว้ว่าผู้สมัครบัตรเครดิตจะต้องมีรายได้ 15,000 บาทขึ้นไป และโดยทั่วไป วงเงินสินเชื่อในการอนุมัติบัตรเครดิตจะอยู่ที่ประมาณ 1-2 เท่าของรายได้ ดังนั้นข้อมูล CREDIT-LIMIT ขั้นต่ำจะอยู่ที่ประมาณ 30,000 บาท
- **CREDIT-USE** (จำนวนเงินที่ใช้ไป)
- **OVERDUE_MONTHS** (จำนวนงวดหนี้ - เดือน) ที่ลูกค้าค้างชำระ ณ. ปัจจุบันที่สมาชิกส่งข้อมูล)
- **NO_OF_CHILDREN** (จำนวนบุตร)
- **RESIDENTIAL_STATUS** (สถานภาพของที่อยู่อาศัย) โดยมีความหมายดังนี้
    i. C คือ COMPANY PROVIDED-บริษัทจัดหาให้
    ii. F คือ MORTGAGE-ติดจำนอง
    iii. H คือ OWNS / BUYING- เป็นของตัวเอง หรือซื้อมา
    iv. P คือ LIVING WITH PARENTS OR RELATIONS- อาศัยอยู่กับครอบครัวหรือญาติ
- **OCCUPATION_CODE** (อาชีพ) ทำการจัดกลุ่มอาชีพออกเป็น 5 กลุ่มดังต่อไปนี้
    i. E – คือกลุ่มลูกค้าที่มีอาชีพ 1 – ลูกจ้าง
    ii. G - คือกลุ่มลูกค้าที่มีอาชีพ 2 – พนักงานรัฐวิสาหกิจ
        คือกลุ่มลูกค้าที่มีอาชีพ 3 – ข้าราชการ
    iii. M - คือกลุ่มลูกค้าที่มีอาชีพ 4 – ผู้มีกิจการเป็นของตนเอง
    iv. P - คือกลุ่มลูกค้าที่มีอาชีพ 5 – ผู้มีอาชีพอิสระ เช่น แพทย์, สถาปนิก
        คือกลุ่มลูกค้าที่มีอาชีพ 7 – PROFESIONAL
    v. C - คือกลุ่มลูกค้าที่มีอาชีพ 6 – ผู้มีรายได้จากค่าคอมมิชชั่นเป็นหลัก
        คือกลุ่มลูกค้าที่มีอาชีพ 8 – COMMISSION EARN

ส่วน Field ข้อมูลที่เป็น Class Label นั้น ใช้ข้อมูลจาก Loan Class มาทำการจัดกลุ่มลูกค้าออกเป็น 2 กลุ่มดังนี้

1. สำหรับข้อมูลที่มี Loan class = 01 และ 02 เป็น Class 'G'
2. สำหรับข้อมูลที่มี Loan class > 02 เป็น Class 'B'

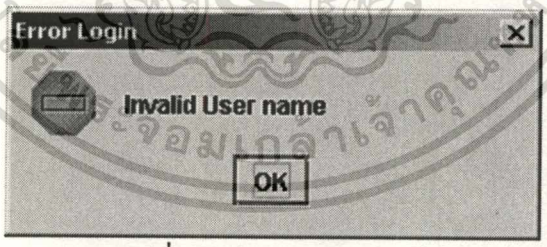
4.2.2 การวิเคราะห์และออกแบบส่วนของหน้าจอ

หน้าจอหลักของระบบนี้แบ่งออกเป็น 6 หน้าจอ

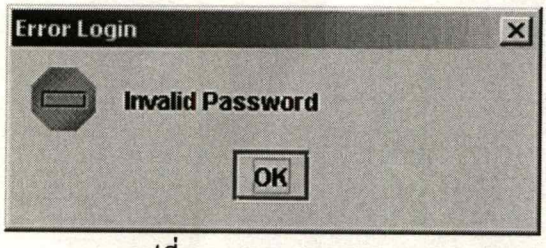
1. Logon เป็นหน้าจอสำหรับใส่ User / Password เพื่อ Logon เข้าสู่ระบบ

รูปที่ 9 Logon Screen

สำหรับหน้าจอนี้จะมี Text box User name สำหรับใส่ User name และ Text box Password สำหรับใส่ Password หากใส่ User name/ Password ไม่ถูกต้อง จะขึ้นหน้าจอ Dialog ดังนี้

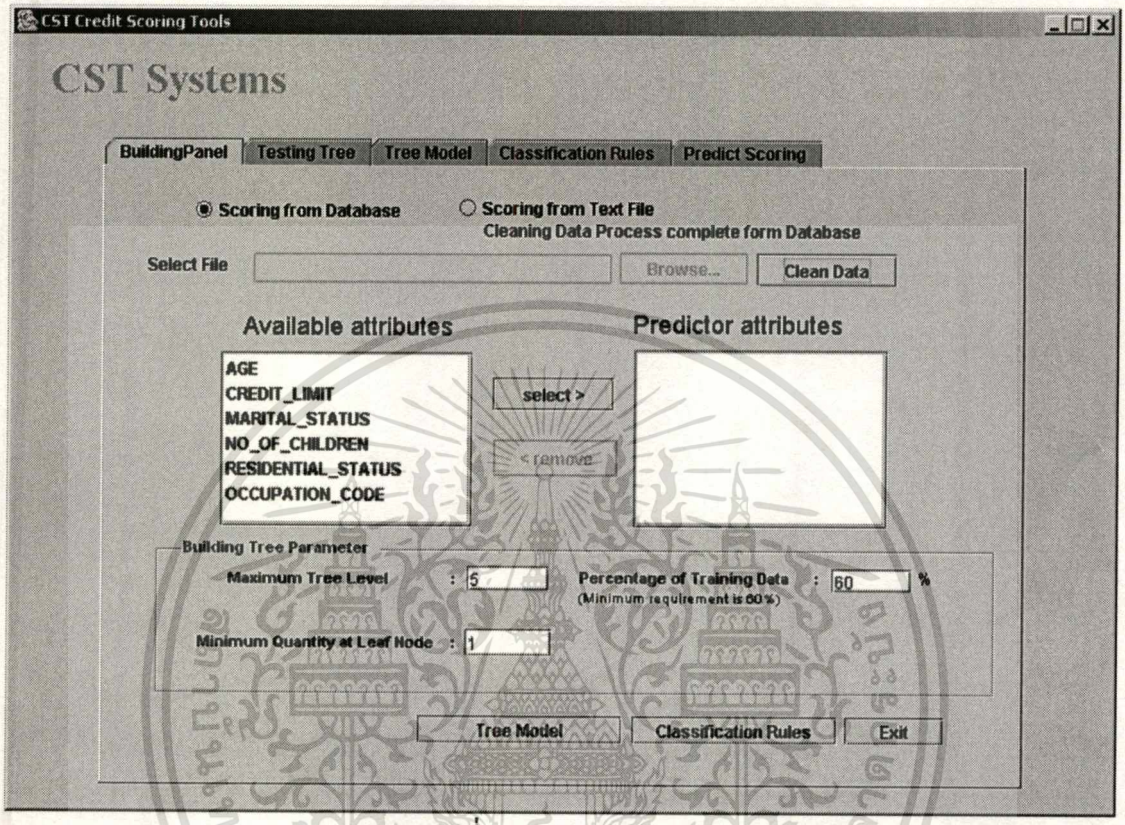


รูปที่ 10 Invalid User name



รูปที่ 11 Invalid Password

2. Building Panel เป็นหน้าจอแรกหลังจาก Logon เข้าสู่ระบบ



รูปที่ 12 Building Panel

สำหรับหน้าจอ Building Panel นี้เป็นหน้าสำหรับทำการ Build Tree Model ด้วย Training Data หน้าจอนี้ประกอบไปด้วย

- Radio Button Scoring from Database สำหรับเลือกว่าจะใช้ข้อมูล Input จากการอ่านข้อมูลใน Database
- Radio Button Scoring from Text File สำหรับเลือกว่าจะใช้ข้อมูล Input จากการอ่านข้อมูลจาก Text File โดยที่ Text File มีรูปแบบเฉพาะตามที่กำหนดไว้ ดังตัวอย่างต่อไปนี้

```
20031212055140000723759 9;26;50000.00;2000.00;0;01;A F;2;3;U;0;
20031212055141000723762 1;26;50000.00;60000.00;0;01;A 0;2;3;U;0;
20031212055139000723759 5;26;50000.00;0.00;0;01;A P;2;3;U;0;
20031215125117000723855 6;26;50000.00;3500.00;2;01;A C;2;3;U;0;
20031215125117000723855 6;26;50000.00;3500.00;2;01;A C;2;3;U;0;
20031212055143000723763 9;26;50000.00;0.00;0;01;A 0;2;3;U;0;
20031212055145000723766 2;26;50000.00;0.00;0;01;A 0;2;3;U;0;
20031212055143000723763 4;26;50000.00;0.00;0;01;A 0;2;3;U;0;
```

20031212055142000723762 5;26;50000.00;0.00;0;01;A 0;2;3;U;0;
20031212055142000723762 9;26;50000.00;0.00;0;01;A 0;2;3;U;0;
20031215125117000723856 9;26;50000.00;0.00;0;01;A X;2;3;U;0;
20031215125119000723858 9;26;50000.00;0.00;0;01;A 0;2;3;U;0;
20031215125117000723856 9;26;50000.00;0.00;0;01;A X;2;3;U;0;
20031215125117000723856 5;26;50000.00;0.00;0;01;A P;2;3;U;0;
20031215125117000723856 5;26;50000.00;0.00;0;01;A P;2;3;U;0;
20031215125120000723859 7;26;50000.00;0.00;0;01;A 0;2;3;U;0;
20031215125120000723859 7;26;50000.00;0.00;0;01;A 0;2;3;U;0;
20031215125120000723859 3;26;50000.00;0.00;0;01;A 0;2;3;U;0;
20031215125119000723858 9;26;50000.00;0.00;0;01;A 0;2;3;U;0;

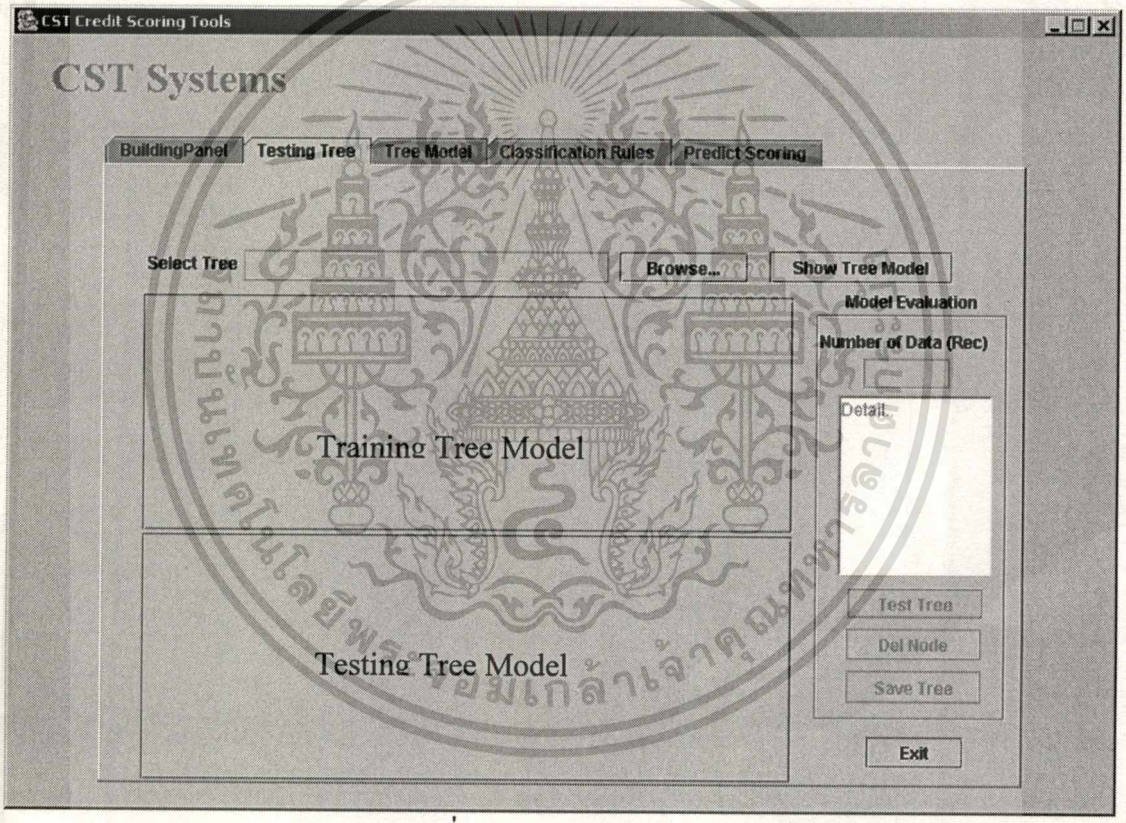
ตารางที่ 2 : ตัวอย่าง Text File ที่ใช้เป็น Input ในการ Mining

- Select File และ Browse... Button สำหรับเลือก Text File ที่เป็น Input กรณีเลือกนำเข้า Input จาก Text File
- Clean Data Button สำหรับ Clean Data ซึ่งต้องทำทุกครั้งทั้งแบบ Database และ Text File ก่อนการใช้ข้อมูลสร้างโมเดลทรี โดยมีเงื่อนไขในการ Clean Data ดังต่อไปนี้
  - ถ้าฟิลด์ loan class = null โปรแกรมจะทำการลบเรคคอร์ด
  - ถ้าฟิลด์ overdue_month = null โปรแกรมจะทำการลบเรคคอร์ด
  - ถ้าฟิลด์ birthdate = null โปรแกรมจะทำการลบเรคคอร์ด
- Available attributes จะแสดงฟิลด์ทั้งหมดที่สามารถนำมาใช้ในการสร้างโมเดลทรีได้
- Predictor attributes ฟิลด์ที่ถูกเลือกด้วยปุ่ม select> จาก Available attributes ทางซ้าย ซึ่งสามารถ remove ฟิลด์ที่ไม่ต้องการออกจาก Predictor attributes ได้ด้วยปุ่ม <remove ให้ฟิลด์ที่ถูก remove กลับไปอยู่ที่ Available attributes เหมือนเดิม
- Building Tree Parameter ประกอบด้วย 3 พารามิเตอร์ ดังนี้
  - Maximum Tree Level : ใช้ในการกำหนดระดับของแบบจำลองต้นไม้ที่สามารถแตกได้ซึ่งระดับ Root จะถือเป็นระดับที่ 0
  - Minimum Quantity at Leaf node : ใช้ในการกำหนดข้อมูลต่ำสุดของแต่ละโหนดที่สามารถนำมาแตกแบบจำลองต้นไม้ได้ ถ้ามีจำนวนข้อมูลน้อยกว่าค่าที่กำหนดไว้ระบบก็จะทำการหยุดแตกแบบจำลองต้นไม้
  - Percentage of Training Data : เป็นส่วนที่ใช้ในการกำหนดขนาดของข้อมูลที่ใช้ในการ Training เพื่อสร้างแบบจำลองต้นไม้ และเป็นการแบ่งข้อมูลไว้ส่วนหนึ่งเพื่อใช้เป็นข้อมูลที่ใช้ในการทดสอบแบบจำลองต้นไม้ ซึ่งการแบ่งข้อมูลไว้สำหรับ

ทดสอบนั้นจะคิดเป็นเปอร์เซ็นต์ของจำนวนข้อมูลทั้งหมด โดยขั้นต่ำของ Training Data กำหนดไว้ที่ 60% ของข้อมูล Input ทั้งหมด

- Tree Model Button เป็นปุ่มที่ทำการสร้างแบบจำลองต้นไม้ จาก Training Data
- Exit Button เป็นปุ่มสำหรับออกจากโปรแกรม

3. Testing Tree เป็นหน้าจอสำหรับทดสอบแบบจำลองต้นไม้ที่ได้จาก Building Panel

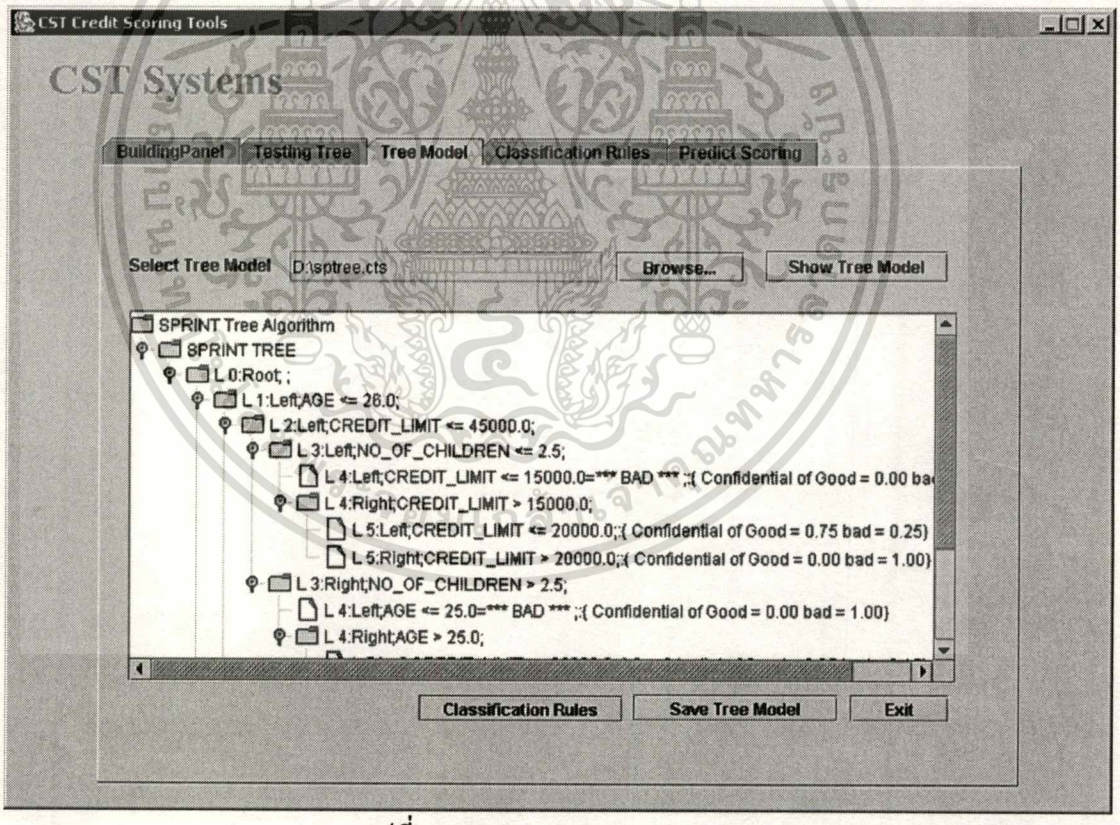


รูปที่ 13 Testing Tree Panel

- Select และ Browse... Button สำหรับเลือก Tree Model ที่บันทึกเก็บไว้มาทำการทดสอบ
- Show Tree Model Button สำหรับสั่งให้แสดง Tree ในกรอบแสดงแบบจำลองต้นไม้ Training Tree Model
- Model Evaluation แสดง
  - Number of Data (Rec.) คือ จำนวน record ทั้งหมดในแต่ละโหนดที่คลิ้กเลือกในโมเดลทรี

- กรอบ Detail แสดงจำนวน record ของแต่ละ Class Lable ในแต่ละโหนดที่คลิ้กเลือกในโมเดลทรี
- Test Tree Button สำหรับสั่งเริ่มต้นการ Test Tree
- Del Node Button สำหรับลบ leaf node ที่ User ต้องการลบเมื่อเห็นว่าเป็นโหนดที่ไม่มี testing data มาตกหรือไม่สมเหตุสมผลในการวิเคราะห์
- Save Tree สำหรับบันทึก Tree ที่ผ่านการทดสอบและ Prune โดย user เรียบร้อยแล้ว โดยจะบันทึก Tree ในรูปของไฟล์ใน Drive C:\Tree> ชื่อ "sptesttree.cst"
- Exit Button สำหรับออกจากโปรแกรม

4. Tree Model เป็นหน้าจอสำหรับแสดงแบบจำลองต้นไม้ที่ได้จาก Building Panel หรือเลือกแสดงจาก Tree ที่ถูกบันทึกไว้



รูปที่ 14 Tree Model Panel

- Select และ Browse… Button สำหรับเลือก Tree Model ที่บันทึกเก็บไว้มาแสดง
- Show Tree Model Button สำหรับสั่งให้แสดง Tree ในกรอบแสดงแบบจำลองต้นไม้
- Classification Rules Button สำหรับ Generate Classification Rules และแสดงใน หน้าจอ Classification Rules

- Save Tree Model สำหรับบันทึก Tree ในรูปของไฟล์ใน Drive C:\Tree> ชื่อ "sptraintree.cst" และมีโครงสร้างดังนี้

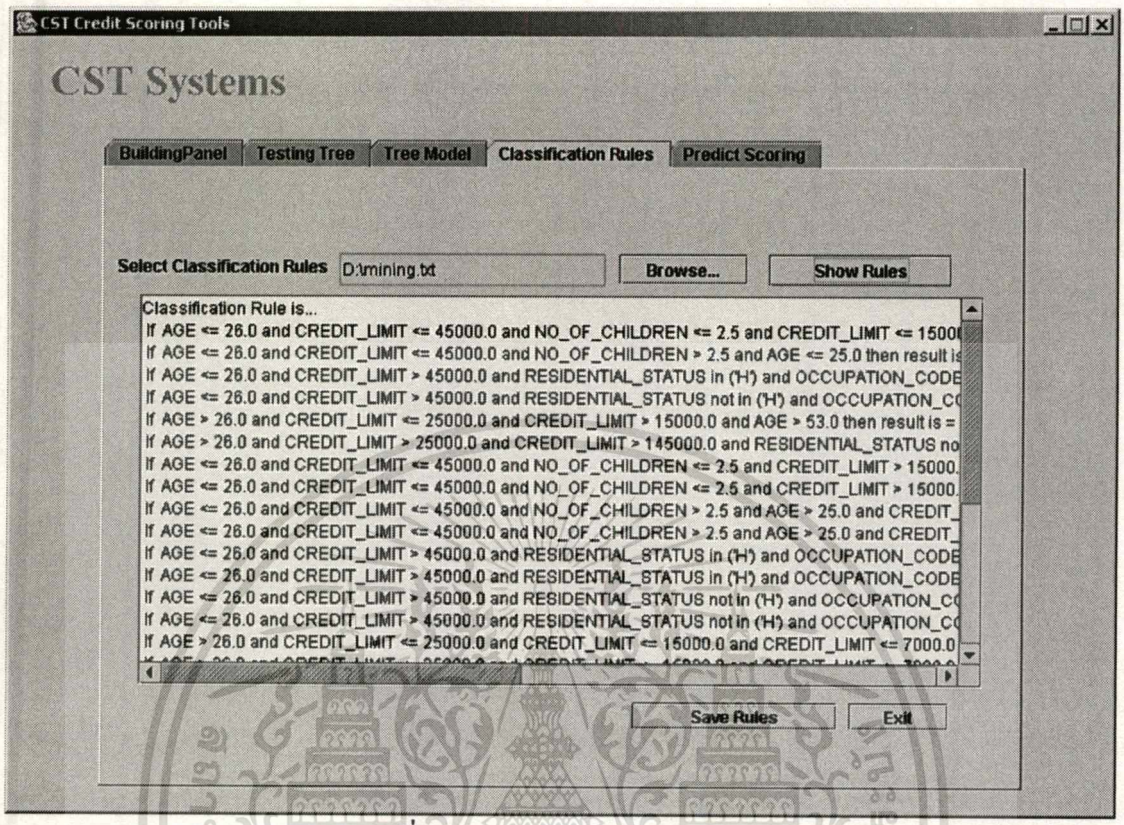
```
public class TreeInterface {
    public int INDEX= 1;  // o = root
    public int LEVEL =1;  // level of node 1 = root
    public int UPINDEX =1;  //  upper node index  1 = root
    public String  NAME = " "; // node name
    public String  DESCRIPTION = " "; // split point of node (parameter)
    public String  SPLITVALUE = " "; // min values of node (parameter)
    public String  WHERECLAUSE = " "; // where clause of node (parameter)
    public boolean FINISH = true; // have parant flag of node (parameter)
    public String LOANCLASS = " "; // LOAN_CLASS (parameter)
    public String CONFIDENTIAL = " "; // Confidential (parameter)
    public double CON_01 = 0; // of good
    public double CON_02 = 0; // of bad
    public String ACCURENCY = " "; // ACCURENCY (parameter)
    public int rec = 0; // number of rec
    public String detail = " "; // number of detail
    public int indexDel = 0; // for remove only (index)
}

1;0;0;L 0:Root; ;26.0; WHERE AGE <= 26.0;false; ; ;0.1822721598002497;0.8177278401997503; ;801; Detail.. /n Good = 146/n Bad = 655;
2;1;1;L 1:Left;AGE <= 26.0;26.0; WHERE AGE <= 26.0;false; ; ;0.08333333333333333;0.9166666666666666; ;36; Detail.. /n Good = 3/n Bad = 33;
3;1;1;L 1:Right;AGE > 26.0;26.0; WHERE AGE > 26.0;false; ; ;0.1869281045751634;0.8130718954248366; ;765; Detail.. /n Good = 143/n Bad = 622;
```

ตารางที่ 3 โครงสร้างและตัวอย่างแบบจำลองต้นไม้ที่ทำการบันทึกเป็นไฟล์

- Exit Button สำหรับออกจากโปรแกรม

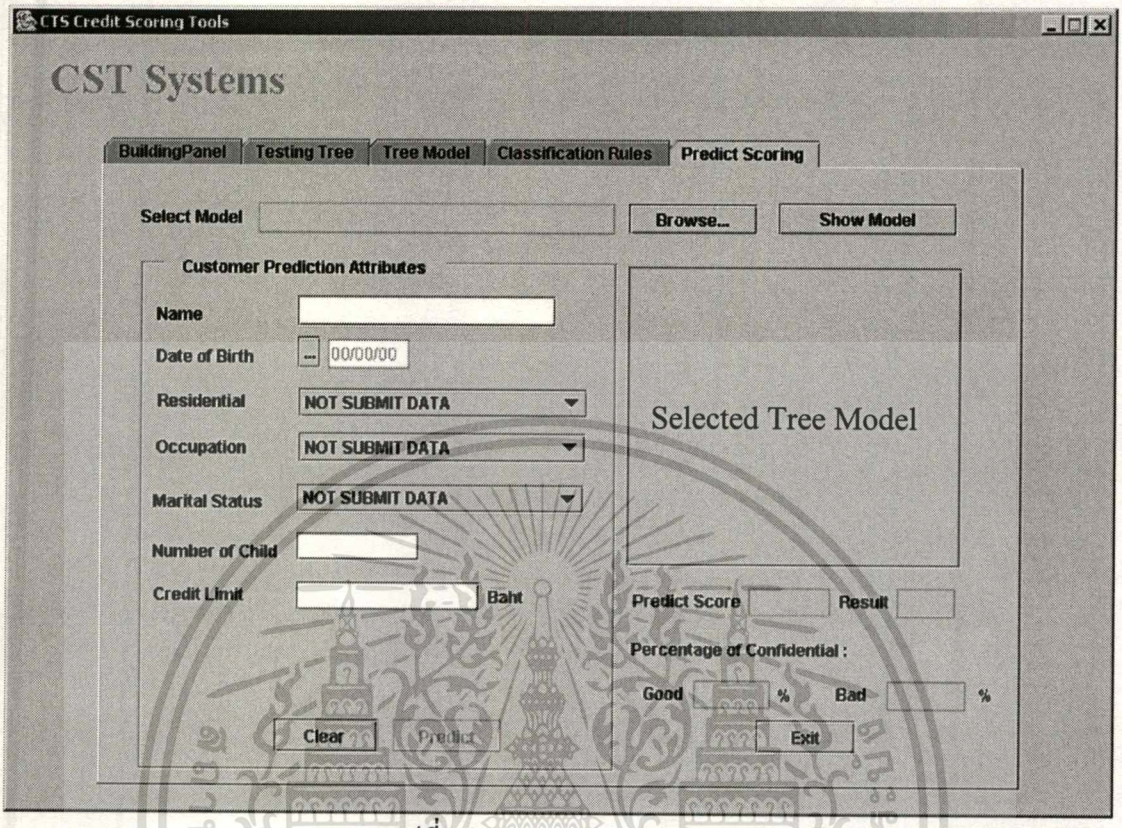
5. Classification Rules เป็นหน้าจอสำหรับแสดง Rules ของแบบจำลองต้นไม้



รูปที่ 15 Classification Rules Panel

- Select และ Browse... Button สำหรับเลือก Classification Rules ที่บันทึกเก็บไว้มาแสดง
- Show Rules Button สำหรับสั่งให้แสดง Rules ในกรอบแสดง Classification Rules
- Save Rules Button สำหรับบันทึก Classification Rules ในรูปของไฟล์ใน Drive C:\Rules> ชื่อ "classrules.txt"
- Exit Button สำหรับออกจากโปรแกรม

7. Predict Scoring เป็นหน้าจอสำหรับพยากรณ์จัดกลุ่มให้ผู้สมัครขอสินเชื่อบัตรเครดิต โดยนำข้อมูลเข้าแบบจำลองต้นไม้ที่เลือก



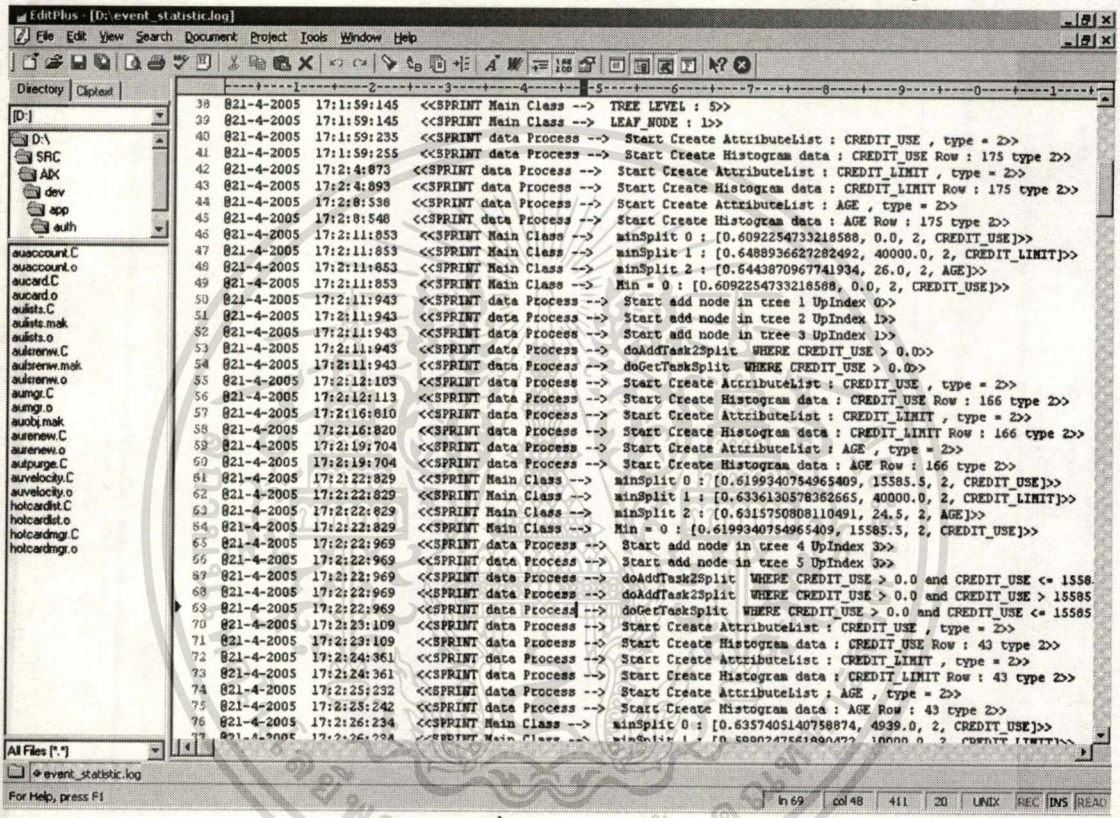
รูปที่ 16 Predict Scoring Panel

- Select และ Browse... Button สำหรับเลือก Tree Model ที่บันทึกเก็บไว้มาแสดง
- Show Model Button สำหรับแสดงแบบจำลองต้นไม้
- Customer Prediction Attributes เป็นข้อมูลที่ใช้ในการพยากรณ์ มีดังนี้
  - ชื่อ – นามสกุล (Name)
  - วันเดือน ปีเกิด เพื่อใช้คำนวณหาอายุ (Date of Birth)
  - สถานะที่อยู่อาศัย (Residential)
  - อาชีพ (Occupation)
  - สถานภาพ (Marital Status)
  - จำนวนบุตร (No of Children)
  - วงเงิน (Credit limit)
- Clear Button สำหรับ reset ค่า input
- Predict Button สำหรับพยากรณ์ ค่าต่างๆ ดังนี้
  - Predict Score เป็นคะแนนที่ให้ โดยคำนวณจาก Confidential of Class Good*100
  - Result เป็น Class ที่มีค่า Confidential ที่มากกว่า

- Percentage of Confidential Class Good/ Bad ค่า Confidential คิดเป็น % ของทั้ง 2 Class

- Exit Button สำหรับออกจากโปรแกรม

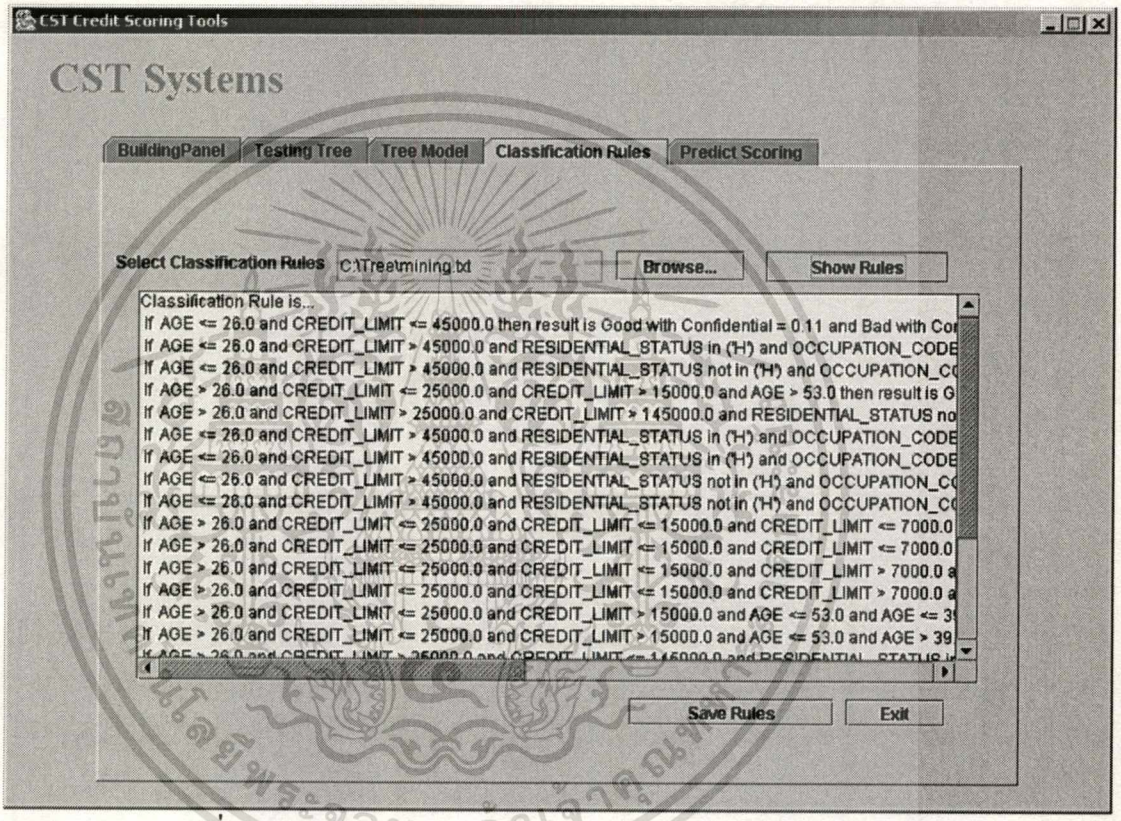
8. Event Log สำหรับบันทึก log ต่างๆ เพื่อการวิเคราะห์และ track กลับเมื่อมีปัญหา



รูปที่ 17 Event Log

บทที่ 5
สรุปผลการทดสอบ/ข้อเสนอแนะ

5.1 สรุปผลการทดลอง

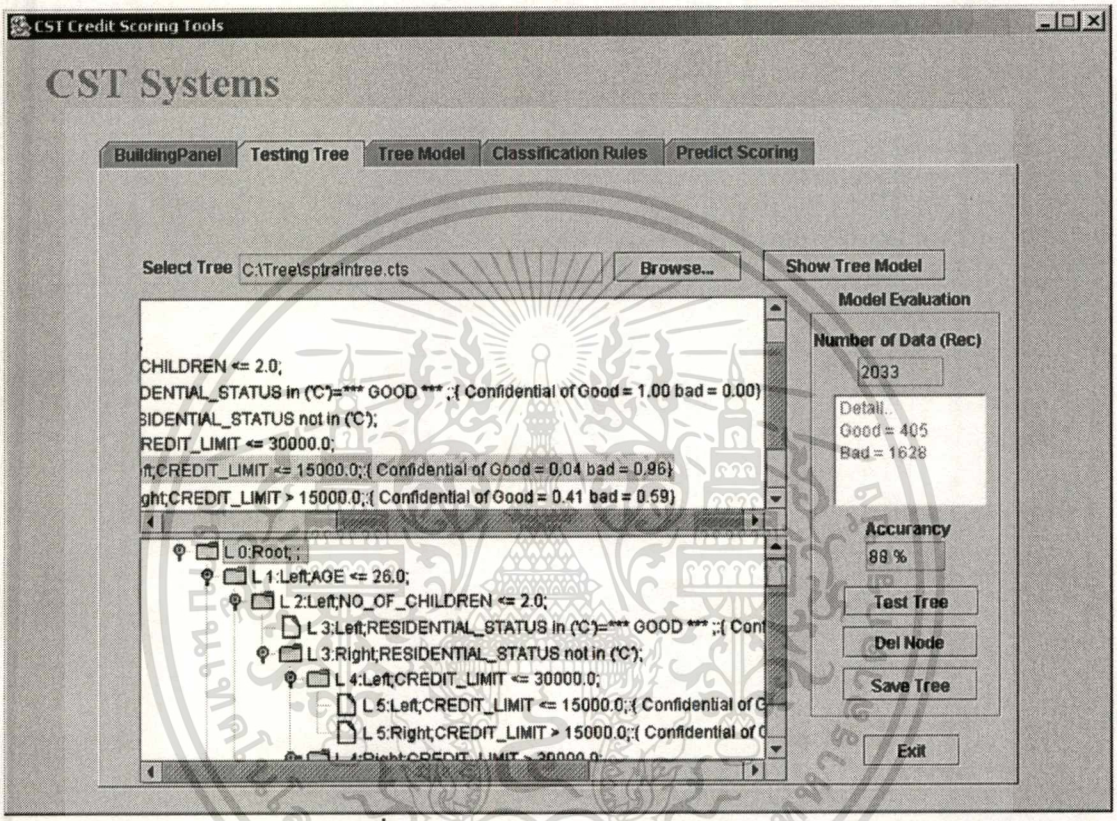


รูปที่ 18 Classification Rules จากการ Training Tree

การทดสอบโมเดลที่ได้ ว่ามีความแม่นยำกับการประยุกต์ใช้งานหรือไม่ โดยวิธีการทดสอบปฏิบัติดังนี้

- นำข้อมูลจำนวน 20% Records ซึ่งเป็นข้อมูลเก่าที่ทราบผลของการพิจารณา Credit Rating อยู่แล้วว่ามีค่าเท่าไหร่ มาผ่านกระบวนการ ในการ Apply Model เข้าไปเพื่อทำนายผลออกมา
- ตรวจสอบผลลัพธ์ที่ได้จากการทำนายโดยใช้โมเดล กับผลการให้ Credit Rating ของเดิม

– สรุปผล พบว่าสามารถทำนายได้ถูกต้อง 1,793 เรคคอร์ด จาก 2,033 เรคคอร์ด หรือคิดเป็น 88 เปอร์เซ็นต์ ซึ่งได้รับผลลัพธ์เป็นที่น่าพอใจ ในการนำโมเดลไปประยุกต์ใช้งาน



รูปที่ 19 Accuracy จากการ Testing Tree

5.2 ข้อเสนอแนะ

1. ระบบที่นำเสนอเป็นต้นแบบของระบบการสนับสนุนการตัดสินใจในการวิเคราะห์ข้อมูลสินเชื่อบัตรเครดิต ฉะนั้นข้อมูลที่เป็นปัจจัยที่มีผลและถูกนำมาพิจารณา อาจมีมากกว่านี้ ซึ่งสามารถนำไปประยุกต์ต่อไปในอนาคตได้ เพื่อให้โมเดลที่ได้มีความแม่นยำมากยิ่งขึ้น
2. นอกเหนือจากข้อมูลสินเชื่อบัตรเครดิตที่ทางศูนย์ข้อมูลเครดิตมีแล้ว ยังมีข้อมูลประเภทอื่น ๆ อีก ยกตัวอย่างเช่น ข้อมูลการกู้ซื้อบ้าน ข้อมูลการเช่าซื้อรถ และข้อมูลบัญชีกระแสรายวัน ซึ่งข้อมูลเหล่านี้ทางธนาคารสมาชิกเป็นผู้ส่งเข้ามาที่ศูนย์ข้อมูลเครดิต ซึ่งในการพัฒนาขั้นตอนถัดไปนั้น สามารถนำข้อมูลเหล่านี้เข้ามาเป็น Input Field ในการพัฒนา Pattern หรือ Rule ต่าง ๆ ที่มีประสิทธิภาพมากยิ่งขึ้น

บรรณานุกรม

Ian, H. Witten and Eibe Frank. 1999. **Data Mining, Practical Machine Learning Tools and Techniques with Java Implementations.**

Jayagopal, Brendan and Amadeus Software Limited. n.d. **Apply Data Mining Techniques to Credit Scoring.**

Jiawei, Han and Micheline Kamber. 2001. **Data Mining Concepts and Techniques.** n.p.

John Shafer. et. al. 2003. **SPRINT : A Scalable Parallel Classifier for Data Mining.** IBM Almeden Research Center

Matthias Schumann and Arbeitsbericht Nr. 16/2001. **Credit Scoring.**

ประวัติผู้เขียน

ชื่อผู้เขียน	นางสาวดวงวิชุดา ศรีศิริศุภโยค
สถานที่เกิด	จังหวัดกรุงเทพ
ระดับประถมศึกษา	โรงเรียนเซนต์โยเซฟคอนเวนต์
ระดับมัธยมศึกษาตอนต้น	โรงเรียนเซนต์โยเซฟคอนเวนต์
ระดับมัธยมศึกษาตอนปลาย	โรงเรียนเซนต์โยเซฟคอนเวนต์
ระดับอุดมศึกษา	คณะพาณิชยศาสตร์และการบัญชี จุฬาลงกรณ์มหาวิทยาลัย
วุฒิการศึกษาระดับปริญญาตรี	สถิติศาสตร์บัณฑิต (สตบ)
ประสบการณ์การทำงาน	บริษัท เทคโนโลยีแอนด์ซอฟต์แวร์ โซลูชั่น