การเปรียบเทียบประสิทธิภาพของอัลกอริธึม FEATURE SELECTION ในการแบ่งกลุ่มข้อมูล

COMPARISON OF FEATURE SELECTION ALGORITHMS' PERFORMANCE IN CLASSIFICATION

โดย

กฤตมุข ลีศิริชัยกุล

KRISTAMUK LEESIRICHAIKUL

อาจารย์ที่ปรึกษา

รศ.ดร. อาริต ธรรมโน

รายงานนี้เป็นส่วนหนึ่งของวิชาการศึกษาอิสระ
หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
คณะเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ภาคฤดูร้อน ปีการศึกษา 2552

COMPARISON OF FEATURE SELECTION ALGORITHMS' PERFORMANCE IN CLASSIFICATION

KRISTAMUK LEESIRICHAIKUL

A REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE

REQUIREMENTS OF THE COURSE

INDEPENDENT STUDY

MASTER OF SCIENCE PROGRAM IN INFORMATION TECHNOLOGY

FACULTY OF INFORMATION TECHNOLOGY

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

SUMMER / 2009

ใบรับรองการศึกษาอิสระ (INDEPENDENT STUDY)

เรื่อง

การเปรียบเทียบประสิทธิภาพของอัลกอริธึม FEATURE SELECTION ในการแบ่งกลุ่มข้อมูล

COMPARISON OF FEATURE SELECTION ALGORITHMS' PERFORMANCE IN CLASSIFICATION

นายกฤตมุข ลีศิริชัยกุล

รหัสประจำตัว 51066519

ขอรับรองว่ารายงานฉบับนี้ ข้าพเจ้าไม่ได้คัดลอกมาจากที่ใด
รายงานฉบับนี้ได้รับการตรวจสอบและอนุมัติให้เป็นส่วนหนึ่งของ
การศึกษาวิชาการศึกษาอิสระ หลักสูตรวิทยาศาสตรมหาบัณฑิต (เทคโนโลยีสารสนเทศ)
ภาคฤดูร้อน ปีการศึกษา 2552

.....อาจารย์ที่ปรึกษา
(รศ.ดร. อาริต ธรรมโน)

.....กรรมการสอบ
(รศ.ดร. วรพจน์ กรีสุระเดช)

.....กรรมการสอบ
(ผศ.ดร. ภัทรชัย ลลิตโรจน์วงศ์)

หัวข้อ	การเปรียบเทียบประสิทธิภาพของอัลกอริธึม Feature Selection ในการแบ่งกลุ่มข้อมูล
นักศึกษา	นายกฤตมุข ลีศิริชัยกุล
รหัสนักศึกษา	51066519
ปริญญา	วิทยาศาสตรมหาบัณฑิต
สาขาวิชา	เทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2552
อาจารย์ที่ปรึกษา	รศ.ดร.อาริต ธรรมโน

บทคัดย่อ

ในปัจจุบันการทำดาต้าไมน์นิ่งนั้นได้มีส่วนสำคัญในการดำเนินธุรกิจของบริษัทต่างๆเป็นอย่างมาก ซึ่งเทคนิคที่เป็นที่นิยมนำมาใช้งานคือการแบ่งกลุ่มข้อมูล โดยกระบวนการขั้นตอนสำคัญส่วนหนึ่งในการแบ่งกลุ่มข้อมูลนั้นคือ การเลือกแอทริบิวต์หรือลักษณะ (Feature) ของข้อมูล โดยมีอัลกอริธึมให้เลือกใช้เป็นจำนวนมาก ซึ่งอัลกอริธึมต่างๆเหล่านี้มีวิธีการและการทำงานที่แตกต่างกันไปและเหมาะสมกับประเภทข้อมูลที่แตกต่างกัน จึงเป็นที่มาของโครงการที่ต้องการทำการทดลองเปรียบเทียบประสิทธิภาพของอัลกอริธึมบางส่วนในการทำงานกับประเภทของข้อมูลที่มีลักษณะเหมือนกันแต่มีจำนวนและขนาดที่แตกต่างกัน เพื่อหาข้อดีและข้อเสียรวมทั้งประสิทธิภาพของแต่ละอัลกอริธึมเหล่านี้ และนำไปใช้ประยุกต์ใช้กับกระบวนการแบ่งกลุ่มข้อมูลต่างๆเพื่อให้ได้ผลลัพธ์ที่มีประสิทธิภาพและแม่นยำมากขึ้นต่อไป

Title	Comparison of Feature Selection algorithms' performance in Classification
Student	Mr. Kristamuk Leesirichaikul
Student ID.	51066519
Degree	Master of Science
Program	Information Technology
Major	Information Science
Academic Year	2009
Advisor	Assoc. Prof. Dr. Arit Thammano

ABSTRACT

Data mining has come to play a significant part in business operation of many organizations and companies. One of the most frequently used techniques of data mining is classification which has feature selection as its crucial part. Different algorithms may be used in the selection depending on the usage and date type. This project aims to compare the performance of some algorithms at work through data samples of the same type but differ in size to identify their strengths and weaknesses as well as efficiency. The results will be applied to use in the classification process to obtain results which are more efficient and accurate.

กิตติกรรมประกาศ

โครงการศึกษาการเปรียบเทียบประสิทธิภาพของอัลกอริธึม Feature Selection ในรายงานฉบับนี้ สามารถสำเร็จลุล่วงไปได้ด้วยดีเนื่องด้วยการให้คำแนะนำและคำปรึกษาจาก รศ.ดร. อาริต ธรรมโน ซึ่งเป็นอาจารย์ผู้ปรึกษาโครงการศึกษานี้ โดยอาจารย์ได้สละเวลาให้คำปรึกษาและแนะแนวข้าพเจ้าเป็นอย่างดี ข้าพเจ้ารู้สึกซาบซึ้งในความอนุเคราะห์ของอาจารย์และขอขอบพระคุณอาจารย์เป็นอย่างสูง

ขอกราบขอบพระคุณ รศ.ดร. วรพจน์ กรีสุระเดช ที่ได้ให้ความรู้ในเรื่องของการทำดาต้าไมน์นิ่งและเทคนิคต่างๆในการทำดาต้าไมน์นิ่ง รวมทั้งได้ให้คำแนะนำต่างๆตลอดระยะเวลาที่ได้ศึกษาอยู่ในมหาวิทยาลัยแห่งนี้

ขอกราบขอบพระคุณคณาจารย์ ภาควิชาวิทยาการสารสนเทศ คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยี พระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุกๆท่านที่ได้ประสิทธิ์ประสาทวิชาให้กับข้าพเจ้า

และสุดท้ายนี้ขอกราบขอบพระคุณบิดา มารดา และทุกคนในครอบครัวของข้าพเจ้า รวมทั้งเพื่อนๆที่ช่วยผลักดัน สนับสนุนในทุกๆ เรื่อง และคอยเป็นกำลังใจอยู่ตลอดเวลา ทำให้ข้าพเจ้าสามารถทำวิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี

คุณค่าและประโยชน์อันพึงมาจากวิทยานิพนธ์ฉบับนี้ ข้าพเจ้าขอมอบแด่ผู้มีพระคุณทุกท่าน หากผิดพลาดประการใด ขออภัยไว้ ณ ที่นี้ด้วย

กฤตมุข ลีศิริชัยกุล

สารบัญ

สารบัญ (ต่อ)

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

สารบัญตาราง (ต่อ)

ตารางที่ หน้า

สารบัญรูป

สารบัญรูป (ต่อ)

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันข้อมูลข่าวสารต่างๆในระบบสารสนเทศนั้นมีความสำคัญต่อการดำเนินงานและใช้ในการกำหนดทิศทางเป้าหมายขององค์กรต่างๆ ซึ่งการจะได้มาซึ่งข้อมูลอันเป็นประโยชน์จากข้อมูลที่มีอยู่นั้น จะต้องผ่านกระบวนการทำงานที่เรียกว่าการทำ "ดาต้าไมน์นิ่ง" ซึ่งกำลังได้รับความนิยมและถูกใช้งานโดยองค์กรต่างๆ โดยเทคนิคในการทำดาต้าไมน์นิ่งนั้นจะมีหลากหลายประเภทขึ้นอยู่กับการนำไปใช้งาน และเทคนิคหนึ่งที่นิยมนำไปใช้งาน คือ "การแบ่งกลุ่มข้อมูล (Classification)" นั่นเอง ซึ่งเทคนิคการแบ่งกลุ่มข้อมูลนี้ก็จะประกอบด้วยขั้นตอนต่างๆ โดยขั้นตอนหนึ่งที่มีความสำคัญต่อผลลัพธ์ในการแบ่งกลุ่มข้อมูลคือการทำ "Feature Selection" ซึ่งมีอัลกอริธึมที่ใช้ในการทำหลากหลาย โดยแต่ละอัลกอริธึมจะมีลักษณะการทำงานที่แตกต่างกันไปและให้ผลลัพธ์การทำงานที่แตกต่างกัน ทำให้บางครั้งผู้ใช้งานยากที่จะตัดสินใจในการเลือกใช้อัลกอริธึมในการ Feature Selection

จากปัญหาดังกล่าว ในรายงานฉบับนี้จึงได้ทำการศึกษาเกี่ยวกับอัลกอริธึมในการทำ Feature Selection ส่วนหนึ่ง ซึ่งประกอบด้วย Relief-F, Fast Correlation-Based Filter (FCBF) และ Las Vegas Filter (LVF) และมีความสนใจที่ต้องการนำผลลัพธ์การทำงานของอัลกอริธึมเหล่านี้มาเปรียบเทียบกันว่ามีผลลัพธ์อย่างไร เมื่อนำผลลัพธ์ที่ได้ไปผ่านเทคนิคการแบ่งกลุ่มข้อมูล โดยทางผู้จัดทำรายงานนี้ได้เลือกใช้โครงข่ายประสาทเทียม (Neural Networks) ในการแบ่งกลุ่มข้อมูลในการเปรียบเทียบประสิทธิภาพผลลัพธ์ที่ได้จากอัลกอริธึมเหล่านี้ จึงเป็นที่มาของโครงการนี้

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

เพื่อเปรียบเทียบประสิทธิภาพการทำงานของอัลกอริธึมในการทำ Feature Selection เพื่อใช้ในการแบ่งกลุ่มข้อมูล มีวัตถุประสงค์ดังต่อไปนี้

1. เพื่อศึกษาการทำงานเกี่ยวกับการแบ่งกลุ่มข้อมูลโดยใช้โครงข่ายประสาทเทียม
2. เพื่อศึกษาการทำงานของอัลกอริธึม Relief-F, Fast Correlation-Based Filter (FCBF) และ Las Vegas Filter (LVF) ในการทำ Feature Selection
3. เพื่อทำการเปรียบเทียบประสิทธิภาพของผลลัพธ์ที่ได้จากการแบ่งกลุ่มข้อมูล โดยการใช้ Relief-F, Fast Correlation-Based Filter (FCBF) และ Las Vegas Filter (LVF) ในการทำ Feature Selection

1.3 สมมติฐานของการศึกษา

อัลกอริธึม Relief-F, Fast Correlation-Based Filter (FCBF) และ Las Vegas Filter (LVF) ที่ใช้ในการทำ Feature Selection นั้น มีความเหมาะสมกับประเภทหรือชนิดของข้อมูลที่แตกต่างกันไป และขนาดของข้อมูลที่แตกต่างด้วย โดยคาดว่าผลลัพธ์ที่ได้จากการทำ Feature Selection จะมีความแตกต่างกันไป และเมื่อนำผลลัพธ์ที่ได้มาใช้ในการแบ่งกลุ่มข้อมูล ก็จะทำให้ผลลัพธ์การแบ่งกลุ่มข้อมูลที่ได้นั้น มีความแม่นยำที่แตกต่างกันไปด้วย

1.4 ขอบเขตของการศึกษา

การศึกษาโครงงานนี้มีขอบเขตในการศึกษาในการเปรียบเทียบประสิทธิภาพผลลัพธ์ที่ได้จากการแบ่งกลุ่มข้อมูล โดยการใช้อัลกอริธึม Relief-F, Fast Correlation-Based Filter (FCBF) และ Las Vegas Filter (LVF) ในการทำ Feature Selection โดยหัวข้อหลักในการศึกษาประกอบด้วย

1. ระบบการแบ่งกลุ่มข้อมูลโดยโครงข่ายประสาทเทียมและใช้ Relief-F ในการทำ Feature Selection
2. ระบบการแบ่งกลุ่มข้อมูลโดยโครงข่ายประสาทเทียมและใช้ Fast Correlation-Based Filter (FCBF) ในการทำ Feature Selection
3. ระบบการแบ่งกลุ่มข้อมูลโดยโครงข่ายประสาทเทียมและใช้ Las Vegas Filter (LVF) ในการทำ Feature Selection

1.5 ขั้นตอนของการศึกษา

ขั้นตอนในการศึกษาเปรียบเทียบประสิทธิภาพของการทำ Feature Selection โดยใช้อัลกอริธึม Relief-F, Fast Correlation-Based Filter (FCBF) และ Las Vegas Filter (LVF) นั้น มีขั้นตอนดังนี้

1. กำหนดวัตถุประสงค์ในการศึกษา
2. ศึกษาขั้นตอนและวิธีการทำดาต้าไมน์นิ่ง
3. ศึกษาเทคนิคการแบ่งกลุ่มข้อมูลโดยใช้โครงข่ายประสาทเทียม
4. ศึกษาการทำ Feature Selection โดยใช้อัลกอริธึม Relief-F, Fast Correlation-Based Filter (FCBF) และ Las Vegas Filter (LVF)
5. ออกแบบและพัฒนาระบบในการแบ่งกลุ่มข้อมูลและการทำ Feature Selection
6. ตรวจสอบและแก้ไขระบบให้มีความถูกต้องและสมบูรณ์

7. วิเคราะห์และตรวจสอบผลลัพธ์ที่ได้
8. สรุปผลการศึกษา

1.6 ประโยชน์ที่คาดว่าจะได้รับ

จากโครงงานเปรียบเทียบประสิทธิภาพของอัลกอริธึมในการทำ Feature Selection เพื่อแบ่งกลุ่มข้อมูลนั้น ผู้จัดทำคาดว่าจะได้รับประโยชน์ ดังนี้

1. เข้าใจหลักการและขั้นตอนการทำงานต่างๆในกระบวนการทำดาต้าไมน์นิ่ง
2. เข้าใจหลักการและขั้นตอนการทำงานในการแบ่งกลุ่มข้อมูลโดยใช้โครงข่ายประสาทเทียมแบบ Backpropagation
3. เข้าใจหลักการและขั้นตอนการทำงานในการทำ Feature Selection โดยใช้อัลกอริธึม Relief-F, Fast Correlation-Based Filter (FCBF) และ Las Vegas Filter (LVF)
4. สามารถบอกถึงความแตกต่าง และข้อดีข้อเสียของการใช้อัลกอริธึมทั้ง 3 ในการทำ Feature Selection เพื่อแบ่งกลุ่มข้อมูล
5. สามารถเลือกใช้อัลกอริธึมที่เหมาะสมกับข้อมูลในการทำ Feature Selection เพื่อใช้ในการแบ่งกลุ่มข้อมูลต่อไป

บทที่ 2
เทคนิคการทำดาต้าไมน์นิ่งด้วยวิธีแบ่งกลุ่มข้อมูลและอัลกอริธึม ในการทำ Feature Selection เพื่อแบ่งกลุ่มข้อมูล

2.1 ประวัติความเป็นมาของดาต้าไมน์นิ่ง

ดาต้าไมน์นิ่งนั้นเป็นคำที่ผู้คนได้ยินได้รู้จักเมื่อไม่กี่ปีที่ผ่านมา แท้จริงแล้วการทำดาต้าไมน์นิ่งนั้นมีประวัติการวิวัฒนาการที่ยาวนาน แต่คำว่าดาต้าไมนนิ่งเพิ่งจะถูกใช้เมื่อไม่นานมานี้ในช่วงปี 90

รากฐานของดาต้าไมน์นิ่งนั้นมาจาก 3 สายงานด้วยกัน โดยเกี่ยวข้องกับสายทางด้านสถิติ (Statistic) มาเป็นเวลายาวนานมากที่สุด จึงถือได้ว่ารากฐานของดาต้าไมน์นิ่งนั้นคือสถิตินั่นเอง ในแรกเริ่มด้าต้าไมน์นิ่งนั้นจะเกี่ยวข้องกับหลัก Concept เรื่อง regression analysis, standard distribution, standard deviation, standard variance, discriminant analysis, cluster analysis และ confidence intervals ซึ่งใช้สำหรับศึกษาเกี่ยวกับข้อมูลและความสัมพันธ์ของข้อมูล และถือว่าเป็นรากฐานในการวิเคราะห์ทางสถิติในขั้นที่สูงขึ้นไป และเป็นหัวใจสำคัญสำหรับเครื่องมือและเทคนิคในการทำดาต้าไมน์นิ่งในปัจจุบันนี้

สายงานที่มีความเกี่ยวข้องต่อมาคือ Artificial Intelligence หรือที่รู้จักกันในชื่อสั้นๆว่า AI ซึ่งมีส่วนช่วยในการประยุกต์วิธีการคิดของมนุษย์มาใช้กับปัญหาทางด้านสถิติ เนื่องจากวิธีการนี้ต้องใช้ความสามารถในการทำงานของคอมพิวเตอร์เป็นจำนวนมาก จึงเพิ่งเริ่มมีการนำมาใช้งานในยุคปี 80 ที่คอมพิวเตอร์มีความสามารถมากพอและมีราคาถูกลงเหมาะสมกับการนำมาใช้งาน

สายงานสุดท้ายที่มีความเกี่ยวข้องก็คือ Machine Learning โดยในขณะที่ AI นั้นไม่ประสบความสำเร็จทางด้านพาณิชย์เท่าที่ควร Machine Learning นั้นก็ได้เข้ามามีส่วนช่วยในการทำงานร่วมกับเทคนิคทางด้านสถิติเหล่านี้ เนื่องด้วยความสามารถต่อราคาของคอมพิวเตอร์ในยุคปีที่ 80-90 ที่เพิ่มขึ้นและราคาที่ถูกกว่าการใช้ AI จะเรียกได้ว่า Machine Learning นั้นเป็นวิวัฒนาการต่อมาจาก AI ก็ได้ โดย Machine Learning จะพยายามให้โปรแกรมคอมพิวเตอร์เรียนรู้เกี่ยวกับข้อมูลต่างๆที่สนใจ ซึ่งการได้มาซึ่งผลลัพธ์ในการตัดสินใจของ Machine Learning นั้นก็มาจากปริมาณของข้อมูลที่ได้เรียนรู้ พื้นฐานทางสถิติ เทคนิคการทำงานและอัลกอริธึมของ AI นั่นเอง

ดาต้าไมน์นิ่งจึงเรียกได้ว่ามีรากฐานมาจากผลรวม ของประวัติและการพัฒนาทางด้านสถิติ , AI และ Machine Learning ซึ่งนำเอาเทคนิคจากสิ่งเหล่านี้มาใช้ร่วมกันเพื่อทำดาต้าไมน์นิ่งนั่นเอง

2.2 กระบวนการทำดาต้าไมน์นิ่ง

2.2.1 ความหมายของการทำดาต้าไมน์นิ่ง

การทำดาต้าไมน์นิ่ง หรืออาจจะเรียกว่า การค้นหาความรู้ในฐานข้อมูล (Knowledge Discovery in Databases - KDD) เป็นเทคนิคเพื่อค้นหารูปแบบ (pattern) ของจากข้อมูลจำนวนมหาศาล (Large Information) โดยใช้ขั้นตอนวิธีจากวิชาสถิติ การเรียนรู้ของเครื่อง และ การรู้จำแบบ หรือในอีกนิยามหนึ่ง [9] การทำเหมืองข้อมูล คือ กระบวนการที่กระทำกับข้อมูล เพื่อค้นหารูปแบบ แนวทาง และความสัมพันธ์ที่ซ่อนอยู่ในชุดข้อมูลนั้น โดยเป็นสารสนเทศที่มีเหตุผล (Valid) และสามารถนำไปใช้ได้ (Actionable) [8] ซึ่งเป็นสิ่งสำคัญในการที่จะช่วยการตัดสินใจในการทำธุรกิจ โดยอาศัย ซอฟท์แวร์ทูล (Software tool), หลักสถิติ, การรู้จำ การเรียนรู้ของเครื่อง และหลักคณิตศาสตร์

2.2.2 เทคนิคการทำดาต้าไมน์นิ่ง

ดาต้าไมน์นิ่งมีเทคนิคต่างๆที่สามารถนำมาใช้งานได้หลายรูปแบบวิธีด้วยกัน ซึ่งในแต่ละเทคนิคมีความเหมาะสมกับการวิเคราะห์ข้อมูลแตกต่างกันไป แต่เทคนิคที่นิยมนำมาใช้งานส่วนใหญ่มีอยู่ 3 วิธีคือ

- การแบ่งประเภทของข้อมูล (Classification) และการทำนาย (Prediction) โดยการแบ่งประเภทของข้อมูล การจำแนกกลุ่มของข้อมูลโดยการสร้างแบบจำลอง (model) เพื่อจัดการข้อมูลให้อยู่ในกลุ่มที่กำหนดมาให้ ตัวอย่างเช่น หาความสัมพันธ์ระหว่างผลการตรวจร่างกายกับการเกิดโรค โดยใช้ข้อมูลผู้ป่วยและการวินิจฉัยของแพทย์ที่เก็บไว้ เพื่อนำมาช่วยวินิจฉัยโรคของผู้ป่วย ในทางธุรกิจจะใช้เพื่อดูคุณสมบัติของผู้ที่จะก่อหนี้ดีหรือหนี้เสีย เพื่อประกอบการพิจารณาการอนุมัติเงินกู้ ส่วนการทำนายล่วงหน้าเป็นงานที่มีลักษณะคล้ายกับการแบ่งประเภทของข้อมูล เพียงแต่จะใช้สถิติที่ได้บันทึกจากการแบ่งประเภทของข้อมูลมาใช้ในการทำนายอนาคต ตัวอย่างเช่น การทำนายการเปลี่ยนแปลงพฤติกรรมของตลาด, การทำนายจำนวนลูกค้าที่จะออกจากธุรกิจของเราใน 6 เดือนข้างหน้า เป็นต้น
- กฎเชื่อมโยง (Association rule Discovery) เป็นเทคนิคหนึ่งของดาต้าไมน์นิ่งที่สำคัญและสามารถนำไปประยุกต์ใช้ได้จริงกับงานต่างๆ หลักการทำงานของวิธีนี้ คือ การค้นหาความสัมพันธ์ของข้อมูลจากข้อมูลขนาดใหญ่ที่มีอยู่เพื่อนำไปใช้ในการวิเคราะห์ หรือทำนายปรากฏการณ์ต่าง ๆ หรือมาจากการวิเคราะห์การซื้อสินค้าของลูกค้าเรียกว่า "Market Basket Analysis" ซึ่งประเมินจากข้อมูลในตารางที่รวบรวมไว้ ผลการวิเคราะห์ที่ได้จะเป็นคำตอบของปัญหา ซึ่งการวิเคราะห์แบบนี้เป็นการใช้ "กฎความสัมพันธ์" (Association Rule) เพื่อหาความสัมพันธ์ของข้อมูล ตัวอย่างการนำ

เทคนิคนี้ไปประยุกต์ใช้กับงานจริง ได้แก่ ระบบแนะนำหนังสือให้กับลูกค้าแบบอัตโนมัติของ Amazon ข้อมูลการสั่งซื้อทั้งหมดของ Amazon ซึ่งมีขนาดใหญ่มากจะถูกนำมาประมวลผลเพื่อหาความสัมพันธ์ของข้อมูล คือ ลูกค้าที่ซื้อหนังสือเล่มหนึ่งๆ มักจะซื้อหนังสือเล่มใดพร้อมกันด้วยเสมอ ความสัมพันธ์ที่ได้จากกระบวนการนี้จะสามารถนำไปใช้คาดเดาได้ว่าควรแนะนำหนังสือเล่มใดเพิ่มเติมให้กับลูกค้าที่เพิ่งซื้อหนังสือจากร้าน

- การจัดกลุ่มข้อมูล (Clustering) เป็นเทคนิคการลดขนาดของข้อมูลโดยการแบ่งกลุ่มข้อมูลใหม่ที่มี ด้วยการรวมกลุ่มตัวแปรที่มีคุณลักษณะเดียวกันไว้ด้วยกัน โดยไม่มีการจัดกลุ่มข้อมูลตัวอย่างไว้ล่วงหน้า เพื่อนำข้อมูลที่ได้ไปวิเคราะห์ เช่นการแบ่งกลุ่มผู้ป่วยที่เป็นโรคเดียวกันตามลักษณะ อาการ เพื่อนำไปใช้ประโยชน์ในการวิเคราะห์หาสาเหตุของโรค โดยพิจารณาจากผู้ป่วยที่มีอาการคล้ายคลึงกัน เทคนิคที่นิยมใช้ในการจัดกลุ่มข้อมูล ได้แก่ k-means clustering และ expectation maximization (EM) clustering

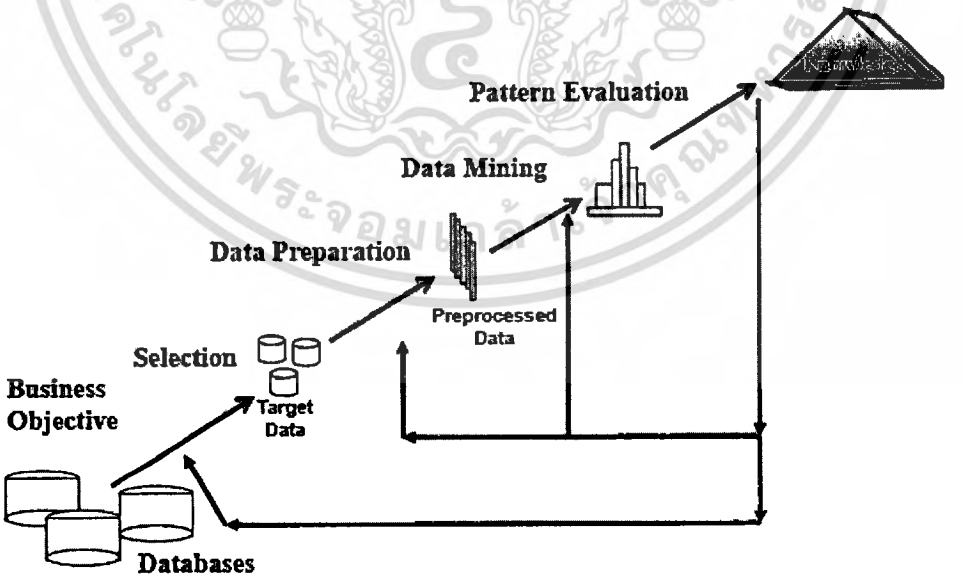
2.2.3 กระบวนการทำดาต้าไมน์นิ่ง

กระบวนการการทำดาต้าไมน์นิ่งนั้นจะประกอบไปด้วยขั้นตอนหลักๆ 5 ขั้นตอนด้วยกัน คือ

1. การวิเคราะห์ความต้องการของธุรกิจ การวิเคราะห์ความต้องการธุรกิจจะช่วยให้เข้าใจถึงประเด็นที่ผู้บริหารต้องตัดสินใจ ซึ่งเกี่ยวกับการสร้างความสำเร็จของธุรกิจ การวิเคราะห์ความต้องการของธุรกิจจะทำให้เกิดความเข้าใจถึงสถานะในปัจจุบันของธุรกิจ และทำให้ผู้บริหารสามารถกำหนดเรื่องที่ต้องตัดสินใจได้ดียิ่งขึ้น
2. การวิเคราะห์ความต้องการข้อมูล เนื่องจากคลังข้อมูลขนาดใหญ่ที่มีอยู่นั้น มีข้อมูลที่หลากหลาย ซึ่งมีทั้งข้อมูลที่จำเป็นต้องใช้ในการทำเหมืองข้อมูลกับข้อมูลอื่นซึ่งไม่เป็นที่ต้องการในขณะนี้ จึงต้องมีขั้นตอนการกำหนดรายการและประเภทของข้อมูลที่จะนำมาใช้ โดยมีการตรวจสอบในด้านของคุณภาพของข้อมูล จำนวน ปริมาณเนื้อหาและการเข้าถึงข้อมูล เพื่อกำหนดเป็นข้อมูลที่ต้องการทำเหมืองในกรณีที่มีข้อมูลจำนวนมาก อาจใช้การเลือกตัวอย่างข้อมูลมาทำเหมืองก่อนได้เพื่อลดค่าใช้จ่าย
3. การจัดเตรียมข้อมูล โดยจะทำการเลือกเอาข้อมูลที่ต้องการใช้ (Data Selection) เมื่อกำหนดข้อมูลที่จะใช้ในการทำเหมืองได้แล้ว ต้องนำข้อมูลนั้นมาทำการจัดเตรียมก่อนทำดาต้าไมน์นิ่งอีกครั้งโดยอาจจะต้องทำ Data Cleaning เพื่อจัดการกับค่าข้อมูลที่หายไป (Missing value), Data Integration เพื่อรวบรวมข้อมูลจากหลายๆที่เข้าด้วยกัน หรือการทำ Data Reduction เพื่อลดจำนวนข้อมูลที่ไม่มีความจำเป็นลง นอกจากนี้อาจต้องทำ Data Transformation เพื่อแปลงหรือปรับเปลี่ยนข้อมูลเพิ่มเติม

ให้อยู่ในรูปแบบที่เหมาะสมกับวิธีการที่จะใช้ในการทำเหมืองข้อมูลซึ่งโดยปกติแล้ว การแปลงข้อมูลจะถูกกำหนดโดยเงื่อนไขของการปฏิบัติงานและวิธีการทำเหมืองข้อมูล

4. การทำดาต้าไมน์นิ่ง เป็นขั้นตอนที่นำเอาวิธีการหรือเทคนิคการทำดาต้าไมน์นิ่งตั้งแต่หนึ่งวิธีขึ้นไป มาทำการสกัดสาระสำคัญออกจากฐานข้อมูลที่มี เช่นการตอบคำถามว่าลูกค้าจะซื้อสินค้าต่อไปหรือไม่ อาจต้องทำการวิเคราะห์ตั้งแต่การจัดกลุ่มของลูกค้าและการจำแนกหน่วยหรือลูกค้าแต่ละคนว่าจะซื้อหรือไม่ซื้อสินค้าต่อไป ทั้งนี้ ในขณะที่ทำเหมืองข้อมูล อาจมีความจำเป็นต้องเข้าถึงข้อมูลอื่นในคลังข้อมูล รวมทั้งต้องแปลงข้อมูลรายการอื่นด้วยก็ได้
5. การแปลผล หรือการประยุกต์ใช้กับธุรกิจ เป็นขั้นตอนที่นำเอาสารสนเทศที่ทำเหมืองได้มาวิเคราะห์เพื่อตอบคำถามที่ผู้ตัดสินใจมีอยู่ ซึ่งการวิเคราะห์ในส่วนนี้จะครอบคลุมการกรองสารสนเทศที่เหมาะสมกับการส่งให้ผู้ใช้ และการแปลผล เช่น ถ้าวัตถุประสงค์ของการทำดาต้าไมน์นิ่งคือการสร้างตัวแบบการจำแนกหน่วย ในขั้นตอนการแปลผล ก็จะต้องพิจารณาความเชื่อถือได้ของตัวแบบที่ได้ด้วยวิธีเช่น cross-validation เป็นต้น ถ้าผลที่ได้ไม่เป็นที่พอใจ ก็จะต้องทำขั้นตอนนี้ซ้ำอีก รวมทั้งขั้นตอนก่อนหน้าด้วย การแปลผลนี้อาจมองได้เป็นการประยุกต์วิทยาศาสตร์และเทคโนโลยีที่มีในการทำดาต้าไมน์นิ่ง ให้เป็นผลทางธุรกิจ ทำให้สามารถทำการประเมินผลที่ได้จากการทำดาต้าไมน์นิ่ง



รูปที่ 2.1 กระบวนการทำดาต้าไมน์นิ่ง

2.3 การแบ่งกลุ่มข้อมูล (Classification)

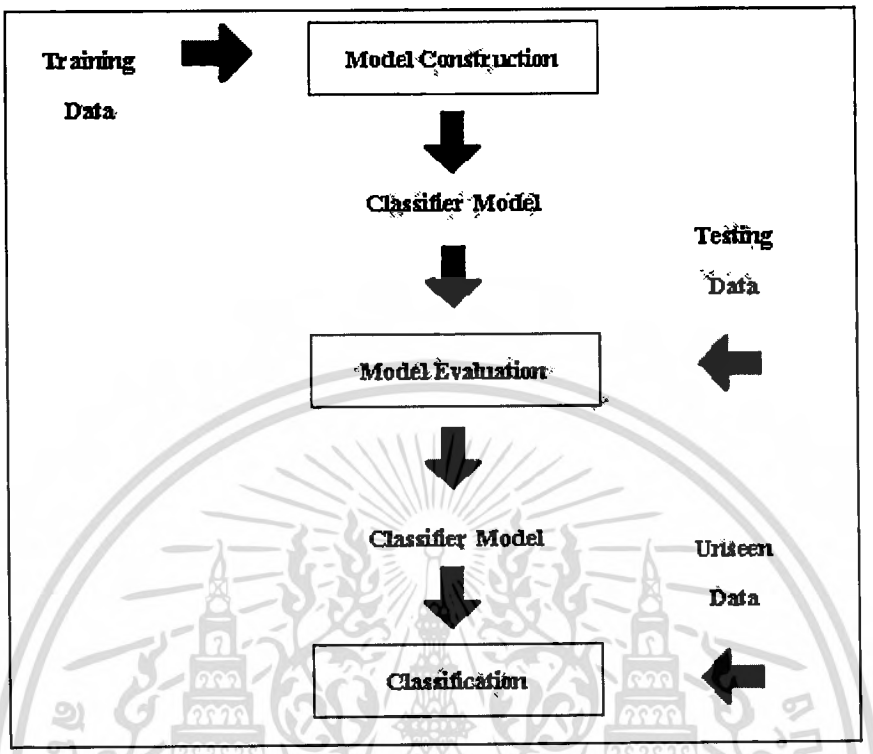
เป็นกระบวนการสร้างแบบจำลองจัดการข้อมูลให้อยู่ในกลุ่มที่กำหนดมาให้ เพื่อแสดงให้เห็นความแตกต่างระหว่างคลาส (Class) หรือกลุ่มของข้อมูล และเพื่อใช้ทำนายว่าข้อมูลนี้ควรจัดอยู่ในคลาสหรือกลุ่มข้อมูลใด โดยการสร้างแบบจำลองจะประกอบด้วยกระบวนการทำงาน 2 ขั้นตอนด้วยกัน คือ

1. การเรียนรู้ (Learning, Training) ซึ่งเป็นการเรียนรู้ของแบบจำลองที่จะใช้ในการจำแนกข้อมูลออกเป็นกลุ่มตามที่กำหนดไว้ (Predefined class) ซึ่งจะขึ้นอยู่กับการวิเคราะห์เซตของข้อมูลทดลอง (Training data) โดยอาศัยอัลกอริธึม Classification ที่จะนำเซตของข้อมูลทดลองนั้นมาสอนให้ระบบเรียนรู้ว่ามีข้อมูลใดอยู่ในคลาสหรือกลุ่มของข้อมูลเดียวกันบ้าง โดยการเรียนรู้นั้นสามารถแบบออกได้เป็น 2 ประเภทด้วยกัน คือ

   - Supervised Learning หรือการเรียนแบบมีการสอน คือ การเรียนรู้ที่มีเซตของข้อมูลทดลองประกอบด้วยข้อมูลเข้าและผลลัพธ์ที่ต้องการ โดยผลลัพธ์อาจจะเป็นค่าที่ต่อเนื่อง หรือกลุ่มของข้อมูลเข้าก็ได้
   - Unsupervised Learning หรือการเรียนแบบไม่มีการสอน คือ การเรียนรู้ที่ไม่มีการบอกเจาะจงถึงผลลัพธ์ที่ต้องการ ซึ่งระบบจะต้องทำการค้นหาและตัดสินใจการจัดการผลลัพธ์ที่จะได้ด้วยตนเอง

   ซึ่งการเรียนรู้ในการแบ่งประเภทข้อมูลนี้จะเรียกว่า "Supervised Learning" ผลลัพธ์ที่ได้จากการเรียนรู้ คือ แบบจำลองจัดประเภทข้อมูล (Classifier model) สำหรับอัลกอริธึมที่จะเลือกใช้ในการแบ่งประเภทข้อมูลนั้น มีปัจจัยในการเลือกใช้งานหลายๆ ปัจจัยด้วยกัน ไม่ว่าจะเป็น ความถูกต้อง (Accuracy) ของผลลัพธ์ที่ได้, ความรวดเร็ว (Speed) ในการสร้างแบบจำลองและการนำแบบจำลองไปใช้งาน, ความเสถียรคงทน (Robustness) ของอัลกอริธึมในการจัดการข้อมูลอาจจะมีลักษณะแตกต่างจากปกติ เช่น มีบางค่าที่หายไป เป็นต้น
2. การประเมินและตรวจสอบความถูกต้อง (Estimate accuracy) จะเป็นการประเมินความถูกต้องของแบบจำลองที่ได้ก่อนจะนำไปใช้งานจริงต่อไป โดยจะนำข้อมูลที่เป็น Predefined class มาเป็นข้อมูลที่ใช้ทดลอง (Testing data) โดยข้อมูลที่ใช้ทดลองนี้ควรจะเป็นอิสระจากเซตของข้อมูลทดลอง มิฉะนั้นจะทำให้เกิดปัญหา Over-Fitting ตามมา ซึ่งข้อมูลที่ใช้ทดลองนี้จะถูกนำมาตรวจสอบ เปรียบเทียบเป็นร้อยละกับกลุ่มของข้อมูลที่หามาได้จากแบบจำลองจัดประเภทข้อมูล เพื่อตรวจสอบความถูกต้อง โดยจะทำการปรับปรุงแบบจำลองจนกว่าจะได้ค่าความถูกต้องในระดับที่น่า

พอใจ หลังจากนั้นจึงจะนำเอาแบบจำลองไปใช้ในการแบ่งประเภทของกลุ่มข้อมูลที่จะใช้งานจริงๆ ต่อไป



รูปที่ 2.2 ขั้นตอนการสร้างแบบจำลองแบ่งประเภทข้อมูล

2.4 การแบ่งกลุ่มข้อมูลด้วยวิธีต่างๆ

การแบ่งกลุ่มข้อมูลในดาตาไมน์นิ่งนั้นสามารถทำได้วิธีการต่างๆหลายวิธี เช่น Decision Tree, โครงข่ายประสาทเทียม, Nearest-Neighbor Classifiers เป็นต้น โดยจะขอกล่าวถึงวิธีที่เป็นที่นิยมใช้งานเป็นส่วนใหญ่ 2 วิธีด้วยกัน คือ

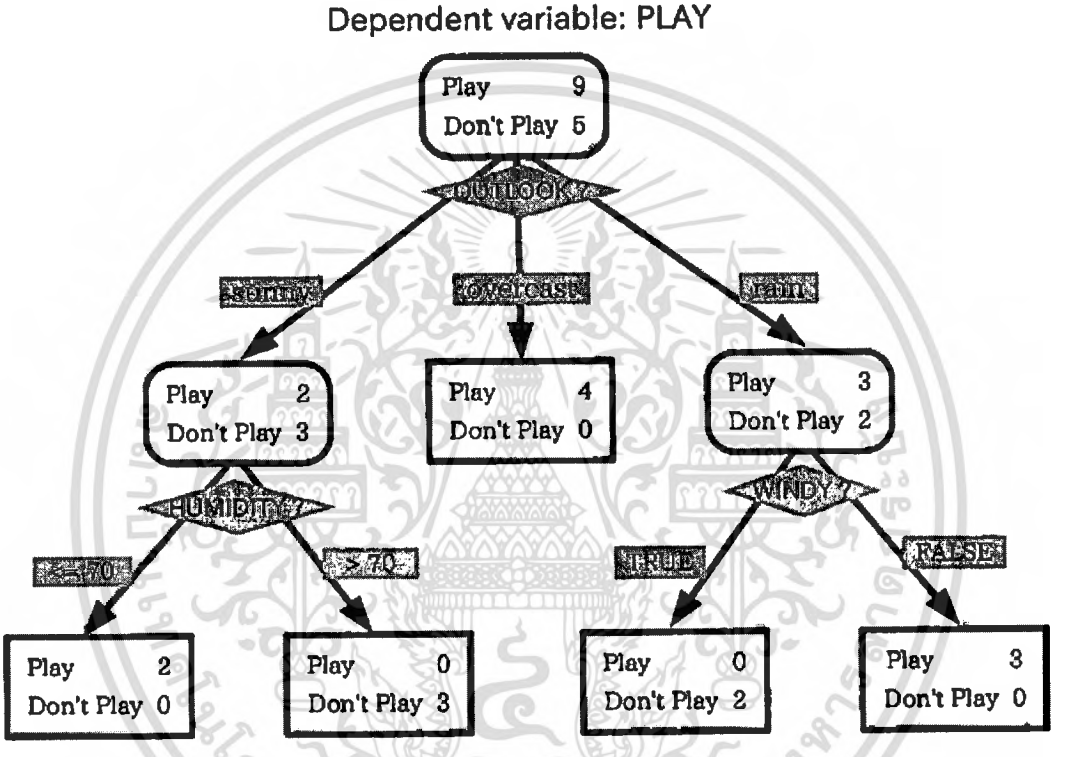
2.4.1 Decision Tree

Decision Tree เป็นหนึ่งวิธีที่สำคัญและนิยมในการทำดาต้าไมน์นิ่งด้วยเทคนิคการแบ่งกลุ่มข้อมูล โดย Decision Tree จะเป็นแบบจำลองทางคณิตศาสตร์ที่มีลักษณะคล้ายโครงสร้างต้นไม้ โครงสร้างของ Tree จะประกอบด้วยโหนด (Node) หลายๆโหนดต่อกันจนเป็น Tree โหนดด้านบนสุดที่เรียกว่า Root node จาก Root node จะแตกกิ่งออกเป็นโหนดลูก (Child node) และที่โหนดลูกแต่ละโหนดก็อาจจะจะมีโหนดลูกของตัวเองอีกที ซึ่งโหนดที่ระดับล่างสุดหรือโหนดปลายทางจะเรียกว่า "ลีฟโหนด (Leaf node)"

ในการสร้าง Decision Tree นั้นจะนิยมใช้วิธีการพื้นฐานคือการใช้ Greedy algorithm ที่ลักษณะเรียกว่า "Divide and Conquer" และ "Top Down Recursive " โดยวิธีการนั้นเริ่มแรกจะข้อมูลทุกตัวจะอยู่ที่ตำแหน่งรูทโหนด (Root node) หลังจากนั้นข้อมูลต่างๆ จะถูกแบ่งประเภท

ตามค่าแอททริบิวต์ (Attribute) ของข้อมูลที่เป็นเงื่อนไขในการแบ่งประเภทข้อมูล ซึ่งค่าแอททริบิวต์ที่ใช้ตรวจสอบจะแสดงอยู่ที่โหนดของ Tree ที่ไม่ใช่ลีฟโหนด (Non-leaf node) และเงื่อนไขที่ใช้ตรวจสอบนั้นจะแสดงอยู่ที่กิ่งของโหนด (Branch) โดย Decision Tree อาจจะมีกิ่งๆ เดียวหรือมากกว่านั้นก็ได้ ทั้งนี้ขึ้นอยู่กับประเภทของค่าแอททริบิวต์ของที่ใช้ เช่น Nominal, Numeric, Integer เป็นต้น

โดยปกติการทดลองข้อมูลที่โหนดจะเป็นการเปรียบเทียบค่าระหว่างค่าแอททริบิวต์กับค่าคงที่ แต่จะมีบาง Decision Tree ที่จะเปรียบเทียบค่ากันเองระหว่างค่าแอททริบิวต์



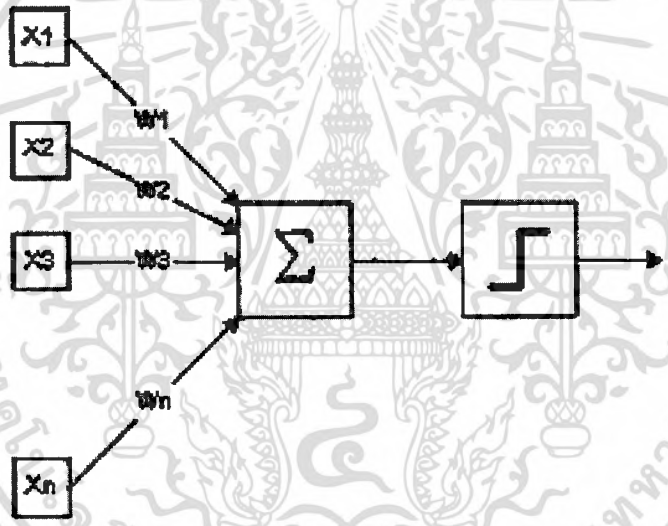
รูปที่ 2.3 ตัวอย่างของ Decision Tree

สำหรับข้อเสียของวิธีการแบ่งกลุ่มข้อมูลด้วย Decision Tree นั่นก็คือ การแบ่งกลุ่มแบบ Decision Tree จะไม่รองรับข้อมูลที่มีค่าต่อเนื่อง (Continuous data) เช่น ข้อมูลรายได้ ข้อมูลราคา เป็นต้น จะต้องมีการแบ่งให้เป็นข้อมูลแบบไม่ต่อเนื่อง (Discrete data) เสียก่อน นอกจากนี้ยังอาจจะมีปัญหา เรื่อง Overfitting / Overtraining หรือการที่ Tree มีกิ่งสาขาแตกออกจากโหนดจำนวนมากเกินไป ซึ่งมีสาเหตุมาจากการที่โมเดลเข้าถึงรายละเอียดของข้อมูลมากเกินไป ทำให้เกิดโหนดที่เป็นส่วนเฉพาะเจาะจงกับกลุ่มข้อมูล ซึ่งจะต้องหาวิธีในการตัดกิ่งนี้ออกไป โดยสามารถใช้วิธี Tree Pruning เพื่อนำมาใช้แก้ปัญหานี้

2.4.2 Neural Networks

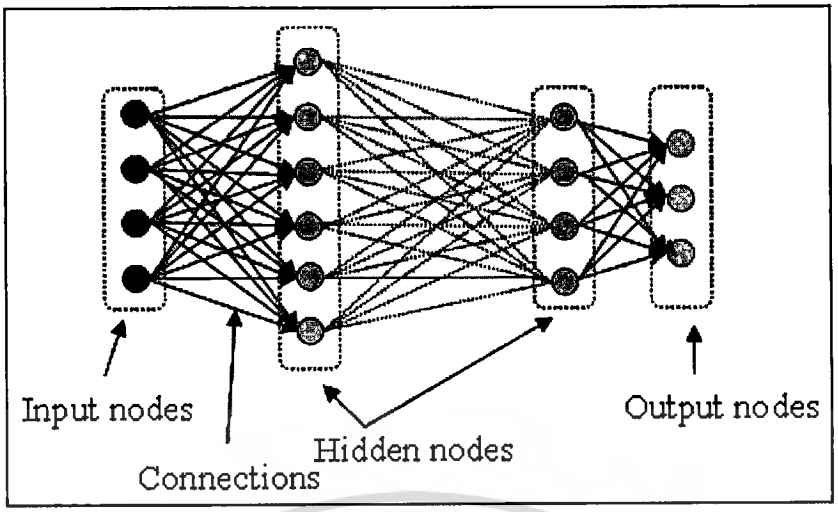
โครงข่ายประสาทเทียม (Artificial Neural Networks) คือ โมเดลทางคณิตศาสตร์ สำหรับประมวลผลสารสนเทศด้วยการคำนวณแบบคอนเนคชันนิสต์ (Connectionist) เพื่อจำลองการ

ทำงานของโครงข่ายประสาทในสมองมนุษย์ ด้วยวัตถุประสงค์ที่จะสร้างเครื่องมือซึ่งมีความสามารถในการเรียนรู้การจดจำรูปแบบ (Pattern recognition) และการอุปมานความรู้ (Knowledge deduction) เช่นเดียวกับความสามารถที่มีในสมองมนุษย์ [1] แนวคิดเริ่มต้นของเทคนิคนี้ได้มาจากการศึกษาข่ายงานไฟฟ้าชีวภาพ (Bioelectric network) ในสมอง ซึ่งประกอบด้วย เซลล์ประสาท หรือ "นิวรอน" (Neurons) และ จุดประสานประสาท (Synapses) แต่ละเซลล์ประสาทประกอบด้วยปลายในการรับกระแสประสาท เรียกว่า "เดนไดรท์" (Dendrite) ซึ่งเป็น input และปลายในการส่งกระแสประสาทเรียกว่า "แอคซอน" (Axon) ซึ่งเป็นเหมือน output ของเซลล์ เซลล์เหล่านี้ทำงานด้วยปฏิกิริยาไฟฟ้าเคมี เมื่อมีการกระตุ้นด้วยสิ่งเร้าภายนอกหรือกระตุ้นด้วยเซลล์ด้วยกัน กระแสประสาทจะวิ่งผ่านเดนไดรท์เข้าสู่นิวเคลียสซึ่งจะเป็นตัวตัดสินว่าต้องกระตุ้นเซลล์อื่นๆ ต่อหรือไม่ ถ้ากระแสประสาทแรงพอ นิวเคลียสก็จะกระตุ้นเซลล์อื่นๆ ต่อไปผ่านทางแอคซอนของมัน



รูปที่ 2.4 แบบจำลองโครงข่ายประสาทเทียมทางคอมพิวเตอร์

โครงสร้างของโครงข่ายประสาทเทียมจะประกอบด้วยโหนดสำหรับ Input – Output และการประมวลผล กระจายอยู่ในโครงสร้างเป็นชั้นๆ ได้แก่ Input layer, Output layer และ Hidden layers การประมวลผลของโครงข่ายประสาทเทียมจะอาศัยการส่งการทำงานผ่านโหนดต่างๆ ใน Layer เหล่านี้



รูปที่ 2.5 โครงสร้างแบบจำลองโครงข่ายประสาทเทียม

การทำงานของโครงข่ายประสาทเทียมคือเมื่อมี Input เข้ามายังโครงข่าย ก็นำเอา Input มาคูณกับ Weight ของแต่ละขา ผลที่ได้จาก Input ทุกๆ ขาของ Neuron จะเอามารวมกันแล้วก็เอามาเทียบกับ Threshold ที่กำหนดไว้ ถ้าผลรวมมีค่ามากกว่า Threshold แล้ว Neuron จะส่ง Output ออกไป Output นี้ก็จะถูกส่งไปยัง Input ของ Neuron อื่นๆ ที่เชื่อมกันใน Network ถ้าค่าน้อยกว่า Threshold ก็จะไม่เกิด Output สิ่งสำคัญคือเราต้องทราบค่าถ่วงน้ำหนัก (Weight) และ Threshold สำหรับสิ่งที่เราต้องการเพื่อให้คอมพิวเตอร์รู้จำ ซึ่งเป็นค่าที่ไม่แน่นอน แต่สามารถกำหนดให้คอมพิวเตอร์ปรับค่าเหล่านั้นได้โดยการสอนให้มันรู้จัก Pattern ของสิ่งที่เราต้องการให้มันรู้จำเรียกว่า "Back Propagation" ซึ่งเป็นกระบวนการย้อนกลับของการรู้จำ ในการฝึก Feed-Forward โครงข่ายประสาทเทียมจะมีการใช้อัลกอริทึมแบบ Backpropagation เพื่อใช้ในการปรับปรุงน้ำหนักคะแนนของโครงข่าย (Network weight) หลังจากใส่รูปแบบข้อมูลสำหรับฝึกให้แก่โครงข่ายในแต่ละครั้งแล้ว ค่าที่ได้รับ (Output) จากโครงข่ายจะถูกนำไปเปรียบเทียบกับผลที่คาดหวัง แล้วทำการคำนวณหาค่าความผิดพลาด ซึ่งค่าความผิดพลาดนี้จะถูกส่งกลับเข้าสู่โครงข่ายเพื่อใช้แก้ไขค่าน้ำหนักคะแนนต่อไป

2.5 อัลกอริธึมในโครงข่ายประสาทเทียม

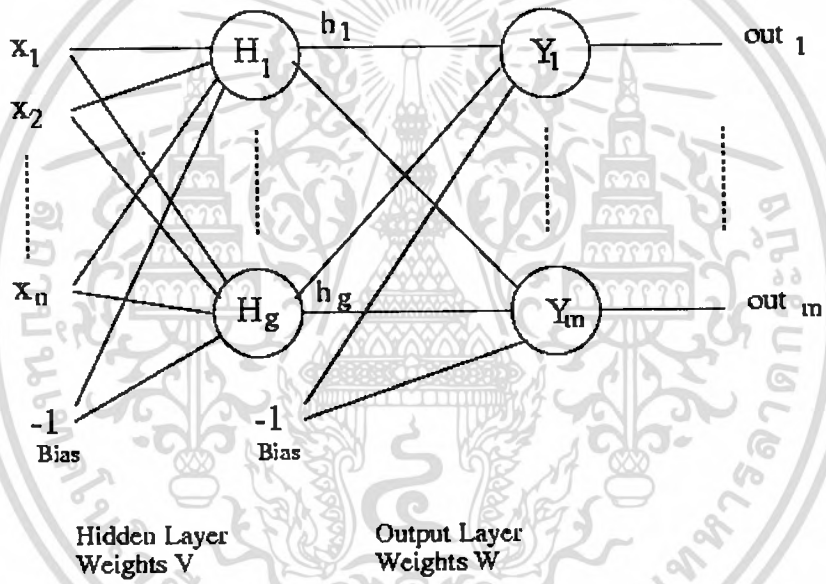
การแบ่งกลุ่มข้อมูลด้วยแบบจำลองโครงข่ายประสาทเทียมนั้น มีอัลกอริธึมที่ใช้ในการสร้างแบบจำลองจำนวนมากเช่นเดียวกันกับการแบ่งกลุ่มข้อมูลด้วย Decision Tree ตัวอย่างเช่น Feedforward network, Feedback network เป็นต้น โดยจะขอแสดงอัลกอริธึมตัวหนึ่งที่มีความสำคัญในโครงข่ายประสาทเทียมคือ Backpropagation Network

2.5.1 Backpropagation Network

Backpropagation Algorithm เป็นอัลกอริทึมที่ใช้ในการเรียนรู้ของโครงข่ายประสาทเทียม วิธีหนึ่งที่นิยมใช้ใน Multilayer perceptron เพื่อปรับค่าน้ำหนักในเส้นเชื่อมต่อระหว่างโหนดให้เหมาะสม โดยการปรับค่านี้จะขึ้นกับความแตกต่างของค่า output ที่คำนวณได้กับค่า output ที่ต้องการ

ขั้นตอนของ Backpropagation Algorithm มีดังนี้

1. Feed-forward computation
2. Backpropagation ในชั้น output layer
3. Backpropagation ในชั้น hidden layer
4. การอัพเดตค่า Weight



รูปที่ 2.6 แบบจำลอง Backpropagation Network

โดยในการคำนวณ Feed-forward computation นั้นจะทำเหมือนแบบจำลองโครงข่ายประสาทเทียมทั่วไป คือ นำผลรวมของผลคูณ Input ที่รับเข้ามาในแต่ละโหนดกับค่า Weight ของเส้นเชื่อมนั้น แล้วจึงนำมาคำนวณหาผลลัพธ์ที่ได้ผ่าน Transfer function โดยมีสูตรดังนี้

$$net = \sum_{i=1}^{n} w_i a_i , \quad f(net) = \frac{2}{1+e^{-\lambda * net}} - 1 \qquad (2.1)$$

โดย w_i แทนค่า Weight ของเส้นเชื่อมที่มากจากโหนดลำดับที่ i ของชั้นก่อนหน้านี้

a_i แทนค่า Input ที่มาจากโหนดลำดับที่ i ของชั้นก่อนหน้านี้

λ แทนค่า Scale factor

สำหรับการทำ Backpropagation เริ่มต้นจะคำนวณค่า Error signal (δ) ในแต่ละโหนดในชั้น Output layer และ Hidden layer ของแบบจำลอง โดยคำนวณได้จากสูตร

- ในชั้น Output layer

$$\delta_k = \tfrac{1}{2}\left(d_k - o_k\right)\left(1 - o_k^{\,2}\right) \quad \text{เมื่อ } k = 1, 2, 3, \ldots, K \qquad (2.2)$$

โดย d_k แทนค่า Output ที่ต้องการของโหนด Output ลำดับที่ k

o_k แทนค่า Output ที่คำนวณได้ของโหนด Output ลำดับที่ k

- ในชั้น Hidden layer

$$\delta_{yj} = \tfrac{1}{2}\left(1 - y_j^{\,2}\right)\sum_{k=1}^{K}\delta_k w_{kj} \quad \text{เมื่อ } j = 1, 2, 3, \ldots, J \qquad (2.3)$$

โดย y_j แทนค่า Output ที่ได้จากโหนดลำดับที่ j ในชั้น Hidden layer ที่ y

w_{kj} แทนค่า Weight ของเส้นเชื่อมจากโหนดลำดับที่ j ในชั้น Hidden layer ที่ y ไปยังโหนดลำดับที่ k ในชั้นถัดไป

เมื่อคำนวณหาค่า Error signal ในโหนดต่างๆ เสร็จ ลำดับถัดมาจะทำการอัพเดตค่า Weight ในแต่ละเส้นเชื่อมของแบบจำลองใหม่ โดยสามารถคำนวณได้จากสูตร

- ในชั้น Output layer

$$w_{kj} \leftarrow w_{kj} + \eta\delta_k y_j \text{ เมื่อ } k = 1, 2, 3, \ldots, K \text{ และ } j = 1, 2, 3, \ldots, J \qquad (2.4)$$

โดย η แทนค่า Learning rate มีค่าระหว่าง 0 ถึง 1

- ในชั้น Hidden layer

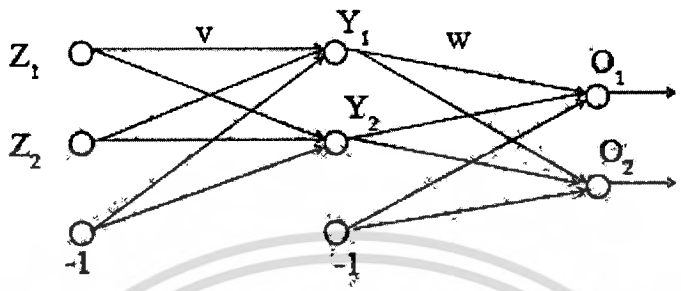
$$v_{ji} \leftarrow v_{ji} + \eta\delta_{yj} z_i \text{ เมื่อ } k = 1, 2, 3, \ldots, K \text{ และ } j = 1, 2, 3, \ldots, J \qquad (2.5)$$

โดย v_{ji} แทนค่า Weight ของเส้นเชื่อมจากโหนดลำดับที่ i ในชั้นก่อนหน้านี้มายังโหนดลำดับที่ j ในชั้น Hidden layer นี้

z_i แทนค่า Output ที่คำนวณได้ของโหนดลำดับที่ i ในชั้นก่อนหน้านี้

หลังจากทำการอัพเดตค่า Weight เสร็จก็จะทำคำนวณค่า Output ที่โหนด Output ใหม่ ว่าได้ตามที่ต้องการ หรือยอมรับได้หรือไม่ ถ้ายังก็จะทำการหาค่า Error signal ใหม่และปรับค่า Weight ใหม่ จนกระทั่งได้ผลลัพธ์ที่ต้องการหรือยอมรับได้

2.5.2 ตัวอย่างการทำ Backpropagation Network



รูปที่ 2.7 ตัวอย่างแบบจำลองโครงข่ายประสาทเทียมในการทำ Backpropagation Network

$$(\vec{Z},\vec{D}) = (\begin{bmatrix}0.5\\1\end{bmatrix},\begin{bmatrix}0\\1\end{bmatrix}),\ V_1 = \begin{bmatrix}1\\0\\0\end{bmatrix},\ V_2 = \begin{bmatrix}0.5\\1\\0.5\end{bmatrix},$$

$$W_1 = \begin{bmatrix}1\\1\\1\end{bmatrix},\ W_2 = \begin{bmatrix}0.5\\0.5\\0.5\end{bmatrix},\quad \eta = 0.1$$

จากตัวอย่าง คำนวณหาค่าผลลัพธ์จากโหนดต่างๆในชั้น Hidden layer และ Output layer

- ในชั้น Hidden layer

net $Y_1 = v_{11}Z_1 + v_{12}Z_2 + v_{13}Z_3$

$= 1(0.5) + 0(1) + 0(-1) = 0.5$

$Y_1 = f(0.5) = 0.245$

net $Y_2 = v_{21}Z_1 + v_{22}Z_2 + v_{23}Z_3$

$= 0.5(0.5) + 1(1) + 0.5(-1) = 1.25$

$Y_2 = f(1.25) = 0.555$

- ในชั้น Output layer

net $O_1 = w_{11}Y_1 + w_{12}Y_2 + w_{13}Y_3$

$= 1(0.245) + 1(0.555) + 1(-1) = -0.397$

$O_1 = f(-0.397) = -0.196$

net $O_2 = w_{21}Y_1 + w_{22}Y_2 + w_{23}Y_3$

$= 0.5(0.245) + 0.5(0.555) + 0.5(-1) = -0.1$

$O_2 = f(-0.1) = -0.05$

จากนั้นคำนวณหาค่า Error signal ในโหนดต่างๆในชั้น Hidden layer และ Output layer

- ในชั้น Output layer

$\delta_{O1} = ½(1 - (-0.196))(1 - (-0.196)^2)$

$= 0.575$

$\delta_{O2} = ½(0 - (-0.05))(1 - (-0.05)^2)$

$= 0.524$

- ในชั้น Hidden layer

$\delta_{Y1} = ½(1 - (0.245)^2)(0.575(1) + 0.525(0.5))$

$= 0.394$

$\delta_{Y2} = ½(1 - (0.555)^2)(0.575(1) + 0.525(0.5))$

$= 0.365$

จากนั้นทำการอัพเดตค่า Weight ของเส้นเชื่อมแต่ละเส้นในแบบจำลองใหม่ โดยขอยกตัวอย่างเพียงบางเส้น คือ

- $w_{11} = w_{11} + \eta\delta_{O1}Y_1$

$= 1 + (0.1)(0.575)(0.245)$

$= 1.014$

- $v_{11} = v_{11} + \eta\delta_{Y1}Z_1$

$= 1 + (0.1)(0.394)(0.5)$

$= 1.02$

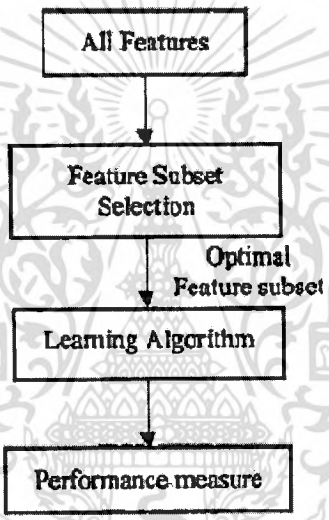
หลังจากนั้นก็ทำการคำนวณหาค่า Output ที่ชั้น Output layer ใหม่ โดยใช้ค่า Weight ที่อัพเดตใหม่ แล้วพิจารณาว่าได้ผลลัพธ์ตามที่ต้องการหรือยอมรับได้หรือไม่ ถ้ายังก็ให้ทำการหาค่า Error signal แล้วทำการอัพเดตค่า Weight ใหม่ตามวิธีข้างต้นจนกระทั่งได้ Output ที่ต้องการ

2.6 ความหมายของ Feature Selection

Feature Selection หรือที่อาจจะรู้จักกันในชื่อ Subset Selection คือกระบวนการที่ทั่วไปใช้ใน Machine learning ซึ่งเกี่ยวข้องกับการเลือกเซตย่อยของ Feature ที่มีอยู่ในข้อมูลมาใช้ในอัลกอริธึมการเรียนรู้ของเครื่อง [3] ซึ่งเซตย่อยที่ดีที่สุดจะประกอบด้วยจำนวน Feature ที่น้อย

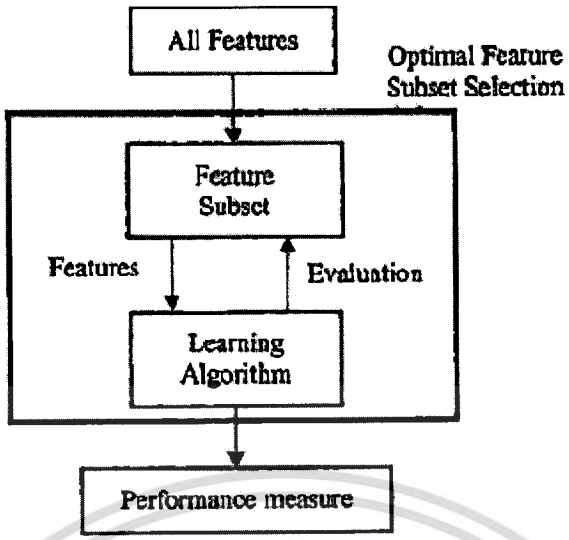
ที่สุดที่ช่วยให้แบบจำลองที่ได้มีความแม่นยำมากที่สุด หรือก็คือมีค่า Error น้อยที่สุดนั่นเอง โดยจะแบ่งออกเป็น 2 ประเภทหลักๆตามหลักการทำงาน คือ

1. Filter คือวิธีการประเภท No-feedback, Pre-selection นั่นคือเป็นอิสระจาก Machine Learning (ML) algorithm ที่ใช้งาน โดยมีวิธีการทำงานเริ่มต้นจากการวิเคราะห์หาเซตของ Feature ของข้อมูลที่มีความเหมาะสมและเกี่ยวข้องกับการนำไปใช้งานมากที่สุด โดยอาศัยเทคนิคทางสถิติ หลังจากนั้นจึงนำ Feature ของข้อมูลไปใช้งานเป็น Training data ในการสร้างแบบจำลองต่อไป-ข้อดีคือมีการทำงานที่รวดเร็วเนื่องจากเป็นอิสระจาก Classifier ทำให้ไม่ต้องเสียเวลาในการคำนวณในส่วนของ Classifier แต่ข้อเสียคือมีแนวโน้มที่จะได้เซตย่อยของ Feature ที่มีขนาดใหญ่



รูปที่ 2.8 วิธีการทำ Feature Selection แบบ Filter

2. Wrapper จะเป็นกระบวนการที่ตรงข้ามกับ Filter คือเป็นวิธีแบบ feedback นั่นคือ ML algorithm นั้นจะมีส่วนร่วมในการทำ Feature Selection โดยการทำ Wrapper จะต้องการมีประเมินผลของแบบจำลองที่ได้ในช่วง Validation แบบจำลองเพื่อนำ feedback ที่ได้มาใช้ในการปรับปรุงการทำ Feature Selection อีกครั้งหนึ่ง จนกระทั่งได้ผลลัพธ์เป็นที่พอใจ ตัวอย่างเช่น hill-climbing, random search และ Genetic Algorithms (GAs) ข้อดีของ Wrapper คือมีความแม่นยำในการคำนวณสูงกว่าแบบ Filter เนื่องจากมีการทำงานร่วมกันกับ Classifier และมีกลไกในการหลีกเลี่ยงปัญหา Overfitting โดยการใช้ Cross validation แต่ข้อเสียคือมีการทำงานที่ค่อนข้างช้าเนื่องจากต้องเสียเวลาในการทำงานในส่วนของ Classifier ด้วย



รูปที่ 2.9 วิธีการทำ Feature Selection แบบ Wrapper

2.7 อัลกอริธึมในการทำ Feature Selection

ในปัจจุบันได้มีการศึกษาถึงวิธีการทำ Feature Selection ในแบบจำลอง Classification เป็นจำนวนมาก โดยอาศัยทั้งหลักทางสถิติ, คณิตศาสตร์ หรือทางชีววิทยาเพื่อได้ซึ่งวิธีการหรืออัลกอริธึมที่มีความเหมาะสมกับลักษณะของแบบจำลองมากที่สุด ซึ่งสามารถแบ่งขั้นตอนในการทำ Feature Selection ออกเป็น 2 ช่วงด้วยกันคือ

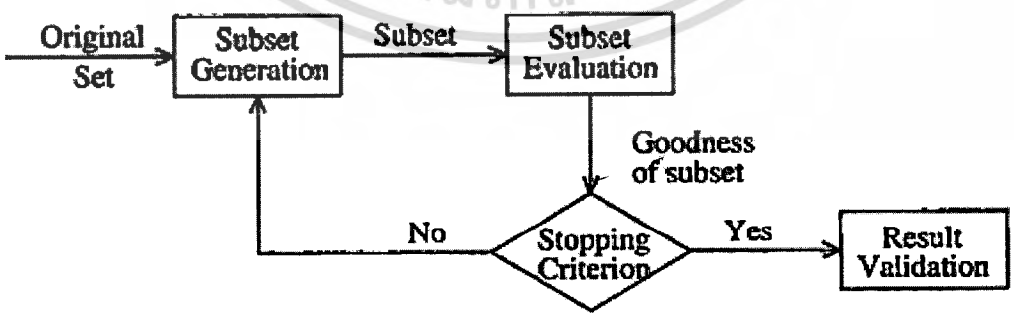
1. Generation คือ ขั้นตอนในการสร้างกลุ่มของ Feature ที่จะใช้เป็นประเมินผลเพื่อใช้เป็น Training data ต่อไป โดยจะแบ่งออกเป็น 3 ชนิดด้วยกัน คือ

   - Complete คือการสร้างกลุ่มของ Feature ที่เป็นไปได้ทั้งหมด แล้วนำไปประเมินผลเพื่อเลือกกลุ่มที่เหมาะสมที่สุดนำไปใช้งานต่อไป ข้อดีของวิธีนี้คือจะได้เซตย่อย (Subset) ของ Feature ที่ดีที่สุด แต่ข้อเสียก็คือถ้ามี Feature จำนวนมาก จะทำให้ต้องเสียเวลาในการทดลองเซตย่อยที่เป็นไปได้ทั้งหมดของ Feature ต่างๆเหล่านั้น
   - Heuristic คือการสร้างกลุ่มของ Feature โดยมีแนวทางในการเลือก Feature ที่จะนำไปใช้งานอยู่ก่อนแล้ว ข้อดีคือ ใช้จำนวนครั้งที่ทดลองไม่มากและใช้ระยะเวลาน้อยกว่าวิธีอื่นๆในการทดลอง ข้อเสียคืออาจจะพลาดโอกาสที่จะได้ Feature ที่มีความเกี่ยงข้องกับ Class ที่เป็น Output
   - Random คือการสร้างกลุ่มของ Feature โดยไม่มีการใช้แนวทางอะไรเลยในการเลือก โดยจะสุ่มเลือก Feature ขึ้นมาสร้าง โดยการจะได้เซตย่อยที่ดีหรือไม่นั้น

ขึ้นอยู่กับจำนวนครั้งที่ ผู้ใช้งานกำหนดเพื่อ Random เซตย่อยของ Feature ต่างๆ ขึ้นมาทดลอง

2. Evaluation คือขั้นตอนในการประเมินผลกลุ่ม Feature ที่จะนำไปใช้เป็น Training data ของแบบจำลอง ว่ามีความเหมาะสมมากน้อยเพียงใด โดยจะแบ่งออกได้เป็น 5 ชนิด ตามหลักเกณฑ์ที่ใช้ในการประเมินผล คือ

   - Distance คือการใช้ระยะห่างระหว่าง Feature แต่ละตัวในการประเมิน โดยอาศัยแนวคิดที่ว่า Feature ที่จัดอยู่ในกลุ่มเดียวกัน ควรจะมีระยะห่างใกล้ๆกัน
   - Information measure คือการใช้ค่า Entropy, Information gain ที่คำนวณหาได้จาก Feature ต่างๆ ในการประเมินผล
   - Dependency measure คือการใช้ความสัมพันธ์ระหว่าง Feature กับผลลัพธ์ที่ได้ของ Feature แต่ละตัว ว่ามีความสัมพันธ์มากน้อยเพียงใดในการประเมิน โดยอาจจะอาศัยค่าที่วัดได้แน่นอนจาก Distance หรือ Information ในการเปรียบเทียบ
   - Consistency measure คือการใช้ความสอดคล้องของค่าของ Feature และผลลัพธ์ที่ได้มาใช้ในการประเมินผล โดยอาศัยหลักที่ว่า Feature ที่มีค่าเหมือนกันควรจะให้ผลลัพธ์ออกมาเหมือนกัน ถ้าไม่อย่างนั้นจะถือว่า Feature เหล่านั้นไม่มีความสอดคล้องกัน (Inconsistency)
   - Classifier error rate คือการใช้ค่า Error rate ที่ได้จากการ Validation แบบจำลองมาประเมินผล โดยการประเมินผลชนิดนี้จะใช้กับวิธีการทำงานประเภท Wrapper ซึ่งจะใช้เวลาในการทำงานค่อนข้างสูง แต่ผลลัพธ์ที่ได้จะมีความแม่นยำสูง



รูปที่ 2.10 กระบวนการทำ Feature Selection

2.8 Relief

เป็นอัลกอริธึมที่ถูกคิดค้นและนำเสนอโดย Kira และ Rendell ซึ่งอาศัยการวัดค่า Weight ของ Feature แต่ละตัวและนำไปประเมินผลเทียบกับค่า Threshold ที่ user เป็นคนกำหนดในการทำ Feature Selection ซึ่ง Feature ที่จะถูกเลือกนั้นจะต้องมีค่า Weight เกินกว่าค่า Threshold โดย Relief จัดเป็นวิธีประเภท Filter ที่ถูกพัฒนาขึ้นเพื่อจัดการกับปัญหาประเภทของ Boolean (Two-class problem) เท่านั้น ต่อมาจึงได้มีการพัฒนาเป็น Relief-F ซึ่งมีความเหมาะสมในการนำไปใช้งานกับการจัดกลุ่มข้อมูลมากขึ้น ซึ่งได้มีการนำหลักการ Regression มาประยุกต์ใช้ โดยอัลกอริธึมหลักในการทำงานจะแสดงดังรูปด้านล่างนี้

RELIEF(*Dataset*, m, ...)

1. For 1 to m:
   1.1 E_1 = random example from *Dataset*.
   1.2 *Neighbours* = Find some of the nearest examples to E_1.
   1.3 For E_2 in *Neighbours*:
      1.3.1 Perform some evaluation between E_1 and E_2
2. Return the evaluation

รูปที่ 2.11 อัลกอริธึมของ Relief

Algorithm ReliefF

1. set all weights $W[A] := 0.0$;
2. **for** i := 1 **to** m **do begin**
3. randomly select an instance R_i;
4. find k nearest hits H_j;
5. **for** each class $C \neq class(R_i)$ **do**
6. from class C find k nearest misses $M_j(C)$;
7. **for** A := 1 **to** a **do**
8. $W[A] := W[A] - \sum_{j=1}^{k} \text{diff}(A,R_i,H_j)/(m \cdot k) +$
9. $\sum_{C \neq class(R_i)} [\frac{P(C)}{1-P(class(R_i))} \sum_{j=1}^{k} \text{diff}(A,R_i,M_j(C))]/(m \cdot k)$;
10. **end**;

รูปที่ 2.12 อัลกอริธึมของ Relief-F

วิธีการทำงานของ Relief คือจะหา Neighbor ที่ใกล้ที่สุดของ E_1 หนึ่งตัวจากทุก Class จากนั้นจะทำการคำนวณหาค่า Weight จาก Neighbor เหล่านี้ ใน Relief จะประเมินจากค่าความสัมพันธ์ของ Feature ทุกๆ ตัวไปคำนวณเป็นค่า Weight โดย Neighbor ที่มาจาก Class

เดียวกันกับ E_1 เรียกว่าเป็น "Near-hit instance" ในขณะที่ Neighbor ที่มาจาก Class ที่ต่างกันจะเรียกว่าเป็น "Near-miss instance" โดยค่า Weight สามารถคำนวณได้จากสมการด้านล่างนี้

$$W(f) = W(f) - diff(E_1, nearhit) + diff(E_1, nearmiss) \quad (2.6)$$

โดยค่า Weight จะมีขอบเขตอยู่ในช่วง -1 ถึง 1 เนื่องจากต้องนำมาหารด้วยจำนวนชุดข้อมูลที่สุ่มเลือก (m) ซึ่งถ้าค่า Weight ≤ 0 จะเป็นการบอกนัยๆว่า Feature นั้นไม่มีความเกี่ยงข้องกับ Class ที่เป็น Output แต่ถ้า Weight > 0 จะแสดงว่า Feature ที่มี Class ต่างกันนั้น มีระยะห่างกันตามที่คาดหวังไว้ สำหรับค่าของ diff (E_1,nearhit) และ diff (E_1, nearmiss) จะมีค่าเท่ากับ

- กรณีที่เป็นค่าไม่ต่อเนื่อง จะมีค่าเท่ากับ 0 เมื่อ ค่าของ E_1และ near-hit มีค่าเท่ากันและมีค่าเท่ากับ 1 ในกรณีอื่นๆ
- กรณีที่เป็นค่าต่อเนื่อง จะมีค่าเท่ากับ

$$\frac{|value(f,E_1) - value(f,E_2)|}{max(f) - min(f)} \quad (2.7)$$

เมื่อ max(f) คือ ค่าสูงสุดของ Feature

min(f) คือ ค่าต่ำสุดของ Feature

value(f, E_1) คือ ค่าของ Feature f ของ E_1

value(f, E_2) คือ ค่าของ Feature f ของ E_2ซึ่ง

ค่าของฟังก์ชั่น diff () ที่ได้นี้นอกจากจะนำไปใช้ในการคำนวณหาค่า Weight แล้วยังใช้ในการหา Neighbor ที่ใกล้ที่สุดอีกด้วย

สำหรับ Relief-F นั้นจะมีการนำเอาอัลกอริธึม k nearest neighbor มาใช้ในการหาค่า Weight ของ Feature ต่างๆ โดยใช้ระยะห่างจาก E_1 ในการคำนวณหาเช่นเดียวกันกับ Relief แต่จะมีการหา Near-hit instance และ Near-miss instance ได้มากกว่าหนึ่งตัวในแต่ละ Class โดยสามารถคำนวณได้จากสมการด้านล่างนี้

$$W(f) = W(f) - \frac{\sum_{j=1}^{k} diff(E_1, nearhit_k)}{(m \cdot k)} + \frac{\sum_{C \neq class(E_1)} \left[\frac{P(C)}{1 - P(class(E_1))} \sum_{j=1}^{k} diff(E_1, nearmiss_k) \right]}{(m \cdot k)} \quad (2.8)$$

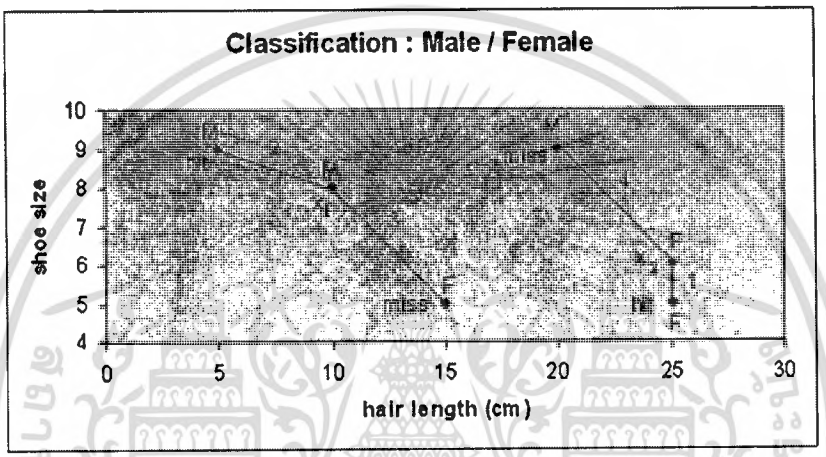
เมื่อ k คือ จำนวนของ Near-hit instance และ Near-miss instance ในการหาค่า Weight แต่ละรอบ

m คือ จำนวนค่าที่สุ่มเพื่อหาค่า Weight

P(X) คือ ความน่าจะเป็นที่จะได้ผลลัพธ์เป็น Class X

สำหรับข้อเสียของการทำ Feature Selection ด้วยวิธีนี้คือ ไม่สามารถตรวจสอบการซ้ำซ้อนของ Feature (Redundant attribute) หรือความสัมพันธ์ซ้ำซ้อน (Redundancy relation) ได้ เช่น ในกรณีที่มี Feature มีค่า Weight เท่ากันมากกว่า 1 ตัว อัลกอริธึมจะไม่สามารถเลือกที่จะใช้ Feature ใด Feature หนึ่งเพียงอย่างเดียวได้ ในขณะที่การเลือกใช้ Feature ทั้งหมดเลยก็จะทำให้แบบจำลองที่ได้ ไม่ได้รับประโยชน์จากการทำ Feature Selection เท่าที่ควร

2.8.1 ตัวอย่างการทำ Feature Selection ด้วย Relief



รูปที่ 2.13 ตัวอย่างข้อมูลการทำ Relief

จากตัวอย่างจะมี Feature ที่เป็น Input อยู่ 2 Feature คือ Shoe size และ Hair length และมี Class ที่เป็น Output ของการจัดกลุ่มอยู่ 2 Class คือ Male และ Female และมีข้อมูลอยู่ทั้งหมด 6 ชุด ตามที่แสดงเป็นจุดบนกราฟ โดยเริ่มต้น

- กำหนดค่า m = 2 และ Threshold = 0
- สุ่มข้อมูลขึ้นมาจำนวน m ชุด ในตัวอย่างคือ 2 ชุด คือ X_1 และ X_2
- คำนวณหา Near-hit instance และ Near-miss instance ของชุดข้อมูล X_1 และ X_2 จากฟังก์ชั่น diff()
- คำนวณหาค่า Weight ของ Feature ทั้ง 2 Feature
  - Shoe size

$$W(X_1) = 0 - |(8-9)/(9-5)| + |(8-5)/(9-5)|$$
$$= 0.5$$
$$W(X_2) = 0.5 - |(6-9)/(9-5)| + |(6-5)/(9-5)|$$
$$= 0$$

W(Shoe size) = 0/2 = 0

– Hair length

$W(X_1) = 0 - |(10-5)/(25-5)| + |(10-15)/(25-5)|$

$= 0$

$W(X_2) = 0 - |(25-20)/(25-5)| + |(25-25)/(25-5)|$

$= -0.25$

W(Hair Length) = - 0.25/2

= - 0.125

ดังนั้น Feature Shoe size จะถูกเลือกใช้เป็น Training data เนื่องจากมีค่าเกินกว่า Threshold ที่กำหนด

2.8.2 ตัวอย่างการทำ Feature Selection ด้วย Relief-F

Independent/Condition attributes					Dependent/ Decision attributes
Animal	**Warm-blooded**	**Feathers**	**Fur**	**Swims**	**Lays Eggs**
Ostrich	Yes	Yes	No	No	Yes
Crocodile	No	No	No	Yes	Yes
Raven	Yes	Yes	No	No	Yes
Albatross	Yes	Yes	No	No	Yes
Dolphin	Yes	No	No	Yes	No
Koala	Yes	No	Yes	No	No

รูปที่ 2.14 ตัวอย่างข้อมูลการทำ Relief-F

จากตัวอย่างด้านบน จะมี Feature ของข้อมูล อยู่ 4 กลุ่มด้วยกัน คือ คือ Warm-blooded, Feathers, Fur, Swims และมี Class เป้าหมายคือ Lays Eggs โดยกำหนดให้จำนวนของ Near-hit instance และ Near-miss instance ที่ใช้ในการคำนวณค่า Weight แต่ละรอบ (k) เท่ากับ 2 และจำนวนค่าที่สุ่ม (m) เท่ากับ 2 และมีค่า Threshold = 0 สมมติให้สุ่มได้ข้อมูลชุดในแถวที่ 2 และ 4 ในทุกๆ Feature จะได้ว่า

– Warm-blooded

W(Crocodile) = 0 – (1 + 1) + [(2/6)*(6/2))(1 + 1)]

= (-2 + 2)/4 = 0

W(Albatross) = 0 – (0 + 0) + [(2/6)*(6/2))(0 + 0)]

= 0/4 = 0

W(Warm-blooded) = 0

– Feathers

W(Crocodile) = 0 – (1 + 1) + [(2/6)*(6/2))(0 + 0)]

= -2/4 = - 0.5

W(Albatross) = - 0.5 – (0 + 0) + [(2/6) *(6/2) (1 + 1)]

= - 0.5 + (2/4) = 0

W(Feathers) = 0

– Fur

W(Crocodile) = 0 – (0 + 0) + [(2/6)*(6/2))(0 + 1)]

= 1/4 = 0.25

W(Albatross) = 0.25 – (0 + 0) + [(2/6) *(6/2) (0 + 1)]

= 0.25 + (1/4) = 0.5

W(Fur) = 0.5

– Swims

W(Crocodile) = 0 – (1 + 1) + [(2/6)*(6/2))(0 + 1)]

= -1/4 = - 0.25

W(Albatross) = - 0.25 – (0 + 0) + [(2/6) *(6/2) (0 + 1)]

= - 0.25 + (1/4) = 0

W(Swims) = 0

ดังนั้นทุก Feature ของข้อมูลจะถูกเลือกเลือกใช้เป็น Training data เนื่องจากมีค่าเกินกว่า Threshold ที่กำหนดไว้

2.9 Fast Correlation-Based Filter (FCBF)

เป็นอัลกอริธึมที่อาศัย "Correlation-Based measure" และแนวคิดในเรื่อง "Predominant correlation" ในการประเมินความสัมพันธ์ (Relevant) ของ Feature ต่างๆเพื่อใช้ในการจัดกลุ่มข้อมูล ซึ่งได้รับการพัฒนามาจากอัลกอริธึม "Correlation-Based Filter Solution " หรือ CFS โดยใช้หลักการที่ว่า Feature ที่ดีต้องมีความสัมพันธ์กับ Class ที่เป็น Feature เป้าหมายของการจัดกลุ่มข้อมูลและต้องไม่มีความซ้ำซ้อน (Redundant) กับ Feature อื่นๆที่มีความสัมพันธ์กับ Class ที่เป็น Feature เป้าหมายด้วยเช่นกัน นั่นแปลว่า Feature นั้นต้องมีค่าความสัมพันธ์กับ Class เป้าหมายที่

สูงในระดับที่กำหนดไว้และมีค่าความสัมพันธ์กับ Feature ที่เหลืออื่นๆไม่สูงกว่าค่าความสัมพันธ์กับ Class เป้าหมาย ซึ่งโดยทั่วไปจะมีวิธีที่ใช้ในการวัดค่าความสัมพันธ์ด้วยกัน 2 วิธี คือ

- Based on classical linear correlation ซึ่งวิธีที่เป็นที่รู้จักกันโดยทั่วไป จะอาศัยการวัดที่เรียกว่า "Linear correlation efficient" โดยสำหรับคู่ลำดับ (X,Y) ใดๆ ค่า Linear correlation efficient (r) จะมีค่าดังนี้

$$r = \frac{\sum_i (x_i - \overline{x_i})(y_i - \overline{y_i})}{\sqrt{\sum_i (x_i - \overline{x_i})^2}\sqrt{\sum_i (y_i - \overline{y_i})^2}} \tag{2.9}$$

เมื่อ $\overline{x_i}$ คือ ค่าเฉลี่ยของ X

$\overline{y_i}$ คือ ค่าเฉลี่ยของ Y

ซึ่ง r จะมีค่าระหว่าง –1 ถึง 1 โดยถ้า X และ Y เป็นอิสระต่อกัน ค่า r จะมีค่าเท่ากับ 0 นอกจากวิธีด้านบนแล้ว ยังมีวิธีการวัดอื่นๆที่สามารถใช้ได้เช่นกัน เช่น "Least square regression error" หรือ "Maximal information compression index" เป็นต้น แต่เนื่องจากวิธีการเหล่านี้อาศัยสมมติฐานว่าความสัมพันธ์ของตัวแปรนั้นเป็นความสัมพันธ์เชิงเส้น (Linear correlation) จึงไม่สามารถที่จะนำไปประยุกต์ใช้ได้กับทุกๆกรณีในโลกของความเป็นจริงได้

- Based on information theory ซึ่งเป็นวิธีที่อาศัยแนวคิดทฤษฎีทาง Information หรือ "Entropy" ในการหาค่าความสัมพันธ์ของตัวแปรต่างๆ โดยค่า Entropy ของ X จะมีค่าเท่ากับ

$$H(X) = -\sum_i P(x_i)\log_2(P(x_i)) \tag{2.10}$$

เมื่อ i คือ จำนวนค่าที่แตกต่างกันในตัวแปร X

ค่า Entropy ของ X เมื่อพิจารณาค่าของตัวแปร Y ร่วมด้วย จะมีค่าเท่ากับ

$$H(X|Y) = -\sum_j P(y_j)\sum_i P(x_i|y_j)\log_2(P(x_i|y_j)) \tag{2.11}$$

เมื่อ j คือ จำนวนค่าที่แตกต่างกันในตัวแปร Y

ค่า Information gain ของ X เมื่อพิจารณาค่าของตัวแปร Y ร่วมด้วย จะมีค่าเท่ากับ

$$IG(X|Y) = H(X) - H(X|Y) \tag{2.12}$$

โดยอาศัยค่า Information gain นี้เราสามารถหาค่าความสัมพันธ์ของตัวแปรต่างๆ ได้ โดยถ้า X มีความสัมพันธ์กับ Y มากกว่า Z แล้ว $IG(X|Y)$ จะมีค่ามากกว่า $IG(X|Z)$ แต่เนื่องจากว่าการใช้ Information gain มีแนวโน้มที่มีอคติ (Bias) โดยจะเอนเอียงต่อตัวแปรที่มีค่าของตัวแปรแตกต่างกันเป็นจำนวนมาก ด้วยเหตุผลนี้รวมทั้งการที่ค่าของตัวแปรต่างๆสมควรที่จะมีการทำ Normalized เพื่อทำให้มั่นใจว่าตัวแปรต่างๆเหล่านี้สามารถนำมาใช้เปรียบเทียบกันได้ จึงมีการนำการวัดค่าที่เรียกว่า "Symmetrical uncertainty" มาใช้ในการวัดค่าความสัมพันธ์แทน โดยจะมีสูตรดังนี้

$$SU(X,Y) = 2\left[\frac{IG(X|Y)}{H(X)+H(Y)}\right] \tag{2.13}$$

สำหรับค่า Symmetrical uncertainty ที่ได้นั้นจะมีค่าระหว่าง [0,1] โดยเมื่อมีค่าเท่ากับ 1 จะแสดงว่าเมื่อทราบค่าของตัวแปรหนึ่งแล้วจะสามารถทำนายค่าของอีกตัวแปรหนึ่งได้ด้วย แต่ถ้ามีค่าเท่ากับ 0 จะแสดงว่าตัวแปรทั้งสองจะเป็นอิสระต่อกัน สำหรับอัลกอริธึมการทำงานของ FCBF โดยอาศัยค่า SU จะมีการทำงานตามรูปด้านล่างนี้

```
input:  S(F_1, F_2, ..., F_N, C) // a training data set
        δ                        // a predefined threshold
output: S_best                   // an optimal subset
1   begin
2     for i = 1 to N do begin
3       calculate SU_{i,c} for F_i;
4       if (SU_{i,c} ≥ δ)
5         append F_i to S'_list;
6     end;
7     order S'_list in descending SU_{i,c} value;
8     F_p = getFirstElement(S'_list);
9     do begin
10      F_q = getNextElement(S'_list, F_p);
11      if (F_q <> NULL)
12        do begin
13          F'_q = F_q;
14          if (SU_{p,q} ≥ SU_{q,c})
15            remove F_q from S'_list;
16            F_q = getNextElement(S'_list, F'_q);
17          else F_q = getNextElement(S'_list, F_q);
18        end until (F_q == NULL);
19      F_p = getNextElement(S'_list, F_p);
20    end until (F_p == NULL);
21  S_best = S'_list;
22  end;
```

รูปที่ 2.15 อัลกอริธึมของ Fast Correlation-Based Filter

จากการทำงานของอัลกอรธึม เราจะสามารถหาเซตย่อยของ Feature ที่เหมาะสมสำหรับการจัดกลุ่มข้อมูลได้ โดยการทำงาน 2 ขั้นตอนด้วยกัน คือ ขั้นตอนแรกจะตรวจสอบว่าตัวแปรใดที่มีความสัมพันธ์กับ Class ที่เป็นเป้าหมาย ด้วยการกำหนดค่า Threshold (δ) โดยตัวแปรใดที่มีค่า SU $\geq \delta$ จะถือว่าตัวแปรนั้นมีความสัมพันธ์กับ Class เป้าหมาย และจะนำมาเก็บไว้ในกลุ่มของเซตย่อยขั้นตอนสุดท้ายคือการตรวจสอบความซ้ำซ้อนของตัวแปร โดยการนำตัวแปรภายในเซตย่อยมาเรียงลำดับตามค่า SU ตามลำดับ หลังจากนั้นจะหาค่าของ $SU_{p,\ q}$ (ค่า SU ระหว่าง Feature P และ Feature Q) และ $SU_{q,\ c}$ (ค่า SU ระหว่าง Feature Q และ Class เป้าหมาย C) นำมาเปรียบเทียบกัน เริ่มต้นให้ P คือ Feature ที่มีค่า SU มากที่สุด และ Q คือ Feature ที่มีค่า SU รองลงมา โดยจะมีการเปลี่ยน Feature P และ Q ไปเรื่อยๆ ซึ่ง P มีค่า SU มากกว่า Q เสมอ จนกระทั่งทุกๆ Feature ที่อยู่ในเซตย่อยถูกนำมาเปรียบเทียบกันจนครบทุกตัว เพื่อกำจัด Feature ที่มีความซ้ำซ้อนออกไป ก็จะได้ผลลัพธ์ที่ต้องการ

2.10 Las Vegas Filter (LVF)

อัลกอริธึม Las Vegas Filter นั้นเป็นอัลกอริธึมอาศัยแนวคิดเรื่องความน่าจะเป็น (Probabilisty) ในการทำ Feature Selection โดยการสุ่มเลือกเซตย่อยของ Feature ที่เป็นไปได้ขึ้นมา โดยในแต่ละเซตย่อยที่ถูกสุ่มเลือกขึ้นมานั้นจะถูกนำมาเปรียบเทียบกับเซตย่อยที่ดีที่สุดในขณะนั้น ถ้ามีจำนวนของ Feature น้อยกว่าเซตย่อยที่ดีที่สุดในขณะนั้น และมีค่าวัดที่เรียกว่า "Inconsistency rate" น้อยกว่าค่า Inconsistency rate ที่กำหนดไว้ (โดยปกติแล้วจะมีค่า Default เท่ากับ 0) แล้วเซตย่อยนั้นจะถูกเลือกเป็นเซตย่อยที่ดีที่สุดในขณะนั้นแทนที่เซตย่อยเก่า โดยจำนวนครั้งในการสุ่มเลือกเซตย่อย (MAX-TRIES) นั้นผู้ใช้งานจะเป็นผู้กำหนดขึ้นมาเอง

สำหรับค่า Inconsistency rate นั้นถือว่าเป็นหัวใจสำคัญสำหรับอัลกอริธึม LVF ซึ่งจะเป็นตัวชี้วัดในการที่จะเลือกว่าเซตย่อยใดนั้นเหมาะสมที่สุดโดย Inconsistency rate จะมีค่าเท่ากับผลรวม Inconsistency count ของรูปแบบข้อมูลที่เป็นไปได้ทั้งหมดหารด้วยจำนวนของแถวข้อมูลทดลองทั้งหมด ซึ่งชุดข้อมูลสองชุดจะถือว่า Inconsistent กันเมื่อชุดข้อมูลทั้งคู่มีค่าของ Feature เหมือนกันทุกประการแต่จัดอยู่ใน Class เป้าหมายคนละ Class สำหรับค่า Inconsistency count ของแต่ละรูปแบบข้อมูลนั้นจะมีค่าเท่ากับจำนวนของแถวชุดข้อมูลที่เหมือนกันทั้งหมดโดยไม่พิจารณา Class เป้าหมายลบด้วยจำนวนแถวที่มากที่สุดเมื่อพิจารณาแยกตามกลุ่ม Class เป้าหมายที่แตกต่างกัน ตัวอย่างเช่น ชุดข้อมูลชุดหนึ่งมีจำนวนแถวข้อมูลที่เหมือนกันทั้งหมด n แถว โดยจัดอยู่ใน Class C_1 จำนวน c_1 แถว, จัดอยู่ใน Class C_2 จำนวน c_2 แถว และจัดอยู่ใน Class C_3 จำนวน c_3 แถว ซึ่ง c_3 มีค่ามากที่สุด จะได้ว่า Inconsistency count เท่ากับ ($n - c_3$) โดยจะมีอัลกอริธึมในการทำงานดังนี้

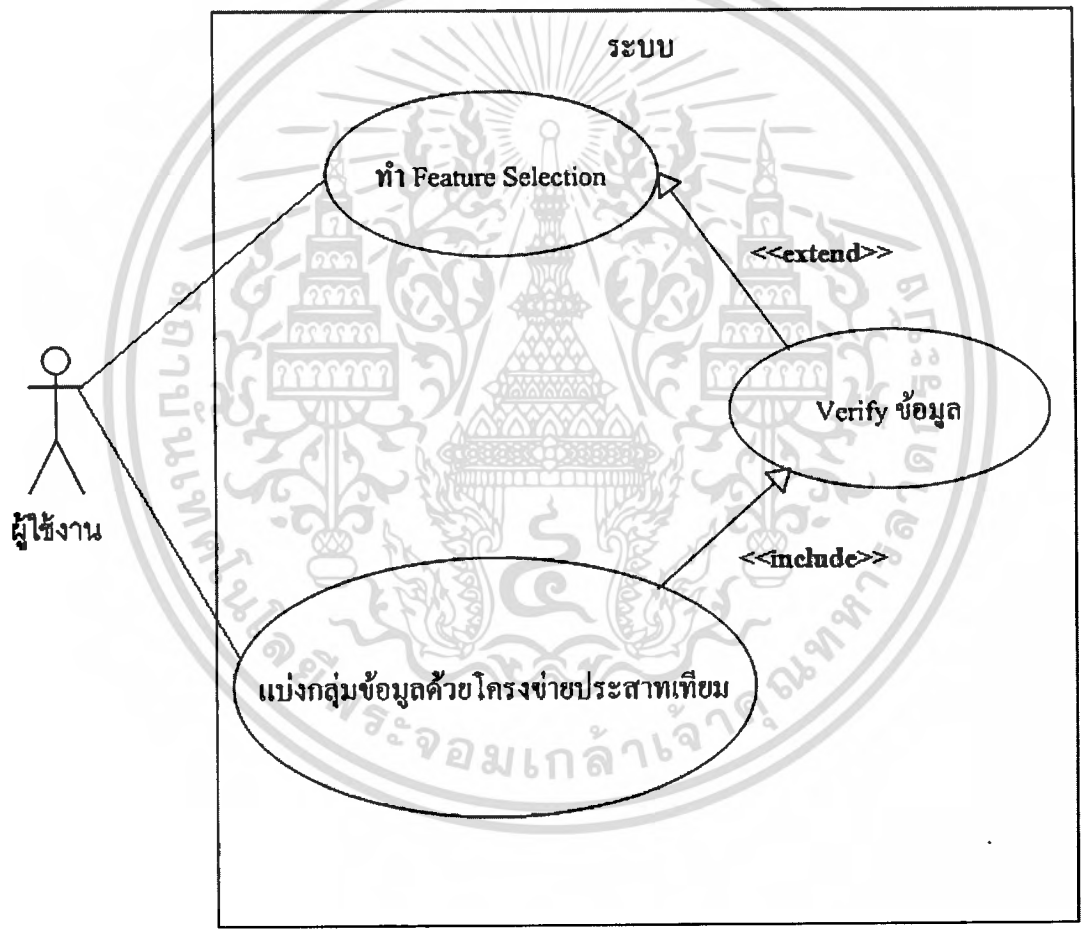
```
LVF algorithm
    Input: MAX-TRIES,
           D - dataset,
           N - number of attributes;
           γ - allowable inconsistency rate,
    Output: sets of M features satisfying
            the inconsistency criterion
    C_best = N;
    for i=1 to MAX-TRIES
       S = randomSet(seed);
       C = numOfFeatures(S);
       if (C < C_best)
          if (InconCheck(S, D) < γ);
             S_best = S; C_best = C;
             print_Current_Best(S)
       else if ((C = C_best) and
                (InconCheck(S, D) < γ))
          print_Current_Best(S)
    end for
```

รูปที่ 2.16 อัลกอริธึมของ Las Vegas Filter

จากการทำงานของอัลกอริธึม LVF นั้นจะสังเกตได้ว่าเซตย่อยของผลลัพธ์ที่ได้นั้นจะดีหรือไม่นั้น ส่วนหนึ่งขึ้นอยู่กับจำนวนครั้งในการเลือกสุ่มชุดเซตย่อยขึ้นมา ถ้ามีจำนวนครั้งน้อยเกินไปอาจจะทำให้พลาดโอกาสที่จะได้เซตย่อยที่ดีที่สุด แต่ในขณะเดียวกันถ้ามีจำนวนครั้งมากเกินไปก็จะทำให้การทำงานของอัลกอริธึมใช้ระยะเวลานานกว่าที่จะได้เซตย่อยที่ดีที่สุด ซึ่งจุดนี้ผู้ใช้งานจะต้องเป็นผู้พิจารณาว่าจำนวนครั้งเท่าไหร่ถึงจะมีความเหมาะสม ที่ผู้ใช้งานจะยอมรับได้ในเรื่องของเซตย่อยที่ถูกเลือกเป็นผลลัพธ์และระยะเวลาที่ใช้ในการทำงาน

บทที่ 3
การออกแบบโปรแกรมประยุกต์

ในการศึกษาเกี่ยวกับการศึกษาเปรียบเทียบประสิทธิภาพของอัลกอริธึมในการทำ Feature Selection เพื่อใช้ในการแบ่งกลุ่มข้อมูล ผู้จัดทำรายงานได้พัฒนาโปรแกรมประยุกต์เพื่อใช้ในการทดลองด้วยภาษา VB.NET โดยออกแบบให้มีการแบ่งการทำงานหลักๆออกเป็น 2 ส่วนด้วยกัน คือ ส่วนที่เกี่ยวข้องกับการทำ Feature Selection และส่วนที่เกี่ยวข้องกับการแบ่งกลุ่มข้อมูลด้วยโครงข่ายประสาทเทียมแบบ Backpropagation โดยสามารถเขียนเป็น Use case diagram ได้ดังนี้



รูปที่ 3.1 Use case diagram ของโปรแกรม

สำหรับในแต่ละ Use case จะสามารถเขียนรายละเอียดการทำงานได้ โดยมีรายละเอียดดังต่อไปนี้

ตารางที่ 3.1 Use case description ของ Use case ทำ Feature Selection

<table>
<tr><td>Use-Case Name :</td><td colspan="2">ทำ Feature Selection</td></tr>
<tr><td>Scenario :</td><td colspan="2">การทำงานหลัก</td></tr>
<tr><td>Trigger Event :</td><td colspan="2">เลือกทำงานในส่วนของ Feature Selection</td></tr>
<tr><td>Brief Description :</td><td colspan="2">เมื่อผู้ใช้เลือกทำ Feature Selection ระบบจะทำการคำนวณและประเมินค่าน้ำหนักของแต่ละแอททริบิวต์ตามอัลกอริธึมที่ผู้ใช้งานเลือกและแสดงแอททริบิวต์ที่ผ่านการประเมินให้ผู้ใช้งานเมื่อจบการทำงาน</td></tr>
<tr><td>Actor :</td><td colspan="2">ผู้ใช้</td></tr>
<tr><td>Related Use Cases :</td><td colspan="2">Verify ข้อมูล</td></tr>
<tr><td>Preconditions :</td><td colspan="2">-</td></tr>
<tr><td>Postconditions :</td><td colspan="2">แสดงข้อมูลแอททริบิวต์ที่ผ่านการประเมินและค่าน้ำหนักของแต่ละแอททริบิวต์</td></tr>
<tr><td rowspan="2">Flow of Events :</td><td>Actor</td><td>System</td></tr>
<tr><td>1. เลือกไฟล์ข้อมูล Training และ Testing
3. เลือกอัลกอริธึมที่ต้องการทำ Feature Selection และกำหนดค่า Threshold
5. กดปุ่มเพื่อเริ่มทำงาน

7. กดปุ่มทำงานในขั้นถัดไป</td><td>2. อ่านไฟล์ข้อมูลเก็บไว้ในหน่วยความจำ
4. เก็บค่า Threshold และอัลกอริธึมที่ผู้ใช้กำหนด
6.1 ในกรณีที่ผู้ใช้เลือก Relief-F จะแสดงตัวอย่างข้อมูลนำเข้าพร้อมทั้งการ Verify ข้อมูล
6.2 ในกรณีที่ผู้ใช้เลือก FCBF และ LVF จะแสดงตัวอย่างข้อมูลนำเข้าเท่านั้น
8. คำนวณและประเมินแอททริบิวต์ตามอัลกอริธึมที่ผู้ใช้งานเลือกก่อนหน้านี้และแสดงผลให้ผู้ใช้งาน</td></tr>
<tr><td>Exception Conditions :</td><td colspan="2">-</td></tr>
</table>

ตารางที่ 3.2 Use case description ของ Use case แบ่งกลุ่มข้อมูลด้วยโครงข่ายประสาทเทียม

<table>
<tr><td>Use-Case Name :</td><td colspan="2">แบ่งกลุ่มข้อมูลด้วยโครงข่ายประสาทเทียม</td></tr>
<tr><td>Scenario :</td><td colspan="2">การทำงานหลัก</td></tr>
<tr><td>Trigger Event :</td><td colspan="2">เลือกทำงานในส่วนของการแบ่งกลุ่มข้อมูล</td></tr>
<tr><td>Brief Description :</td><td colspan="2">เมื่อผู้ใช้เลือกแบ่งกลุ่มข้อมูลด้วยโครงข่ายประสาทเทียม ระบบทำการสร้างโครงข่ายประสาทเทียมตามที่ผู้ใช้ระบุ และสอนโครงข่ายด้วยข้อมูลในไฟล์ Training และทดสอบความถูกต้องด้วยข้อมูลในไฟล์ Testing และแสดงผลลัพธ์การทำงานให้กับผู้ใช้</td></tr>
<tr><td>Actor :</td><td colspan="2">ผู้ใช้</td></tr>
<tr><td>Related Use Cases :</td><td colspan="2">Verify ข้อมูล</td></tr>
<tr><td>Preconditions :</td><td colspan="2">-</td></tr>
<tr><td>Postconditions :</td><td colspan="2">แสดงผลลัพธ์ความถูกต้องในการทำงานของการแบ่งกลุ่มข้อมูลด้วยโครงข่ายประสาทเทียมที่สร้างขึ้น</td></tr>
<tr><td rowspan="2">Flow of Events :</td><td>Actor</td><td>System</td></tr>
<tr><td>1. เลือกไฟล์ข้อมูล Training และ Testing
3. กำหนดค่าต่างๆให้กับโครงข่ายประสาทเทียม
5. กดปุ่มเพื่อเริ่มทำงาน
7. ระบุประเภทของแต่ละของแอททริบิวต์และแอททริบิวต์ที่จะใช้งาน
9. กดปุ่มทำงานในขั้นถัดไป</td><td>2. อ่านไฟล์ข้อมูลเก็บไว้ในหน่วยความจำ
4. เก็บค่าต่างๆของโครงข่ายประสาทเทียม
6. แสดงตัวอย่างข้อมูลนำเข้าพร้อมทั้งการ Verify ข้อมูล
8. เก็บรายละเอียดประเภทและตำแหน่งของแอททริบิวต์ที่ผู้ใช้งานเลือก
10. สร้างโครงข่ายประสาทเทียมและแสดงผลลัพธ์การทำงานให้กับผู้ใช้</td></tr>
<tr><td>Exception Conditions :</td><td colspan="2">-</td></tr>
</table>

ตารางที่ 3.3 Use case description ของ Use case Verify ข้อมูล

Use-Case Name :	Verify ข้อมูล	
Scenario :	การทำงานหลัก	
Trigger Event :	เลือกในส่วนของ Feature Selection หรือการแบ่งกลุ่มข้อมูล	
Brief Description :	เมื่อผู้ใช้เลือกทำ Feature Selection ด้วยอัลกอริธึม Relief-F หรือเลือกแบ่งกลุ่มข้อมูลด้วยโครงข่ายประสาทเทียม ระบบจะให้ผู้ใช้ระบุตำแหน่งและประเภทของแอททริบิวต์ที่ต้องการใช้งาน	
Actor :	ผู้ใช้	
Related Use Cases :	ทำ Feature Selection, แบ่งกลุ่มข้อมูลด้วยโครงข่ายประสาทเทียม	
Preconditions :	ผู้ใช้เลือกทำ Feature Selection ด้วยอัลกอริธึม Relief-F หรือเลือกทำแบ่งกลุ่มข้อมูลด้วยโครงข่ายประสาทเทียม	
Postconditions :	โปรแกรมนำแอททริบิวต์ของข้อมูลที่ผู้ใช้เลือกไปใช้ในการทำงานขั้นต่อไป	
Flow of Events :	**Actor**	**System**
	1.1 กรณีที่เลือกทำ Feature Selection ด้วยอัลกอริธึม Relief-F จะระบุประเภทของแต่ละของแอททริบิวต์ 1.2 กรณีที่เลือกทำแบ่งกลุ่มข้อมูลด้วยโครงข่ายประสาทเทียม จะระบุประเภทของแต่ละของแอททริบิวต์และแอททริบิวต์ที่จะใช้งาน	2.1 โปรแกรมเก็บรายละเอียดประเภทของแอททริบิวต์ที่ผู้ใช้งานเลือก 2.2 โปรแกรมเก็บรายละเอียดประเภทและตำแหน่งของแอททริบิวต์ที่ผู้ใช้งานเลือก
Exception Conditions :	-	

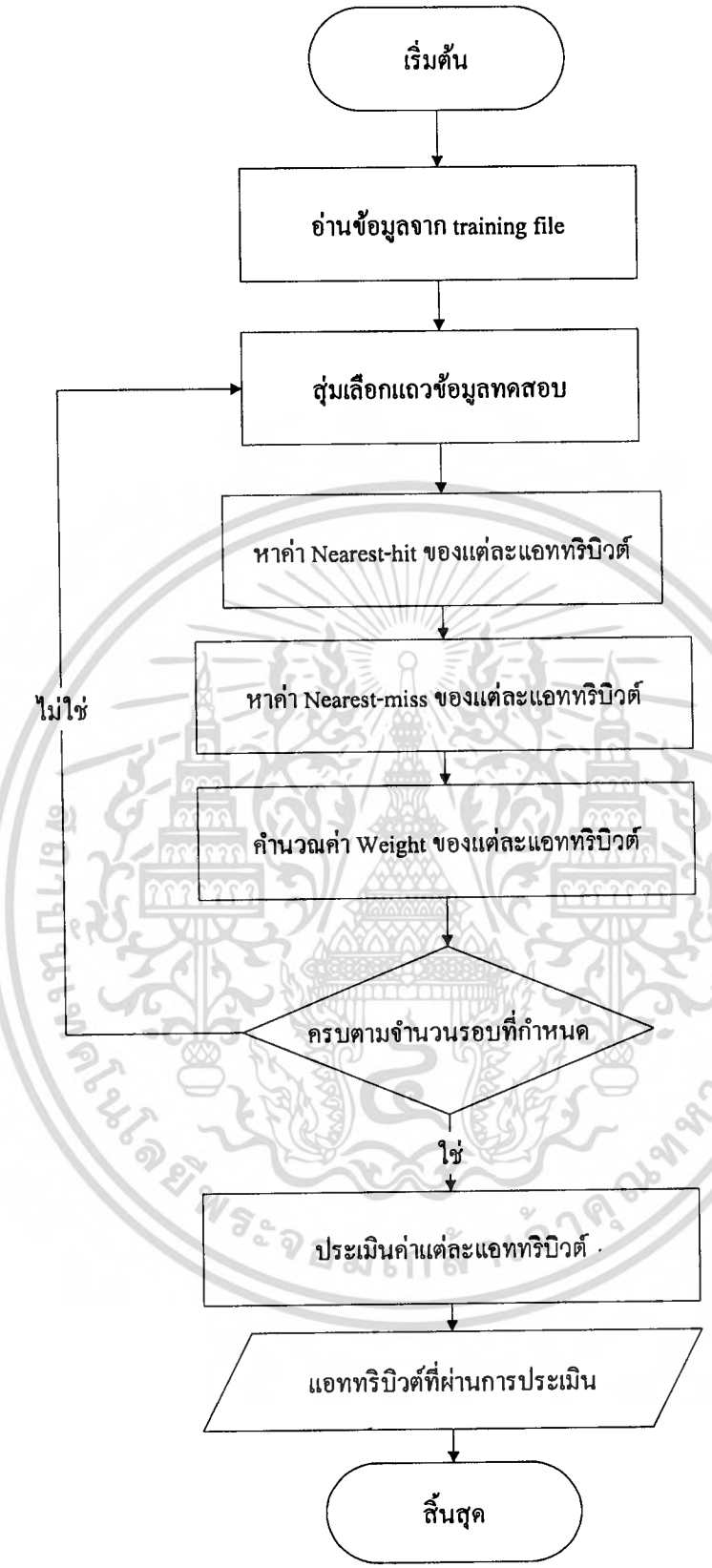
3.1 ขั้นตอนการทำ Feature Selection

สำหรับในขั้นตอนการทำ Feature Selection นี้ ผู้ใช้งานสามารถที่จะเลือกอัลกอริธึมที่จะใช้การทำงานได้ 3 อัลกอริธึมด้วยกัน ประกอบด้วย Relief-F, Fast Correlation-Based Filter (FCBF) และ Las Vegas Filter (LVF) โดยในแต่ละอัลกอริธึมนั้นจะมีขั้นตอนการทำงานที่แตกต่างกันออกไปซึ่งมีรายละเอียดขั้นตอนการทำงานดังนี้

3.1.1 Relief-F

ในการทำงานของ Relief-F นั้นจะแบ่งการทำงานออกเป็นขั้นตอนย่อยๆ โดยมีลำดับการทำงาน ดังนี้

1. อ่านข้อมูลจากไฟล์ Training ที่ผู้ใช้งานเลือกไว้ในหน้าจอการทำงานหลักและหน้าจอ Verify และแสดงตัวอย่างข้อมูล มาเก็บไว้ในหน่วยความจำของโปรแกรม
2. สุ่มกำหนดเลขที่ลำดับแถวของข้อมูล Training ที่อ่านไว้ โดยจำนวนครั้งที่สุ่มจะมีค่าประมาณ 10% ของจำนวนแถวข้อมูลทั้งหมด
3. ค้นหาค่าของข้อมูลในแต่ละแอททริบิวต์ที่มีค่าใกล้เคียงกับค่าของแอททริบิวต์ของแถวที่สุ่มเลือกมา โดยแยกเป็นค่าออกเป็น 2 กลุ่ม คือ กลุ่มที่มี Class เป้าหมายเดียวกับแถวที่สุ่มเลือกไว้ และกลุ่มที่มี Class เป้าหมายไม่ตรงกับแถวที่สุ่มเลือกไว้ ซึ่งการคำนวณค่าระยะห่างระหว่างตัวแปรในแต่ละแอททริบิวต์นั้นจะใช้หลักเกณฑ์ดังที่ได้อธิบายไว้ในบทที่ 2 ในส่วนของอัลกอริธึม Relief-F
4. คำนวณค่า Weight ของแต่ละแอททริบิวต์ โดยเริ่มต้นให้ค่า Weight ของแต่ละแอททริบิวต์มีค่าเท่ากับ 0 และอัพเดตค่า Weight ใหม่ทุกครั้งที่ทำงานในแต่ละรอบ
5. กลับไปทำงานใหม่ตั้งแต่ข้อ 3 จนครบตามจำนวนรอบที่กำหนดไว้
6. ประเมินค่า Weight ของแต่ละแอททริบิวต์ และแสดงรายชื่อแอททริบิวต์ที่มีค่า Weight เกินกว่าค่า Threshold ที่ผู้ใช้งานกำหนดไว้

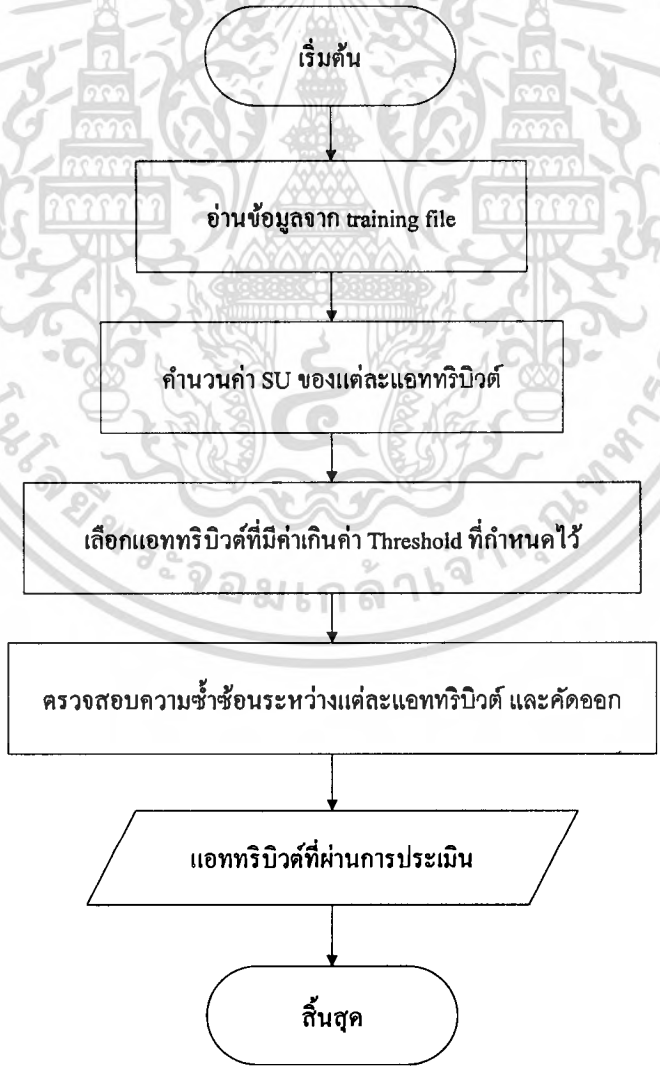


รูปที่ 3.2 Flowchart ของการทำ Feature Selection ด้วย Relief-F

3.1.2 Fast Correlation-Based Filter (FCBF)

ในการทำงานของ Fast Correlation-Based Filter นั้นจะแบ่งการทำงานออกเป็นขั้นตอนย่อยๆ โดยมีลำดับการทำงาน ดังนี้

1. อ่านข้อมูลจากไฟล์ Training ที่ผู้ใช้งานเลือกไว้ในหน้าจอการทำงานหลักและหน้าจอ Verify และแสดงตัวอย่างข้อมูล มาเก็บไว้ในหน่วยความจำของโปรแกรม
2. คำนวณค่า Symmetrical uncertainty (SU) ของแต่ละแอททริบิวต์ และเลือกแอททริบิวต์ที่มีค่าเกินค่า Threshold ที่ผู้ใช้งานกำหนดเก็บไว้ โดยวิธีการคำนวณค่า SU นั้นจะเป็นไปตามที่ได้อธิบายไว้ในบทที่ 2 ในส่วนของอัลกอริธึม Fast Correlation-Based Filter
3. ตรวจสอบความซ้ำซ้อนโดยเปรียบเทียบค่า SU ระหว่างแต่ละแอททริบิวต์ที่คัดเลือกเก็บไว้และคัดแอททริบิวต์ที่มีความซ้ำซ้อนออกจากกลุ่ม
4. แสดงรายชื่อแอททริบิวต์ที่ผ่านการประเมิน

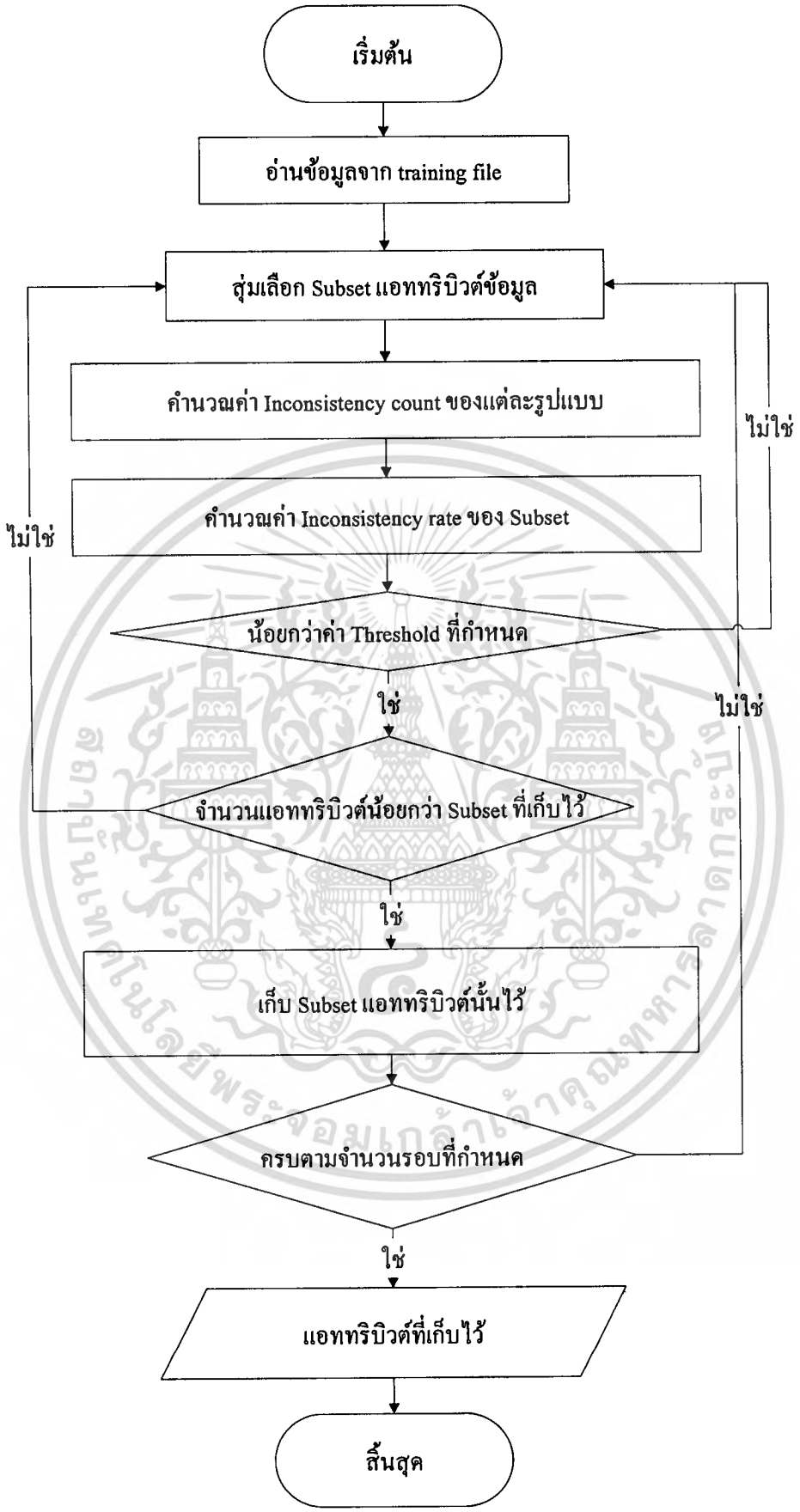


รูปที่ 3.3 Flowchart ของการทำ Feature Selection ด้วย Fast Correlation-Based Filter

3.1.3 Las Vegas Filter (LVF)

ในการทำงานของ Las Vegas Filter นั้นจะแบ่งการทำงานออกเป็นขั้นตอนย่อยๆ โดยมีลำดับการทำงาน ดังนี้

1. อ่านข้อมูลจากไฟล์ Training ที่ผู้ใช้งานเลือกไว้ในหน้าจอการทำงานหลักและหน้าจอ Verify และแสดงตัวอย่างข้อมูล มาเก็บไว้ในหน่วยความจำของโปรแกรม
2. สุ่มเลือกเซตย่อย (Subset) ของกลุ่มแอททริบิวต์ข้อมูล โดยจำนวนครั้งที่สุ่มจะมีค่าประมาณ 77 ครั้ง
3. คำนวณค่า Inconsistency count ของรูปแบบข้อมูลที่ประกอบด้วยเซตย่อยแอททริบิวต์ที่สุ่มเลือกมา โดยวิธีการคำนวณค่า Inconsistency count นั้นจะเป็นไปตามที่ได้อธิบายไว้ในบทที่ 2 ในส่วนของอัลกอริธึม Las Vegas Filter
4. คำนวณค่า Inconsistency rate ของเซตย่อยแอททริบิวต์ที่สุ่มเลือกมา โดยวิธีการคำนวณค่า Inconsistency rate นั้นจะเป็นไปตามที่ได้อธิบายไว้ในบทที่ 2 ในส่วนของอัลกอริธึม Las Vegas Filter
5. ถ้าค่า Inconsistency rate ของเซตย่อยแอททริบิวต์ที่สุ่มเลือกมานั้น มีค่าน้อยกว่าค่า Threshold ที่ผู้ใช้งานกำหนด จะนำมาเปรียบเทียบจำนวนแอททริบิวต์ของเซตย่อยที่เก็บไว้ก่อนหน้านี้ ถ้าน้อยกว่าจะเก็บเซตย่อยแอททริบิวต์นั้นไว้แทน
6. กลับไปทำงานใหม่ตั้งแต่ข้อ 2 จนครบตามจำนวนรอบที่กำหนดไว้
7. แสดงรายชื่อเซตย่อยของแอททริบิวต์ที่ระบบเก็บไว้



รูปที่ 3.4 Flowchart ของการทำ Feature Selection ด้วย Las Vegas Filter

3.2 ขั้นตอนการแบ่งกลุ่มข้อมูลด้วยโครงข่ายประสาทเทียมแบบ Backpropagation

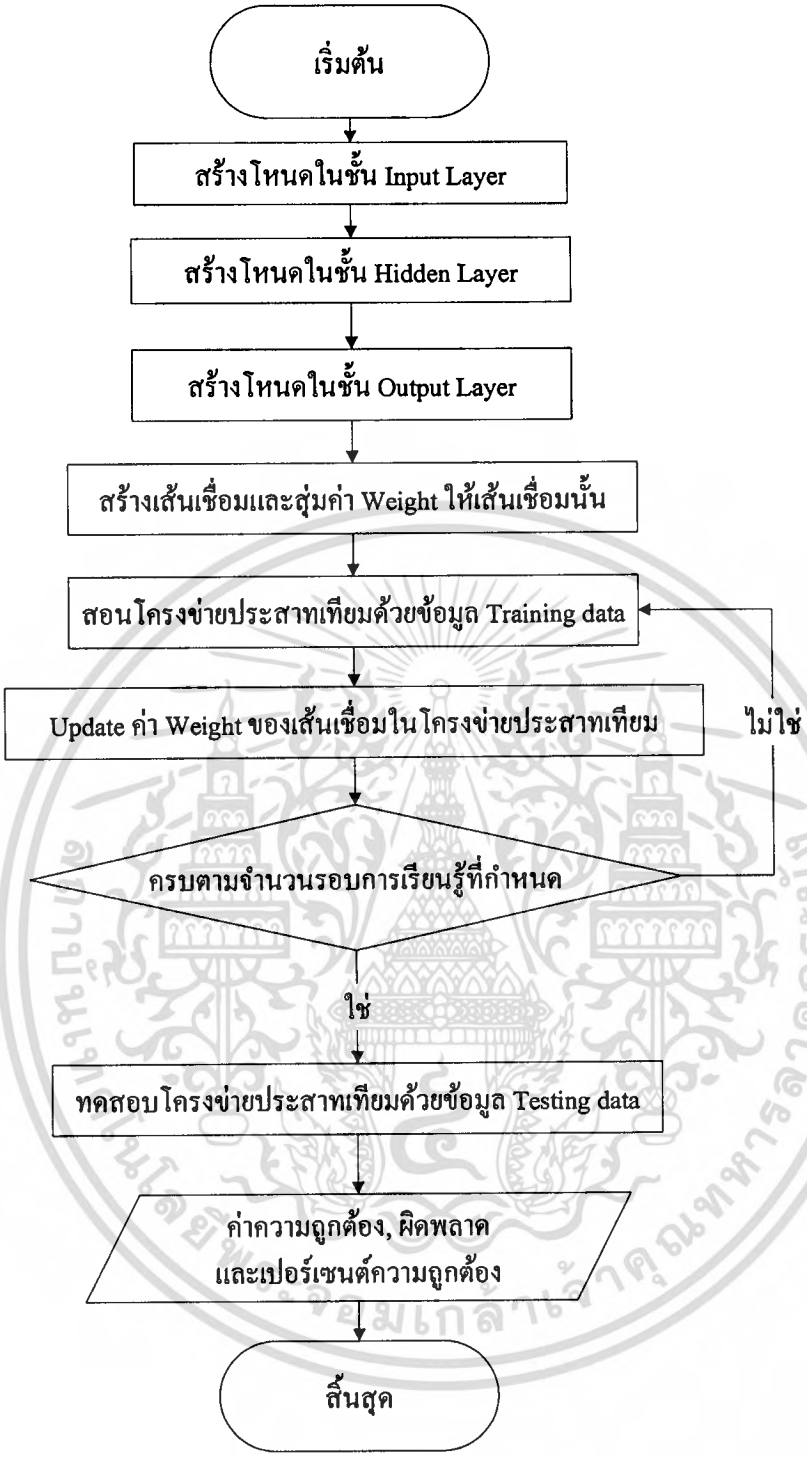
สำหรับการแบ่งกลุ่มข้อมูลนั้นผู้ใช้งานจะเป็นผู้กำหนด Learning rate และจำนวนรอบในการเรียนรู้ให้กับโปรแกรม โดยมีลำดับการทำงาน ดังนี้

1. สร้างโหนดของโครงข่ายประสาทเทียมในชั้น Input layer โดยจะมีจำนวนโหนดเท่ากับจำนวนแอททริบิวต์ของข้อมูลที่ใช้ในการเรียนรู้และทดลอง
2. สร้างโหนดของโครงข่ายประสาทเทียมในชั้น Hidden layer โดยจะมีจำนวนโหนดเท่ากับจำนวนที่ผู้ใช้งานกำหนดไว้
3. สร้างโหนดของโครงข่ายประสาทเทียมในชั้น Output layer โดยกำหนดให้มีจำนวนโหนดเท่ากับ 1 โหนด
4. สร้างเส้นเชื่อมระหว่างโหนดในแต่ละชั้นและสุ่มค่า Weight ให้กับเส้นเชื่อมนั้น โดยทุกๆ โหนดในชั้น Input layer จะสร้างเส้นเชื่อมไปยังทุกๆ โหนดในชั้น Hidden Layer และทุกๆ โหนดในชั้น Hidden layer จะสร้างเส้นเชื่มไปยังโหนดในชั้น Output layer
5. สอนโครงข่ายประสาทเทียมด้วยข้อมูล Training ที่ผู้ใช้งานกำหนดไว้ โดยจะมีการเปลี่ยนแปลงค่าของข้อมูลให้เป็นตัวเลขอยู่ในช่วง 0 ถึง 1 โดยถ้าข้อมูลของแอททริบิวต์เป็นประเภท Numerical จะมีการคำนวณค่าใหม่โดยใช้สูตรดังนี้

$$\text{New value} = \frac{\text{Current value - Min value}}{\text{Max value - Min value}} \qquad (3.1)$$

สำหรับกรณีที่ข้อมูลของแอททริบิวต์เป็นประเภท Categorical นั้นจะใช้วิธีการจัดข้อมูลออกเป็นกลุ่มที่เหมือนกัน และกำหนดค่าให้ข้อมูลแต่ละกลุ่มโดยการแบ่งช่วงกลุ่มแต่ละกลุ่มให้มีค่าระยะห่างเท่าๆกัน ในช่วง 0 ถึง 1

8. อัพเดตค่า Weight ของเส้นเชื่อมในโครงข่ายประสาทเทียม โดยขั้นตอนการอัพเดตค่านั้นจะเป็นไปตามที่ได้อธิบายไว้ในบทที่ 2 ในส่วนของ Backpropagation Network
6. ถ้าทำงานยังไม่ครบตามจำนวนรอบการเรียนรู้ที่กำหนดไว้ ให้กลับไปทำใหม่ตั้งแต่ข้อ 4
7. ทดลองโครงข่ายประสาทเทียมด้วยข้อมูล Testing ที่ผู้ใช้งานกำหนดไว้ โดยจะมีการเปลี่ยนแปลงค่าของข้อมูลให้เป็นตัวเลขอยู่ในช่วง 0 ถึง 1 เหมือนกับขั้นตอนการสอนโครงข่ายประสาทเทียม
8. แสดงค่าความถูกต้องและผิดพลาดของการทดลอง รวมทั้งเปอร์เซนต์ความถูกต้อง

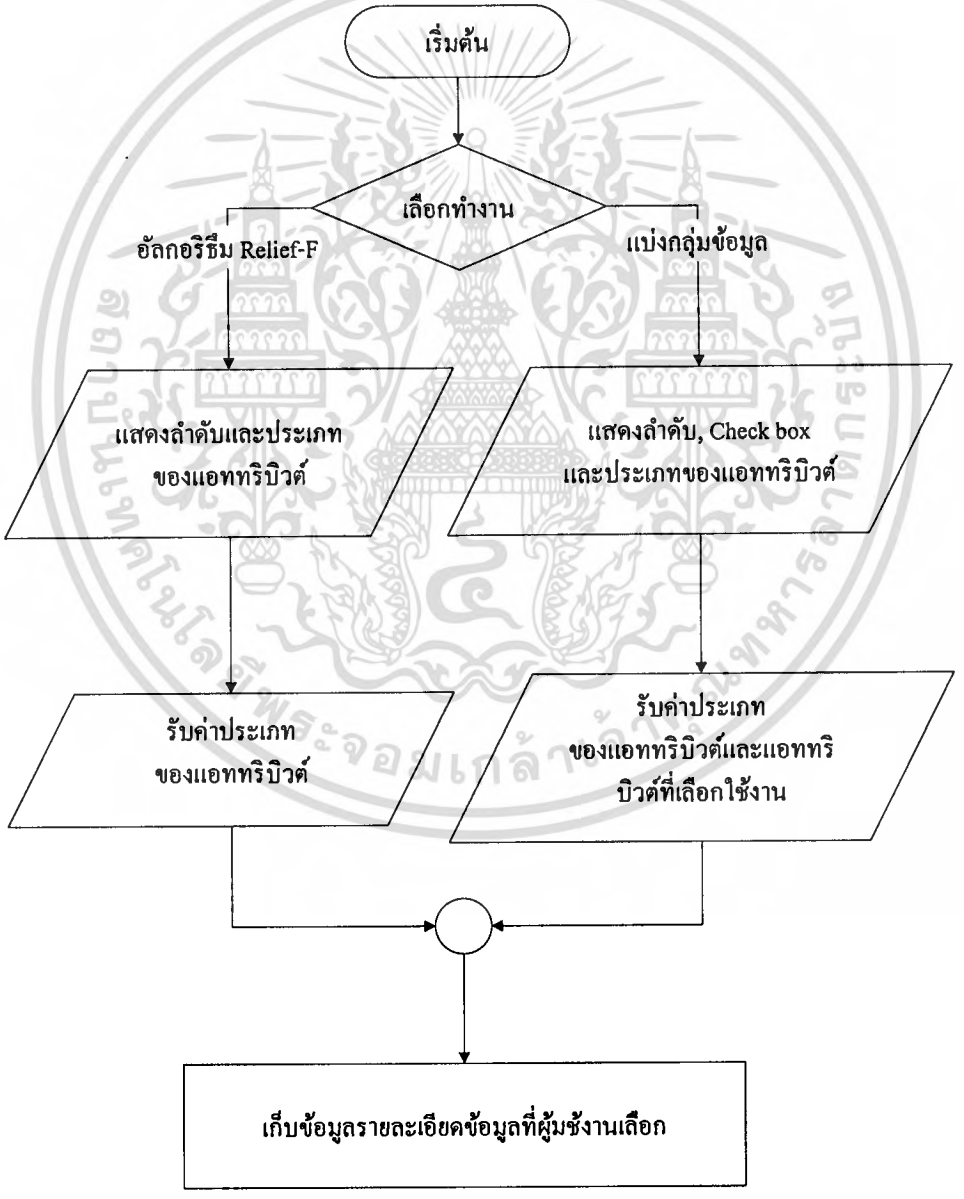


รูปที่ 3.5 Flowchart ของการแบ่งกลุ่มข้อมูลด้วยโครงข่ายประสาทเทียม

3.3 ขั้นตอนการ Verify ข้อมูล

สำหรับการ Verify ข้อมูลนั้นผู้ใช้งานจะเป็นสามารถใช้งานได้เฉพาะกรณีที่ผู้ใช้งานเลือกการทำ Feature Selection โดยใช้อัลกอริธึม Relief-F หรือเลือกการแบ่งกลุ่มข้อมูลโดยใช้โครงข่ายประสาทเทียม โดยมีลำดับการทำงาน ดังนี้

1. กรณีที่ผู้ใช้งานเลือกทำ Feature Selection โดยใช้อัลกอริธึม Relief-F ระบบจะแสดงลำดับและประเภทของแอททริบิวต์ โดยเริ่มต้นโปรแกรมจะกำหนดค่าตั้งต้นของประเภทของแอททริบิวต์เป็น Categorical กรณีที่ผู้ใช้งานเลือกการแบ่งกลุ่มข้อมูลโดยใช้โครงข่ายประสาทเทียม ระบบจะแสดงลำดับ, ประเภทของแอททริบิวต์และ Check box ในการระบุแอททริบิวต์ที่จะเลือกใช้ทำงานโดยเริ่มต้นโปรแกรมจะกำหนดค่าตั้งต้นของประเภทของแอททริบิวต์เป็น Categorical และ Check box มีค่าเป็นจริงซึ่งหมายถึงเลือกใช้แอททริบิวต์ในการทำงาน
2. ผู้ใช้งานเลือกประเภทหรือระบุแอททริบิวต์ที่ต้องการใช้งาน
3. โปรแกรมเก็บรายละเอียดของข้อมูลตามที่ผู้ใช้งานเลือกไว้

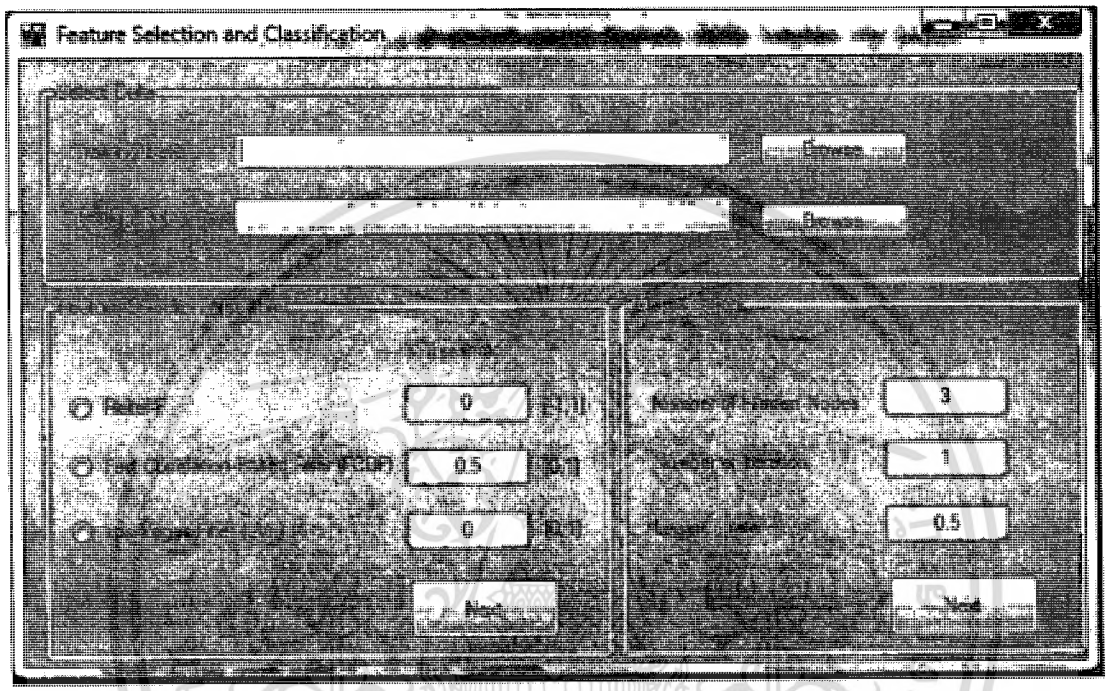


รูปที่ 3.6 Flowchart ของการ Verify ข้อมูล

3.4 หน้าจออินเตอร์เฟสของโปรแกรมประยุกต์

สำหรับหน้าจออินเตอร์เฟส (Interface) ของโปรแกรมประยุกต์นั้นจะแยกออกเป็น 4 อินเตอร์เฟสด้วยกัน โดยมีรายละเอียดดังนี้

3.4.1 หน้าจออินเตอร์เฟสหลัก



รูปที่ 3.7 หน้าจออินเตอร์เฟสหลัก

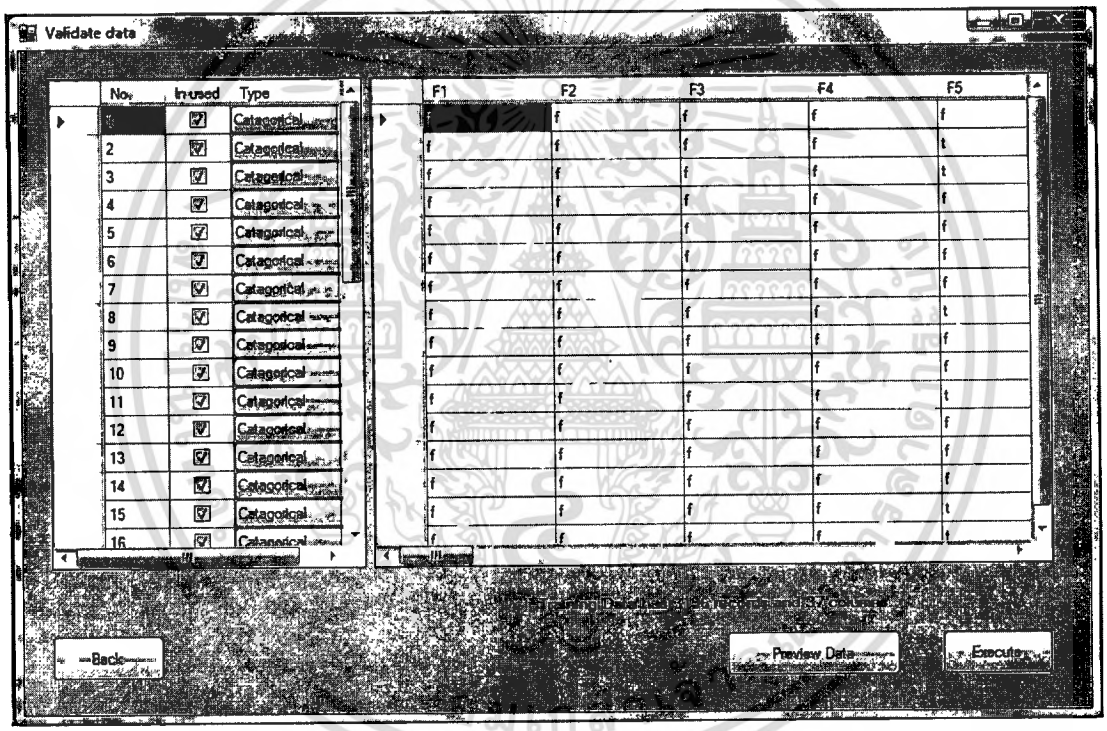
ในส่วนการทำงานของหน้าจออินเตอร์เฟสนี้จะประกอบด้วยส่วนการทำงาน 3 ส่วน ด้วยกัน คือ

- ส่วนที่ 1 ที่ทำหน้าที่ในการรับไฟล์ข้อมูล Training และข้อมูล Testing ซึ่งผู้ใช้งานสามารถเลือกไฟล์ข้อมูลที่ต้องการได้โดยการกดปุ่ม "Browse" โดยไฟล์ที่โปรแกรมรองรับนั้นจะมีการกำหนดประเภทไว้เป็นไฟล์ Text (*.txt) เท่านั้น โดยใช้สัญลักษณ์ "," ในการแยกข้อมูลแต่ละคอลัมภ์ โดยผู้ใช้งานต้องเลือกไฟล์ที่จะใช้เป็นข้อมูล Training และข้อมูล Testing ก่อนที่จะทำงานในขั้นตอนถัดไป
- ส่วนที่ 2 จะใช้ในการกำหนดอัลกอริธึมที่ผู้ใช้งานต้องการใช้ในการทำ Feature Selection โดยมีให้เลือกทั้งหมด 3 อัลกอริธึมด้วยกัน คือ Relief-F, Fast Correlation-Based Filter (FCBF) และ Las Vegas Filter (LVF) ในแต่ละอัลกอริธึมนั้นผู้ใช้งานสามารถที่จะกำหนดค่า Threshold ที่จะใช้ในแต่ละอัลกอริธึมได้ด้วยตนเอง โดยในเบื้องต้นนั้นจะมีค่าตั้งต้นมาให้เป็น 0, 0.5 และ 0 ตามลำดับ ซึ่งค่า Threshold ในแต่ละอัลกอริธึมนั้นจะมีขอบเขตของค่าไม่เท่ากันซึ่งจะมีรายละเอียดแสดงไว้ด้านหลัง Textbox ของแต่ละ

อัลกอริธึม ซึ่งผู้ใช้งานจะต้องระบุค่าที่อยู่ในขอบเขตที่กำหนดไว้ของอัลกอริธึมที่เลือกก่อนที่จะทำงานในขั้นตอนถัดไป โดยผู้ใช้งานสามารถกดปุ่ม "Next" ทางด้านล่างเพื่อเริ่มทำงานในขั้นตอนต่อไปของการทำ Feature Selection

- ส่วนที่ 3 จะใช้ในการกำหนดการสร้างโครงข่ายประสาทเทียมเพื่อใช้ในการแบ่งกลุ่มข้อมูล โดยข้อมูลที่ผู้ใช้งานจะต้องกำหนดค่าให้โปรแกรมประกอบด้วย จำนวนโหนดในชั้น Hidden layer, จำนวนรอบการเรียนรู้ และ Learning rate ของโครงข่ายประสาทเทียม ซึ่งผู้ใช้งานต้องกำหนดค่าให้ครบทุกค่าก่อนที่จะทำงานในขั้นตอนถัดไป โดยผู้ใช้งานสามารถกดปุ่ม "Next" ทางด้านล่างเพื่อเริ่มทำงานในขั้นตอนต่อไปของการแบ่งกลุ่มข้อมูลด้วยโครงข่ายประสาทเทียม

3.4.2 หน้าจออินเตอร์เฟส Verify และแสดงตัวอย่างข้อมูล

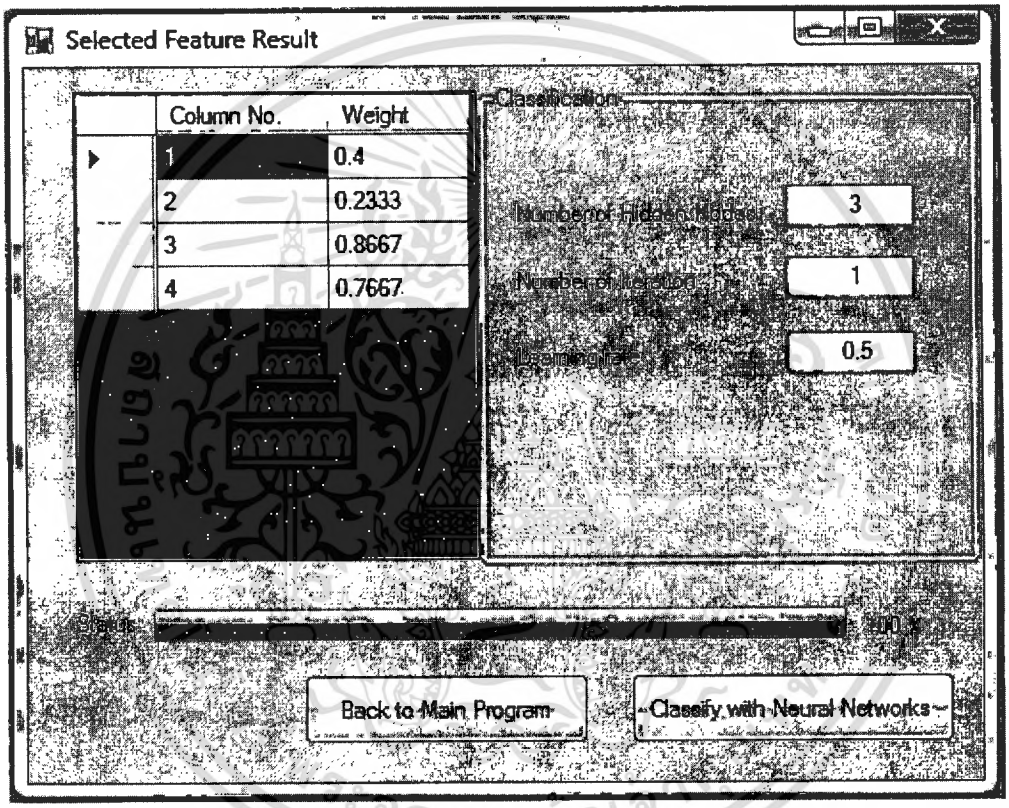


รูปที่ 3.8 หน้าจออินเตอร์เฟส Verify และแสดงตัวอย่างข้อมูล

สำหรับหน้าจออินเตอร์เฟสนี้ จะเป็นหน้าจอการทำงานที่ต่อจากหน้าจออินเตอร์เฟสหลักของโปรแกรม โดยมีหน้าที่เพื่อให้ผู้ใช้งานสามารถตรวจสอบข้อมูลที่จะใช้การทำงานขั้นต่อไปก่อนที่จะเริ่มทำงานจริง โดยโปรแกรมจะแสดงตัวอย่างข้อมูลที่ใช้เป็น Training data ให้ผู้ใช้งานได้เห็น นอกจากนี้ในกรณีที่ผู้ใช้งานเลือกการทำ Feature Selection โดยใช้อัลกอริธึม Relief-F หรือเลือกการแบ่งกลุ่มข้อมูลโดยใช้โครงข่ายประสาทเทียมมาจากหน้าจออินเตอร์เฟสหลักนั้น ผู้ใช้งานสามารถกำหนดประเภทของข้อมูลในแต่ละแอททริบิวต์ได้ ซึ่งมีค่าให้เลือกระหว่าง Categorical เมื่อข้อมูลนั้นเป็นข้อมูลเชิงสัญลักษณ์ หรือ Numerical เมื่อข้อมูลนั้นเป็นข้อมูลเชิง

ตัวเลข ซึ่งโปรแกรมจะกำหนดค่า Default ให้ทุกแอททริบิวต์เป็น Categorical นอกจากนี้ในกรณีที่ผู้ใช้งานเลือกการแบ่งกลุ่มข้อมูลโดยใช้โครงข่ายประสาทเทียมแล้ว ผู้ใช้งานสามารถที่จะเลือกแอททริบิวต์ที่จะใช้ในการสร้างและทดลองโครงข่ายประสาทเทียมได้ด้วย โดยโปรแกรมจะกำหนดตั้งต้นให้ทุกแอททริบิวต์นั้นถูกเลือกใช้งานทั้งหมด ในส่วนของด้านล่างหน้าจอนั้นจะประกอบด้วยปุ่ม "Back" เพื่อใช้สำหรับย้อนกลับไปทำหน้าในหน้าจออินเตอร์เฟสหลัก และปุ่ม "Execute" เพื่อเริ่มทำงานในขั้นตอนถัดไปตามที่ผู้ใช้งานได้เลือกทำงานไว้ในหน้าจออินเตอร์เฟสหลัก

3.4.3 หน้าจออินเตอร์เฟสผลการทำ Feature Selection



รูปที่ 3.9 หน้าจออินเตอร์เฟสผลการทำ Feature Selection

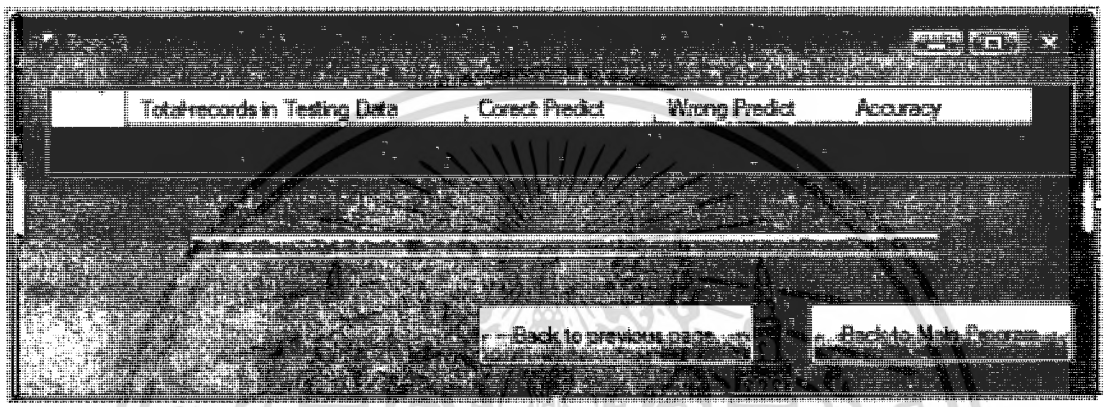
สำหรับหน้าจออินเตอร์เฟสนี้ จะประกอบด้วยส่วนการทำงานหลักๆ 2 ส่วนด้วยกัน คือ

- ส่วนที่ 1 ทางฝั่งซ้ายของโปรแกรมจะแสดงผลลัพธ์การทำ Feature Selection ของโปรแกรม ซึ่งจะแสดงลำดับของแอททริบิวต์ที่ผ่านการประเมินและค่า Weight ของแอททริบิวต์นั้นให้ผู้ใช้งานเห็นเมื่อจบการทำงาน โดยในกรณีที่ไม่มีแอททริบิวต์ใดเลยที่ผ่านการประเมิน โปรแกรมจะไม่แสดงข้อมูลใดๆให้ผู้ใช้งานเห็น
- ส่วนที่ 2 ทางฝั่งขวาของโปรแกรมจะเหมือนหน้าจออินเตอร์เฟสหลักของโปรแกรมในส่วนที่ 3 ที่ให้ผู้ใช้งานกำหนดค่าในการสร้างโครงข่ายประสาทเทียมเพื่อใช้ในการ

แบ่งกลุ่มข้อมูลในกรณีที่ผู้ใช้งานต้องการนำเอาแอททริบิวต์ที่ผ่านการประเมินนี้ไปใช้ในการแบ่งกลุ่มข้อมูลด้วยโครงข่ายประสาทเทียมต่อไป

สำหรับด้านล่างของหน้าจออินเตอร์เฟสจะประกอบด้วยแถบสถานะการทำงานที่เริ่มต้นที่ 0% และจะสิ้นสุดการทำงานเมื่อมีค่าเท่ากับ 100%, ปุ่ม "Back to Main Program" เพื่อใช้สำหรับกลับไปทำงานยังหน้าจออินเตอร์เฟสหลัก และปุ่ม "Classify with Neural Networks" เพื่อใช้สำหรับแบ่งกลุ่มข้อมูลด้วยโครงข่ายประสาทเทียมต่อไป

3.4.4 หน้าจออินเตอร์เฟสผลการแบ่งกลุ่มข้อมูลด้วยโครงข่ายประสาทเทียม



รูปที่ 3.10 หน้าจออินเตอร์เฟสผลการแบ่งกลุ่มข้อมูลด้วยโครงข่ายประสาทเทียม

สำหรับหน้าจออินเตอร์เฟสนี้จะแสดงผลลัพธ์ของการทดลองการแบ่งกลุ่มข้อมูล Testing โดยใช้โครงข่ายประสาทเทียมที่สร้างขึ้นโดยอาศัยข้อมูล Training ที่ผู้ใช้งานกำหนด ซึ่งผลลัพธ์การทำงานที่แสดงให้ผู้ใช้งานทราบจะประกอบด้วย จำนวนแถวของข้อมูลในไฟล์ข้อมูล Testing ทั้งหมด, จำนวนแถวข้อมูลที่โครงข่ายประสาทเทียมสามารถแบ่งกลุ่มได้อย่างถูกต้อง, จำนวนแถวข้อมูลที่โครงข่ายประสาทเทียมไม่สามารถแบ่งกลุ่มได้อย่างถูกต้อง และเปอร์เซนต์ความถูกต้องในการทำงานของโครงข่ายประสาทเทียม สำหรับด้านล่างของหน้าจออินเตอร์เฟสจะประกอบด้วยแถบสถานะการทำงานที่เริ่มต้นที่ 0% และจะสิ้นสุดการทำงานเมื่อมีค่าเท่ากับ 100%, ปุ่ม "Back to previous page" เพื่อกลับไปทำงานในหน้าจอ Verify และแสดงตัวอย่างข้อมูล และปุ่ม "Back to Main Program" เพื่อใช้สำหรับกลับไปทำงานยังหน้าจออินเตอร์เฟสหลัก

บทที่ 4

การทดลอง

4.1 จุดมุ่งหมายในการทดลอง

สำหรับจุดมุ่งหมายในการทดลอง คือ ต้องการเปรียบเทียบผลลัพธ์ที่ได้ของการแบ่งกลุ่มข้อมูลด้วยโครงข่ายประสาทเทียมด้วยวิธีทั่วไปและการใช้อัลกอริมต่างๆในการทำ Feature Selection ก่อนทำการแบ่งกลุ่มข้อมูล โดยเปรียบเทียบในเรื่องของความถูกต้องของการแบ่งกลุ่มข้อมูลด้วยวิธีต่างๆข้างต้นว่าวิธีใดให้ผลลัพธ์ที่ถูกต้องและผิดพลาดมากน้อยเพียงใด และในขณะเดียวกันก็ต้องการเปรียบเทียบผลลัพธ์ในการแบ่งกลุ่มข้อมูลที่ได้จากการทำ Feature Selection ในแต่ละอัลกอริธึมว่าให้ผลลัพธ์ที่แตกต่างกันหรือไม่อย่างไร

4.2 องค์ประกอบในการทดลอง

4.2.1 โปรแกรมที่ใช้ในการทดลอง

ในการทดลองประสิทธิภาพอัลกอริธึมในการทำ Feature Selection นั้น ผู้จัดทำรายงานได้ใช้โปรแกรมประยุกต์ที่ผู้จัดทำรายงานได้พัฒนาขึ้นเองโดยภาษา VB.NET ในการ Feature Selection ตามขั้นตอนในอัลกอริธึม Relief-F, Fast Correlation-Based Filter (FCBF) และ Las Vegas Filter (LVF) และใช้ในการแบ่งกลุ่มข้อมูลด้วยโครงข่ายประสาทเทียมแบบ Backpropagation

4.2.2 เครื่องคอมพิวเตอร์ที่ใช้ในทดลอง

สำหรับเครื่องคอมพิวเตอร์ที่ใช้มีรายละเอียดเครื่องดังนี้

- หน่วยประมวลผลกลาง (CPU) : Intel Core 2 Duo P8400 2.26 GHz.
- พื้นที่หน่วยความจำสำรอง (RAM) : 4 GB.
- พื้นที่ฮาร์ดดิสก์ : SATA 250 GB.
- ระบบปฏิบัติการ : Microsoft Windows XP service pack 3

4.2.3 ข้อมูลที่ใช้ในการทดลอง

สำหรับข้อมูลที่ใช้ในการทดลองนั้น ผู้จัดทำรายงานได้ใช้ข้อมูลจำนวนหนึ่งจากเว็บไซต์ http://archive.ics.uci.edu/ml/ ในการทดลอง โดยเลือกใช้เฉพาะข้อมูลสำหรับการแบ่งกลุ่มข้อมูลที่มีประเภทเป็น Categorical เท่านั้น ซึ่งมีขนาดของจำนวนแถวข้อมูลและแอททริบิวต์ที่แตกต่าง

กัน โดยข้อมูลแต่ละชนิดนั้นจะทำการแยกออกเป็นข้อมูล Training และTesting โดยข้อมูลต่างๆที่ใช้ในการทดลองมีรายละเอียดดังนี้

- Hayes-Roth

1	2,1,1,2,1
2	2,1,3,2,2
3	3,1,4,1,3
4	2,4,2,2,3
5	1,1,3,4,3
6	1,1,3,2,2
7	3,1,3,2,2
8	3,4,2,4,3
9	2,2,1,1,1
10	3,2,1,1,1
11	1,2,1,1,1
12	2,2,3,4,3
13	1,1,2,1,1
14	2,1,2,2,2
15	2,4,1,4,3
16	1,1,3,3,1

รูปที่ 4.1 ตัวอย่างข้อมูล Hayes-Roth

- จำนวนแถวข้อมูล Training : 132
- จำนวนแถวข้อมูล Testing : 28
- จำนวนแอททริบิวต์ : 5 ประกอบด้วย Hobby, Age, Educational, Marital status และ Class

- Balance Scale

1	1,1,1,1,B
2	1,1,1,2,R
3	1,1,1,3,R
4	1,1,1,4,R
5	1,1,1,5,R
6	1,1,2,1,R
7	1,1,2,2,R
8	1,1,2,3,R
9	1,1,2,4,R
10	1,1,2,5,R
11	1,1,3,1,R
12	1,1,3,2,R
13	1,1,3,3,R
14	1,1,3,4,R
15	1,1,3,5,R
16	1,1,4,1,R
17	1,1,4,2,R

รูปที่ 4.2 ตัวอย่างข้อมูล Balance Scale

- จำนวนแถวข้อมูล Training : 625
- จำนวนแถวข้อมูล Testing : 625
- จำนวนแอททริบิวต์ : 5 ประกอบด้วย Left-Weight:, Left-Distance, Right-Weight:, Right-Distance และ Class

- Car Evolution

```
    1  vhigh,vhigh,2,2,small,low,unacc
    2  vhigh,vhigh,2,2,small,med,unacc
    3  vhigh,vhigh,2,2,small,high,unacc
    4  vhigh,vhigh,2,2,med,low,unacc
    5  vhigh,vhigh,2,2,med,med,unacc
    6  vhigh,vhigh,2,2,med,high,unacc
    7  vhigh,vhigh,2,2,big,low,unacc
    8  vhigh,vhigh,2,2,big,med,unacc
    9  vhigh,vhigh,2,2,big,high,unacc
   10  vhigh,vhigh,2,4,small,low,unacc
   11  vhigh,vhigh,2,4,small,med,unacc
   12  vhigh,vhigh,2,4,small,high,unacc
   13  vhigh,vhigh,2,4,med,low,unacc
   14  vhigh,vhigh,2,4,med,med,unacc
   15  vhigh,vhigh,2,4,med,high,unacc
   16  vhigh,vhigh,2,4,big,low,unacc
   17  vhigh,vhigh,2,4,big,med,unacc
```

รูปที่ 4.3 ตัวอย่างข้อมูล Car Evolution

- จำนวนแถวข้อมูล Training : 1,728
- จำนวนแถวข้อมูล Testing : 1,728
- จำนวนแอททริบิวต์ : 7 ประกอบด้วย Buying, Maint, Doors, Persons, Lug_boot, Safety และ Class

- Chess

```
    1  f,f,f,f,f,f,f,f,f,f,f,f,f,l,f,n,f,f,t,f,f,f,f,f,f,f,t,f,f,f,f,f,f,f,t,t,n,won
    2  f,f,f,f,t,f,f,f,f,f,f,f,l,f,n,f,f,t,f,f,f,f,f,f,f,t,f,f,f,f,f,f,f,t,t,n,won
    3  f,f,f,f,t,f,t,f,f,f,f,f,l,f,n,f,f,t,f,f,f,f,f,f,f,t,f,f,f,f,f,f,f,t,t,n,won
    4  f,f,f,f,f,f,f,f,t,f,f,f,l,f,n,f,f,t,f,f,f,t,f,f,f,t,f,f,f,f,f,f,f,t,t,n,won
    5  f,f,f,f,f,f,f,f,f,f,f,f,l,f,n,f,f,t,f,f,f,f,f,t,t,f,f,f,f,f,f,f,t,t,n,won
    6  f,f,f,f,f,f,f,f,f,f,f,f,l,f,n,f,f,t,f,f,f,f,f,t,t,f,f,f,t,f,f,f,t,t,n,won
    7  f,f,f,f,f,f,f,f,t,f,f,f,l,f,n,f,f,t,f,f,f,t,f,t,f,t,f,f,f,f,f,f,t,t,n,won
    8  f,f,f,f,t,f,f,f,f,f,t,f,l,f,n,f,f,t,f,f,f,f,f,f,f,t,f,f,f,f,f,f,f,t,t,n,won
    9  f,f,f,f,f,f,f,f,f,f,t,f,l,f,n,f,f,t,f,f,f,f,f,f,f,t,f,f,f,f,t,f,f,t,t,n,won
   10  f,f,f,f,f,f,t,f,f,f,t,f,l,f,n,f,f,t,f,f,f,f,f,f,f,t,f,f,f,f,f,f,f,t,t,n,won
   11  f,f,f,f,t,f,t,f,f,f,t,f,l,f,n,f,f,t,f,f,f,f,f,f,f,t,f,f,f,f,f,f,f,t,t,n,won
   12  f,f,f,f,f,f,f,f,t,f,t,f,l,f,n,f,f,t,f,f,f,t,f,f,f,t,f,f,f,f,f,f,f,t,t,n,won
   13  f,f,f,f,f,f,f,f,t,f,t,f,l,f,n,f,f,t,f,f,f,t,f,f,f,t,f,f,f,f,t,f,f,t,t,n,won
   14  f,f,f,f,f,f,f,f,t,f,t,f,l,f,n,f,f,t,f,t,f,t,f,f,f,t,f,f,f,f,t,f,f,t,t,n,won
   15  f,f,f,f,t,f,f,f,t,f,t,f,l,f,n,f,f,t,f,f,f,t,f,f,f,t,f,f,f,f,t,f,f,t,t,n,won
   16  f,f,f,f,t,f,f,f,f,f,t,f,l,f,n,f,f,t,f,f,f,f,f,t,t,f,f,f,f,f,f,f,t,t,n,won
   17  f,f,f,f,f,f,f,f,f,f,t,f,l,f,n,f,f,t,f,f,f,f,f,t,t,f,f,f,f,f,t,f,f,t,t,n,won
```

รูปที่ 4.4 ตัวอย่างข้อมูล Chess

- จำนวนแถวข้อมูล Training : 3,196
- จำนวนแถวข้อมูล Testing : 3,196
- จำนวนแอททริบิวต์ : 37 ประกอบด้วย Board_position จำนวน 36 แอททริบิวต์ และ Class

4.3 ปัจจัยที่เกี่ยวข้องกับการทดลอง

สำหรับปัจจัยที่เกี่ยวข้องกับการทดลองนั้นคือขนาดของข้อมูลที่ใช้ในการทดลอง โดยเลือกใช้ข้อมูลที่มีขนาดแตกต่างกันในเรื่องของจำนวนแถวข้อมูลและจำนวนแอททริบิวต์ในแต่ละครั้งที่ทดลอง

4.4 การออกแบบการทดลอง

1. ทำการทดลองโดยใช้ข้อมูลที่มีขนาดแตกต่างกัน คือ ข้อมูลที่มีจำนวนของแอททริบิวต์แตกต่างกันและข้อมูลที่มีขนาดของแถวข้อมูลแตกต่างกัน
2. เริ่มต้นทดลองโดยการแบ่งกลุ่มข้อมูลด้วยโปรแกรมประยุกต์โดยไม่มีการทำ Feature Selection ใดๆ เพื่อหาค่าของจำนวนโหนดในชั้น Hidden layer, จำนวนรอบการเรียนรู้ และ Learning rate ที่เหมาะสมในแต่ละประเภทข้อมูล หลังจากนั้นจึงทำการทดลองใหม่อีก 10 ครั้งเพื่อหาค่าเฉลี่ยและบันทึกผล
3. ทำการทดลองใหม่อีกครั้งโดยครั้งนี้ให้เลือกทำ Feature Selection ก่อน โดยเลือกอัลกอริธึม Relief-F, Fast Correlation-Based Filter (FCBF) และ Las Vegas Filter (LVF) ตามลำดับ โดยใช้ค่า Threshold ของแต่ละอัลกอริธึมในช่วงที่แตกต่างกันออกไป หลังจากนั้นจึงทำการแบ่งกลุ่มข้อมูลโดยใช้ค่าตัวแปรต่างๆเหมือนในขั้นตอนที่ 2 โดยทำการแบ่งกลุ่มข้อมูลที่ผ่านการทำ Feature Selection ในแต่ละอัลกอริธึมอย่างละ 10 ครั้ง เพื่อหาค่าเฉลี่ยและบันทึกผล
4. เปลี่ยนข้อมูลที่ใช้ในการทดสอบใหม่ และเริ่มทดสอบใหม่อีกครั้งตามขั้นตอนที่ 2 และ 3
5. นำผลลัพธ์ที่ได้มาเปรียบเทียบกันและสรุปผล

4.5 ผลการทดลอง

4.5.1. ผลการทดลองแบ่งกลุ่มข้อมูลด้วยโครงข่ายประสาทเทียมอย่างเดียว

สำหรับการแบ่งกลุ่มข้อมูลด้วยโครงข่ายประสาทเทียมนั้น ได้กำหนดให้โครงข่ายประสาทเทียมมีจำนวนโหนดในชั้น Hidden layer, Learning rate และจำนวนรอบการเรียนรู้ โดยมีรายละเอียดและผลการทดลองดังนี้

ตารางที่ 4.1 ผลการแบ่งกลุ่มข้อมูล Hayes-Roth ด้วยโครงข่ายประสาทเทียมอย่างเดียว

Hayes-Roth					
Hidden nodes = 4, Learning rate = 0.1, จำนวนรอบ = 500					
ครั้งที่	1	2	3	4	5
จำนวนแถวที่แบ่งกลุ่มถูกต้อง	17	16	18	15	17
จำนวนแถวที่แบ่งกลุ่มผิดพลาด	11	12	10	13	11
เปอร์เซนต์ความถูกต้อง	60.71%	57.14%	64.29%	53.57%	60.71%
ครั้งที่	6	7	8	9	10
จำนวนแถวที่แบ่งกลุ่มถูกต้อง	15	18	17	15	16
จำนวนแถวที่แบ่งกลุ่มผิดพลาด	13	10	11	13	12
เปอร์เซนต์ความถูกต้อง	53.57%	64.29%	60.71%	53.57%	57.14%

ตารางที่ 4.2 ผลการแบ่งกลุ่มข้อมูล Balance Scale ด้วยโครงข่ายประสาทเทียมอย่างเดียว

Balance Scale					
Hidden nodes = 4, Learning rate = 0.1, จำนวนรอบ = 500					
ครั้งที่	1	2	3	4	5
จำนวนแถวที่แบ่งกลุ่มถูกต้อง	552	552	550	550	550
จำนวนแถวที่แบ่งกลุ่มผิดพลาด	73	73	75	75	75
เปอร์เซนต์ความถูกต้อง	88.32%	88.32%	88.00%	88.00%	88.00%
ครั้งที่	6	7	8	9	10
จำนวนแถวที่แบ่งกลุ่มถูกต้อง	553	552	551	552	551
จำนวนแถวที่แบ่งกลุ่มผิดพลาด	72	73	74	73	74
เปอร์เซนต์ความถูกต้อง	88.48%	88.32%	88.16%	88.32%	88.16%

ตารางที่ 4.3 ผลการแบ่งกลุ่มข้อมูล Car Evo ด้วยโครงข่ายประสาทเทียมอย่างเดียว

Car Evo					
Hidden nodes = 7, Learning rate = 0.2, จำนวนรอบ = 500					
ครั้งที่	1	2	3	4	5
จำนวนเคสที่แบ่งกลุ่มถูกต้อง	1,517	1,518	1,407	1,405	1,537
จำนวนเคสที่แบ่งกลุ่มผิดพลาด	211	210	321	323	191
เปอร์เซ็นต์ความถูกต้อง	87.79%	87.85%	81.42%	81.31%	88.95%
ครั้งที่	6	7	8	9	10
จำนวนเคสที่แบ่งกลุ่มถูกต้อง	1,487	1,476	1,521	1,512	1,403
จำนวนเคสที่แบ่งกลุ่มผิดพลาด	241	252	207	216	325
เปอร์เซ็นต์ความถูกต้อง	86.05%	85.42%	88.02%	87.50%	81.19%

ตารางที่ 4.4 ผลการแบ่งกลุ่มข้อมูล Chess ด้วยโครงข่ายประสาทเทียมอย่างเดียว

Chess					
Hidden nodes = 4, Learning rate = 0.2, จำนวนรอบ = 200					
ครั้งที่	1	2	3	4	5
จำนวนเคสที่แบ่งกลุ่มถูกต้อง	1,899	1,865	1,879	1,894	1,865
จำนวนเคสที่แบ่งกลุ่มผิดพลาด	1,297	1,331	1,317	1,302	1,331
เปอร์เซ็นต์ความถูกต้อง	59.42%	58.35%	58.79%	59.26%	58.35%
ครั้งที่	6	7	8	9	10
จำนวนเคสที่แบ่งกลุ่มถูกต้อง	1,899	1,867	1,877	1,901	1,892
จำนวนเคสที่แบ่งกลุ่มผิดพลาด	1,297	1,329	1,319	1,295	1,304
เปอร์เซ็นต์ความถูกต้อง	59.42%	58.42%	58.73%	59.48%	59.20%

4.5.2. ผลการทดลองทำ Feature Selection ด้วย Relief-F และแบ่งกลุ่มข้อมูลด้วยโครงข่ายประสาทเทียม

ในการทำ Feature Selection ข้อมูลต่างๆด้วย Relief-F ที่ค่า Threshold ในช่วงต่างๆ จำนวน 10 ครั้ง ได้ผลการทดลองดังนี้

ตารางที่ 4.5 ผลการทำ Feature Selection ข้อมูล Hayes-Roth ด้วย Relief-F

ค่า Threshold	-1	-0.8	-0.6	-0.4	-0.2	
ลำดับแอตทริบิวต์ที่ถูกเลือก	ทั้งหมด	ทั้งหมด	ทั้งหมด	ทั้งหมด	ทั้งหมด	
ค่า Threshold	0	0.2	0.4	0.6	0.8	1
ลำดับแอตทริบิวต์ที่ถูกเลือก	(3), (4), (2,3), (2,4), (2,3,4)	(2), (3), (4)	-	-	-	-

ตารางที่ 4.6 ผลการทำ Feature Selection ข้อมูล Balance Scale ด้วย Relief-F

ค่า Threshold	-1	-0.8	-0.6	-0.4	-0.2	
ลำดับแอตทริบิวต์ที่ถูกเลือก	ทั้งหมด	ทั้งหมด	ทั้งหมด	ทั้งหมด	ทั้งหมด	
ค่า Threshold	0	0.2	0.4	0.6	0.8	1
ลำดับแอตทริบิวต์ที่ถูกเลือก	-	-	-	-	-	-

ตารางที่ 4.7 ผลการทำ Feature Selection ข้อมูล Car Evo ด้วย Relief-F

ค่า Threshold	-1	-0.8	-0.6	-0.4	-0.2	
ลำดับแอตทริบิวต์ที่ถูกเลือก	ทั้งหมด	ทั้งหมด	ทั้งหมด	ทั้งหมด	ทั้งหมด	
ค่า Threshold	0	0.2	0.4	0.6	0.8	1
ลำดับแอตทริบิวต์ที่ถูกเลือก	(1,2,4,5,6)	(4,6)	-, (6)	-	-	-

ตารางที่ 4.8 ผลการทำ Feature Selection ข้อมูล Chess ด้วย Relief-F

Chess						
ค่า Threshold	-1	-0.8	-0.6	-0.4	-0.2	
ลำดับแอททริบิวต์ที่ถูกเลือก	ทั้งหมด	ทั้งหมด	ทั้งหมด	ทั้งหมด	ทั้งหมด	
ค่า Threshold	0	0.2	0.4	0.6	0.8	1
ลำดับแอททริบิวต์ที่ถูกเลือก	(21,29), (14,21,29)	-, (21)	-	-	-	-

จากนั้นทำการแบ่งกลุ่มข้อมูลด้วยโครงข่ายประสาทเทียม โดยใช้แอททริบิวต์ที่ได้จากการทำ Feature Selection ด้วย Relief-F ที่ค่า Threshold ในช่วงต่างๆ ซึ่งนำมาเฉพาะช่วงที่มีบางแอททริบิวต์ถูกเลือกเท่านั้น โดยใช้จำนวนโหนดในชั้น Hidden layer, Learning rate และจำนวนรอบการเรียนรู้เหมือนกับการทดลองที่แบ่งกลุ่มข้อมูลด้วยโครงข่ายประสาทเทียมอย่างเดียว สำหรับค่า Threshold ที่ไม่มีแอททริบิวต์ใดถูกเลือกเลยหรือถูกเลือกทั้งหมดจะไม่นำมาทดลองแบ่งกลุ่มข้อมูลใหม่ เนื่องจากถือว่าได้ผลการทดลองไม่แตกต่างจากการทดลองแบ่งกลุ่มข้อมูลด้วยโครงข่ายประสาทเทียมอย่างเดียว โดยมีรายละเอียดดังนี้

ตารางที่ 4.9 ผลการทำ Feature Selection ข้อมูล Hayes-Roth ด้วย Relief-F และแบ่งกลุ่มข้อมูลด้วยโครงข่ายประสาทเทียม

Hayes-Roth			
Hidden nodes = 4, Learning rate = 0.1, จำนวนรอบ = 500			
สำหรับแอททริบิวต์ที่ใช้	2	4	2,4
จำนวนตัวอย่างที่แบ่งกลุ่มถูกต้อง	13	13	13
จำนวนตัวอย่างที่แบ่งกลุ่มผิดพลาด	15	15	15
เปอร์เซ็นต์ความถูกต้องเฉลี่ย	46.43%	46.43%	46.43%
สำหรับแอททริบิวต์ที่ใช้	2,3	2,4	2,3,4
จำนวนตัวอย่างที่แบ่งกลุ่มถูกต้อง	16	13	15
จำนวนตัวอย่างที่แบ่งกลุ่มผิดพลาด	12	15	13
เปอร์เซ็นต์ความถูกต้องเฉลี่ย	57.14%	46.43%	55.56%

ตารางที่ 4.10 ผลการทำ Feature Selection ข้อมูล Car Evo ด้วย Relief-F และแบ่งกลุ่มข้อมูล ด้วยโครงข่ายประสาทเทียม

Car Evo			
Hidden nodes = 7, Learning rate = 0.2, จำนวนรอบ = 500			
สำหรับแอทริบิวต์ที่ใช้	6	4,6	1,2,4,5,6
จำนวนแถวเฉลี่ยที่แบ่งกลุ่มถูกต้อง	285	1,196	1,476
จำนวนแถวเฉลี่ยที่แบ่งกลุ่มผิดพลาด	1,443	532	252
เปอร์เซนต์ความถูกต้องเฉลี่ย	16.49%	69.21%	85.42%

ตารางที่ 4.11 ผลการทำ Feature Selection ข้อมูล Chess ด้วย Relief-F และแบ่งกลุ่มข้อมูล ด้วยโครงข่ายประสาทเทียม

Chess			
Hidden nodes = 4, Learning rate = 0.2, จำนวนรอบ = 500			
สำหรับแอทริบิวต์ที่ใช้	21	21,29	4,21,29
จำนวนแถวเฉลี่ยที่แบ่งกลุ่มถูกต้อง	1,527	1,527	1,527
จำนวนแถวเฉลี่ยที่แบ่งกลุ่มผิดพลาด	1,669	1,669	1,669
เปอร์เซนต์ความถูกต้องเฉลี่ย	47.78%	47.78%	47.78%

สำหรับข้อมูล Balance Scale เนื่องจากไม่มีเซตย่อยของแอททริบิวต์ใดเลยที่ถูกเลือก จึงถือว่าได้ผลการทดลองไม่แตกต่างจากการทดลองแบ่งกลุ่มข้อมูลด้วยโครงข่ายประสาทเทียมอย่างเดียว

4.5.3. ผลการทดลองทำ Feature Selection ด้วย Fast Correlation-Based Filter และแบ่งกลุ่มข้อมูลด้วยโครงข่ายประสาทเทียม

ในการทำ Feature Selection ข้อมูลต่างๆด้วย Fast Correlation-Based Filter ที่ค่า Threshold ในช่วงต่างๆ จำนวน 10 ครั้ง ได้ผลการทดลองดังนี้

ตารางที่ 4.12 ผลการทำ Feature Selection ข้อมูล Hayes-Roth ด้วย Fast Correlation-Based Filter

Hayes-Roth						
ค่า Threshold	0	0.1	0.2	0.3	0.4	
ตำแหน่งที่การเปลี่ยนแปลงที่เลือก	(2,3,4)	(2,3,4)	-	-	-	
ค่า Threshold	0.5	0.6	0.7	0.8	0.9	1
ตำแหน่งที่การเปลี่ยนแปลงที่เลือก	-	-	-	-	-	-

ตารางที่ 4.13 ผลการทำ Feature Selection ข้อมูล Balance Scale ด้วย Fast Correlation-Based Filter

Balance Scale						
ค่า Threshold	0	0.1	0.2	0.3	0.4	
ตำแหน่งที่การเปลี่ยนแปลงที่เลือก	ทั้งหมด	-	-	-	-	
ค่า Threshold	0.5	0.6	0.7	0.8	0.9	1
ตำแหน่งที่การเปลี่ยนแปลงที่เลือก	-	-	-	-	-	-

ตารางที่ 4.14 ผลการทำ Feature Selection ข้อมูล Car Evo ด้วย Fast Correlation-Based Filter

Car Evo						
ค่า Threshold	0	0.1	0.2	0.3	0.4	
ตำแหน่งที่การเปลี่ยนแปลงที่เลือก	ทั้งหมด	(4,6)	-	-	-	
ค่า Threshold	0.5	0.6	0.7	0.8	0.9	1
ตำแหน่งที่การเปลี่ยนแปลงที่เลือก	-	-	-	-	-	-

ตารางที่ 4.15 ผลการทำ Feature Selection ข้อมูล Chess ด้วย Fast Correlation-Based Filter

Chess						
ค่า Threshold	0	0.1	0.2	0.3	0.4	
ตำแหน่งที่การเปลี่ยนแปลงที่เลือก	(3,10,15,16,21,32,33)	(10,21,33)	(21)	-	-	
ค่า Threshold	0.5	0.6	0.7	0.8	0.9	1
ตำแหน่งที่การเปลี่ยนแปลงที่เลือก	-	-	-	-	-	-

จากนั้นทำการแบ่งกลุ่มข้อมูลด้วยโครงข่ายประสาทเทียม โดยใช้แอททริบิวต์ที่ได้จากการทำ Feature Selection ด้วย Fast Correlation-Based Filter ที่ค่า Threshold ในช่วงต่างๆ ซึ่งนำมาเฉพาะช่วงที่มีบางแอททริบิวต์ถูกเลือกเท่านั้น โดยใช้จำนวนโหนดในชั้น Hidden layer, Learning rate และจำนวนรอบการเรียนรู้เหมือนกับการทดลองที่แบ่งกลุ่มข้อมูลด้วยโครงข่ายประสาทเทียมอย่างเดียว สำหรับค่า Threshold ที่ไม่มีแอททริบิวต์ใดถูกเลือกเลยหรือถูกเลือกทั้งหมดจะไม่นำมาทดลองแบ่งกลุ่มข้อมูลใหม่ เนื่องจากถือว่าได้ผลการทดลองไม่แตกต่างจากการทดลองแบ่งกลุ่มข้อมูลด้วยโครงข่ายประสาทเทียมอย่างเดียว โดยมีรายละเอียดดังนี้

ตารางที่ 4.16 ผลการทำ Feature Selection ข้อมูล Hayes-Roth ด้วย Fast Correlation-Based Filter และแบ่งกลุ่มข้อมูลด้วยโครงข่ายประสาทเทียม

Hayes-Roth	
Hidden nodes = 4, Learning rate = 0.1, จำนวนรอบ = 500	
สำหรับแอททริบิวต์ที่ใช้	2,3,4
จำนวนแลวเฉลี่ยที่แบ่งกลุ่มถูกต้อง	15
จำนวนแลวเฉลี่ยที่แบ่งกลุ่มผิดพลาด	13
เปอร์เซนต์ความถูกต้องเฉลี่ย	53.57%

ตารางที่ 4.17 ผลการทำ Feature Selection ข้อมูล Car Evo ด้วย Fast Correlation-Based Filter และแบ่งกลุ่มข้อมูลด้วยโครงข่ายประสาทเทียม

Car Evo	
Hidden nodes = 5, Learning rate = 0.2, จำนวนรอบ = 500	
สำหรับแอททริบิวต์ที่ใช้	4,6
จำนวนแลวเฉลี่ยที่แบ่งกลุ่มถูกต้อง	1,196
จำนวนแลวเฉลี่ยที่แบ่งกลุ่มผิดพลาด	532
เปอร์เซนต์ความถูกต้องเฉลี่ย	69.21%

ตารางที่ 4.18 ผลการทำ Feature Selection ข้อมูล Chess ด้วย Fast Correlation-Based Filter และแบ่งกลุ่มข้อมูลด้วยโครงข่ายประสาทเทียม

Chess			
Hidden nodes = 4, Learning rate = 0.2, จำนวนรอบ = 300			
ลำดับแอททริบิวต์ที่ใช้	21	10,21,33	3,10,15,16,21,32,33
จำนวนตัวอย่างที่แบ่งกลุ่มถูกต้อง	1,527	1,527	1,527
จำนวนตัวอย่างที่แบ่งกลุ่มผิดพลาด	1,669	1,669	1,669
เปอร์เซนต์ความถูกต้องเฉลี่ย	47.78%	47.78%	47.78%

สำหรับข้อมูล Balance Scale เนื่องจากไม่มีเซตย่อยของแอททริบิวต์ใดเลยที่ถูกเลือก จึงถือว่าได้ผลการทดลองไม่แตกต่างจากการทดลองแบ่งกลุ่มข้อมูลด้วยโครงข่ายประสาทเทียมอย่างเดียว

4.5.4. ผลการทดลองทำ Feature Selection ด้วย Las Vegas Filter และแบ่งกลุ่มข้อมูลด้วยโครงข่ายประสาทเทียม

ในการทำ Feature Selection ข้อมูลต่างๆด้วย Las Vegas Filter ที่ค่า Threshold ในช่วงต่างๆ จำนวน 10 ครั้ง ได้ผลการทดลองดังนี้

ตารางที่ 4.19 ผลการทำ Feature Selection ข้อมูล Hayes-Roth ด้วย Las Vegas Filter

Hayes-Roth						
ค่า Threshold	0	0.1	0.2	0.3	0.4	
ลำดับแอททริบิวต์ที่ถูกเลือก	-	-	-	-	(2,3)	
ค่า Threshold	0.5	0.6	0.7	0.8	0.9	1
ลำดับแอททริบิวต์ที่ถูกเลือก	(4),(2,3)	(3),(1,2),(2,3)	(1)	(1)	(1),(3)	(1)

ตารางที่ 4.20 ผลการทำ Feature Selection ข้อมูล Balance Scale ด้วย Las Vegas Filter

Balance Scale						
ค่า Threshold	0	0.1	0.2	0.3	0.4	
ลำดับแอททริบิวต์ที่ถูกเลือก	-	-	-	(2,3),(1,2)	(1)	
ค่า Threshold	0.5	0.6	0.7	0.8	0.9	1
ลำดับแอททริบิวต์ที่ถูกเลือก	(1)	(1)	(1)	(1)	(1)	(1)

ตารางที่ 4.21 ผลการทำ Feature Selection ข้อมูล Car Evo ด้วย Las Vegas Filter

Car Evo						
ค่า Threshold	0	0.1	0.2	0.3	0.4	
ลำดับแอททริบิวต์ที่ถูกเลือก	-	-	-	(1)	(1)	
ค่า Threshold	0.5	0.6	0.7	0.8	0.9	1
ลำดับแอททริบิวต์ที่ถูกเลือก	(1)	(1),(3)	(1)	(1)	(1)	(1)

ตารางที่ 4.22 ผลการทำ Feature Selection ข้อมูล Chess ด้วย Las Vegas Filter

Chess						
ค่า Threshold	0	0.1	0.2	0.3	0.4	
ลำดับแอททริบิวต์ที่ถูกเลือก	-	(2,4,5,7, 9,10,12,13 ,15,19,21, 22,24,25,2 7,28,30,32 ,33)	(10,12,13, 15,16,28, 30,32,33, 35)	(9,10,12, 13,15,27, 29,30,32)	(3,5,21), (3,4,21)	
ค่า Threshold	0.5	0.6	0.7	0.8	0.9	1
ลำดับแอททริบิวต์ที่ถูกเลือก	(1)	(1)	(1)	(1)	-	-

จากนั้นทำการแบ่งกลุ่มข้อมูลด้วยโครงข่ายประสาทเทียม โดยใช้แอททริบิวต์ที่ได้จากการทำ Feature Selection ด้วย Las Vegas Filter ที่ค่า Threshold ในช่วงต่างๆ ซึ่งนำมาเฉพาะช่วงที่มีบางแอททริบิวต์ถูกเลือกเท่านั้น โดยใช้จำนวนโหนดในชั้น Hidden layer, Learning rate และจำนวนรอบการเรียนรู้เหมือนกับการทดลองที่แบ่งกลุ่มข้อมูลด้วยโครงข่ายประสาทเทียมอย่าง

เดียว สำหรับค่า Threshold ที่ไม่มีแอททริบิวต์ใดถูกเลือกเลยหรือถูกเลือกทั้งหมดจะไม่นำมาทดลองแบ่งกลุ่มข้อมูลใหม่ เนื่องจากถือว่าได้ผลการทดลองไม่แตกต่างจากการทดลองแบ่งกลุ่มข้อมูลด้วยโครงข่ายประสาทเทียมอย่างเดียว โดยมีรายละเอียดดังนี้

ตารางที่ 4.23 ผลการทำ Feature Selection ข้อมูล Hayes-Roth ด้วย Las Vegas Filter และแบ่งกลุ่มข้อมูลด้วยโครงข่ายประสาทเทียม

Hayes-Roth			
Hidden nodes = 4, Learning rate = 0.1, จำนวนรอบ = 500			
สำหรับแอททริบิวต์ที่ใช้	1	3	
จำนวนแลวเอลลีย์ที่แบ่งกลุ่มถูกต้อง	13	13	
จำนวนแลวเอลลีย์ที่แบ่งกลุ่มผิดพลาด	15	15	
เปอร์เซนต์ความถูกต้องเฉลี่ย	46.43%	46.43%	
สำหรับแอททริบิวต์ที่ใช้	4	1,2	2,3
จำนวนแลวเอลลีย์ที่แบ่งกลุ่มถูกต้อง	13	13	17
จำนวนแลวเอลลีย์ที่แบ่งกลุ่มผิดพลาด	15	15	11
เปอร์เซนต์ความถูกต้องเฉลี่ย	46.43%	46.43%	60.71%

ตารางที่ 4.24 ผลการทำ Feature Selection ข้อมูล Balance Scale ด้วย Las Vegas Filter และแบ่งกลุ่มข้อมูลด้วยโครงข่ายประสาทเทียม

Balance Scale			
Hidden nodes = 4, Learning rate = 0.1, จำนวนรอบ = 500			
สำหรับแอททริบิวต์ที่ใช้	1	1,2	2,3
จำนวนแลวเอลลีย์ที่แบ่งกลุ่มถูกต้อง	397	422	438
จำนวนแลวเอลลีย์ที่แบ่งกลุ่มผิดพลาด	228	203	187
เปอร์เซนต์ความถูกต้องเฉลี่ย	63.52%	67.52%	70.08%

ตารางที่ 4.25 ผลการทำ Feature Selection ข้อมูล Car Evo ด้วย Las Vegas Filter และแบ่งกลุ่มข้อมูลด้วยโครงข่ายประสาทเทียม

[illegible]		
[illegible]		
จำนวนแถวเฉลี่ยที่แบ่งกลุ่มถูกต้อง	672	384
[illegible]	1,056	1,344
[illegible]	38.89%	22.22%

ตารางที่ 4.26 ผลการทำ Feature Selection ข้อมูล Chess ด้วย Las Vegas Filter และแบ่งกลุ่มข้อมูลด้วยโครงข่ายประสาทเทียม

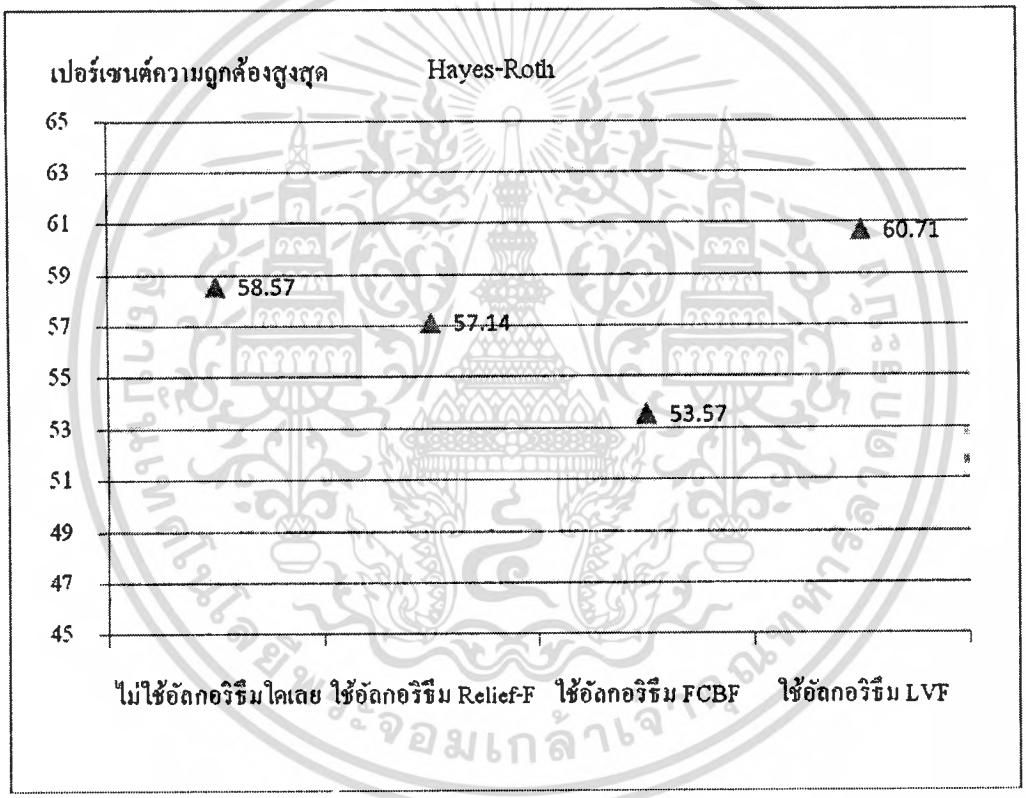
[illegible]			
[illegible]			
[illegible]	1,527	1,527	1,527
[illegible]	1,669	1,669	1,669
[illegible]	47.78%	47.78%	47.78%
[illegible]			
[illegible]			
จำนวนแถวเฉลี่ยที่แบ่งกลุ่มถูกต้อง	1,527	1,527	1,527
[illegible]	1,669	1,669	1,669
[illegible]	47.78%	47.78%	47.78%

4.6 วิเคราะห์ผลการทดลอง

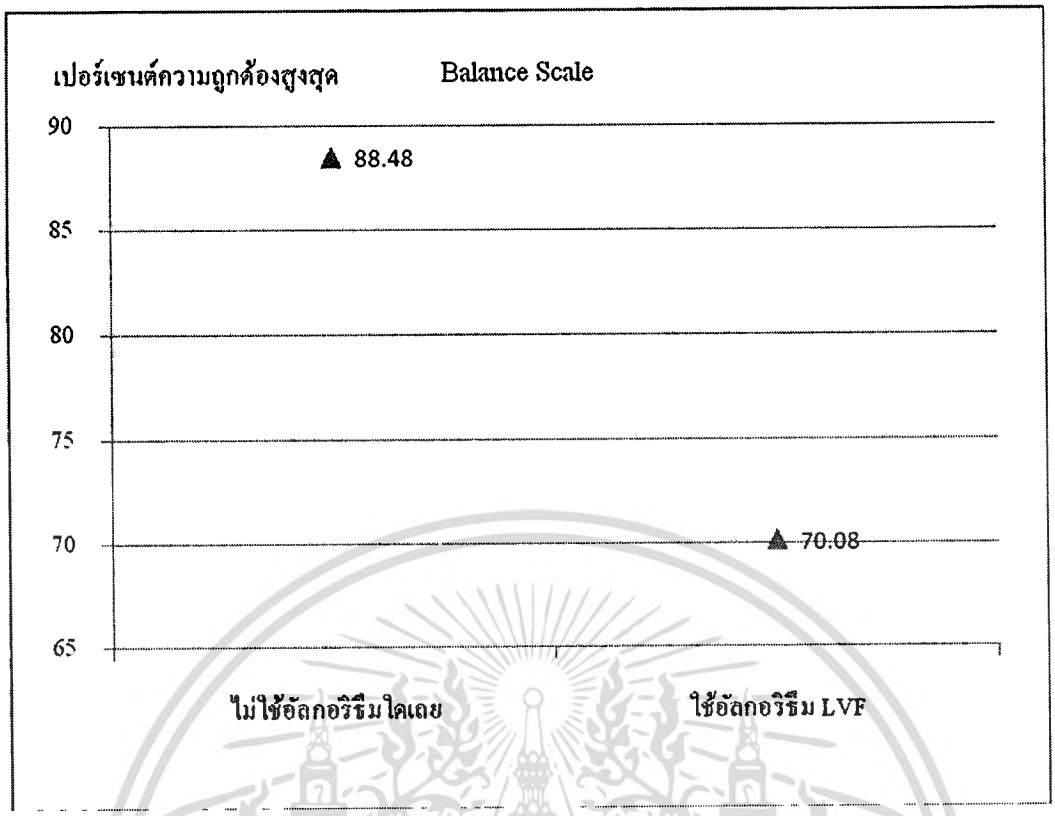
จากการทดลองโดยการเลือกใช้ข้อมูลหลายๆกลุ่มในการทดลองและใช้ค่า Threshold ที่แตกต่างกันในแต่ละอัลกอริธึมนั้น พบว่าการทำ Feature Selection ด้วยอัลกอริธึม Relief-F และ Las Vegas Filter นั้นจะให้ผลลัพธ์ที่หลากหลายในค่า Threshold หนึ่งๆเนื่องจากมีการสุ่มลำดับแถวหรือเซตย่อยของแอททริบิวต์ในแต่ละครั้งที่ไม่เหมือนกัน ทำให้มีโอกาสที่จะได้ค่าน้ำหนัก

ของแอททริบิวต์ที่แตกต่างในแต่ละครั้งแม้ว่าจะเป็นแอททริบิวต์เดิมก็ตาม ในขณะที่ Fast Correlation-Based Filter นั้นจะให้ผลลัพธ์แบบเดียวในค่า Threshold หนึ่งๆ

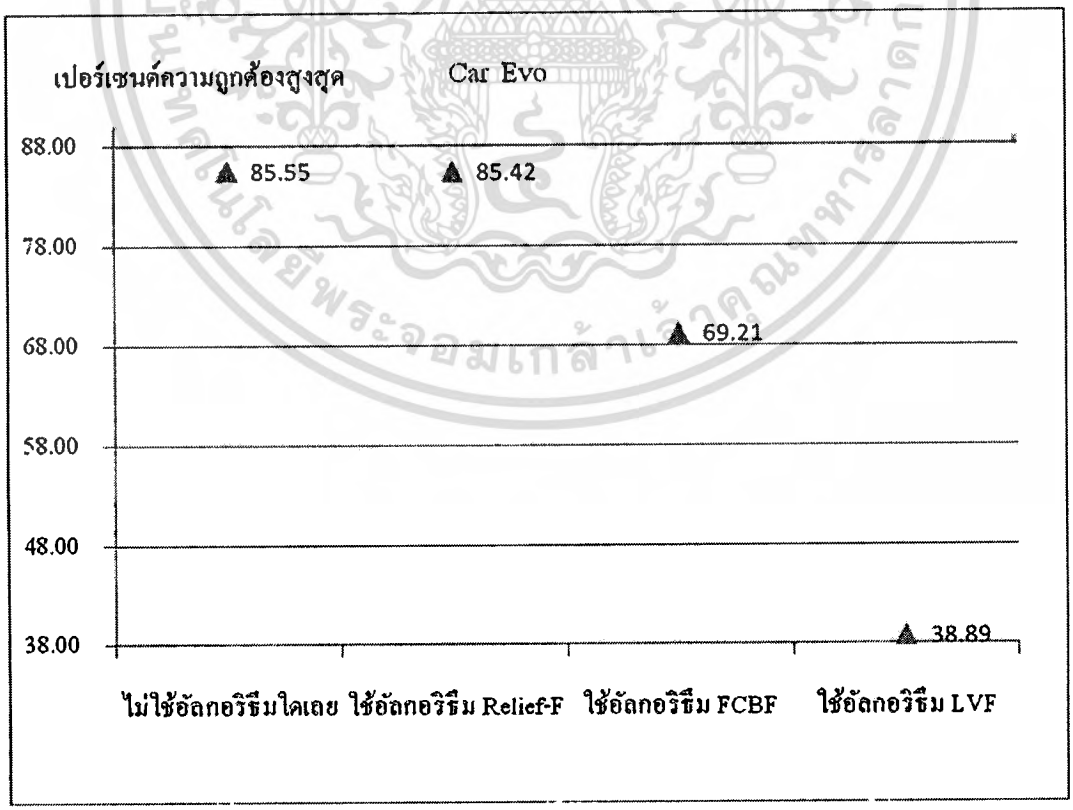
สำหรับผลลัพธ์การแบ่งกลุ่มข้อมูลโดยการทำ Feature Selection ด้วยอัลกอริธึมต่างๆนั้น เมื่อพิจารณาถึงค่าความถูกต้องสูงสุดของการแบ่งกลุ่มข้อมูลแบบใช้และไม่ใช้อัลกอริธึม Feature Selection นำมาเปรียบเทียบกันพบว่าการแบ่งกลุ่มข้อมูลโดยไม่ใช่อัลกอริธึมใดเลยจะให้ค่าความถูกต้องสูงสุด รองลงมาคืออัลกอริธึม Relief-F และ Fast Correlation-Based Filter ตามลำดับ สำหรับอัลกอริธึม Las Vegas Filter นั้นจะให้ค่าความถูกต้องที่มีความแปรปรวน โดยข้อมูลบางชนิดจะให้ค่าความถูกต้องที่สูงกว่าการแบ่งกลุ่มข้อมูลโดยไม่ใช่อัลกอริธึมใดเลยในขณะที่บางชนิดจะให้ค่าความถูกต้องที่ต่ำกว่าการแบ่งกลุ่มข้อมูลโดยไม่ใช่อัลกอริธึมใดเลย



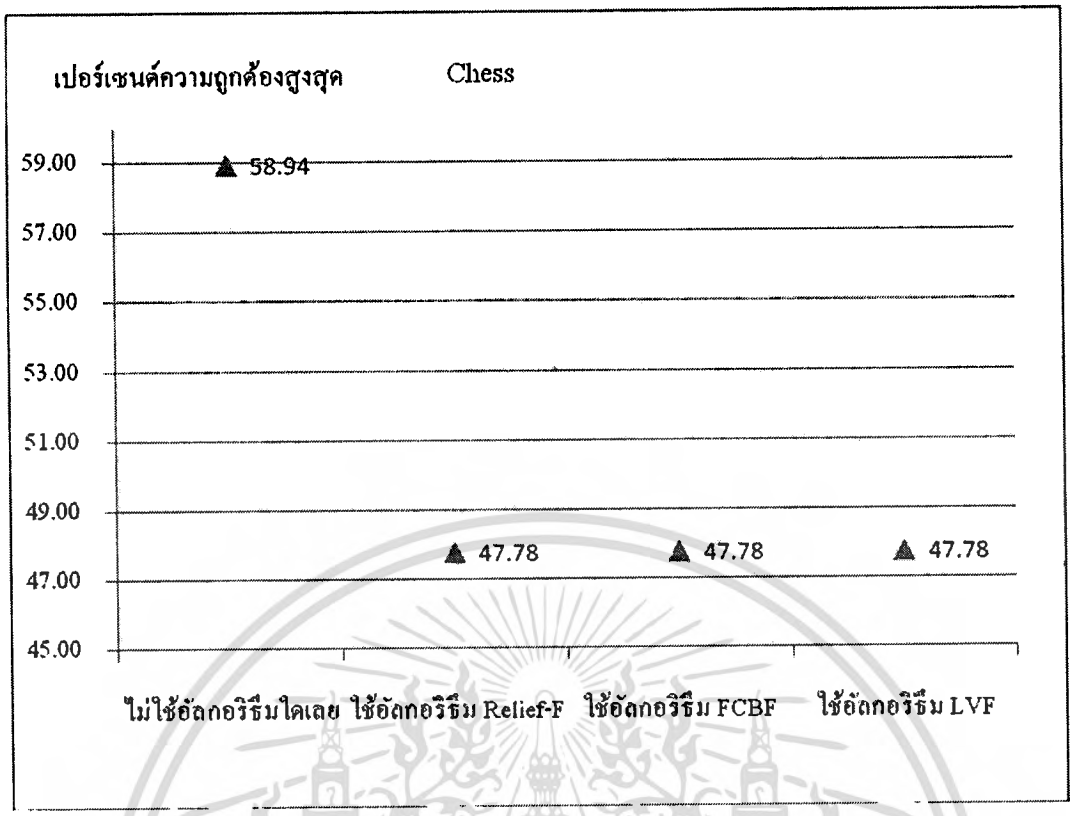
รูปที่ 4.5 กราฟเปรียบเทียบผลการแบ่งกลุ่มข้อมูล Hayes-Roth ด้วยโครงข่ายประสาทเทียมแบบไม่ใช้และใช้อัลกอริธึม Feature Selection



รูปที่ 4.6 กราฟเปรียบเทียบผลการแบ่งกลุ่มข้อมูล Balance Scale ด้วยโครงข่ายประสาทเทียมแบบไม่ใช้และใช้อัลกอริธึม Feature Selection



รูปที่ 4.7 กราฟเปรียบเทียบผลการแบ่งกลุ่มข้อมูล Car Evo ด้วยโครงข่ายประสาทเทียมแบบไม่ใช้และใช้อัลกอริธึม Feature Selection



รูปที่ 4.8 กราฟเปรียบเทียบผลการแบ่งกลุ่มข้อมูล Chess ด้วยโครงข่ายประสาทเทียมแบบไม่ใช้และใช้อัลกอริธึม Feature Selection

เมื่อพิจารณาถึงจำนวนแอททริบิวต์ที่ลดลงเมื่อทำ Feature Selection ในแต่ละอัลกอริธึมนั้น พบว่าที่ช่วงค่าความถูกต้องเฉลี่ยในการแบ่งกลุ่มข้อมูลใกล้ๆกันนั้น อัลกอริธึม Relief-F และ Fast Correlation-Based Filter จะสามารถลดจำนวนแอททริบิวต์ได้พอๆกัน และเมื่อพิจารณาถึงจำนวนแอททริบิวต์สูงสุดที่ลดลงในแต่ละอัลกอรธึมนั้นพบว่า อัลกอริธึม Relief-F และ Las Vegas Filter นั้นจะสามารถลดจำนวนแอททริบิวต์ได้มากที่สุดพอๆกัน

4.7 สรุปผลการทดลอง

จากการทำทดลองสามารถสรุปได้ว่า การทำ Feature Selection ในแต่ละอัลกอริธึมนั้น Relief-F จะให้ค่าความถูกต้องเมื่อนำไปใช้ในการแบ่งกลุ่มข้อมูลสูงที่สุดสุด รองลงมาคือ Fast Correlation-Based Filter ในขณะที่อัลกอริธึม Las Vegas Filter นั้นจะให้ค่าความถูกต้องที่มีความแปรปรวน สูงบ้างต่ำบ้าง ขึ้นอยู่กับชนิดของข้อมูลที่ใช้ โดยอัลกอริธึม Relief-F และ Fast Correlation-Based Filter นั้นจะสามารถลดจำนวนของแอททริบิวต์ได้พอๆกันในระดับความถูกต้องในการแบ่งกลุ่มข้อมูลที่ใกล้เคียงกัน แต่เมื่อไม่พิจารณาถึงค่าความถูกต้องในการแบ่งกลุ่มข้อมูลแล้ว Relief-F และ Las Vegas Filter นั้นจะสามารถลดจำนวนแอททริบิวต์ของข้อมูลได้มากที่สุด

บทที่ 5
สรุปผลการศึกษาและข้อเสนอแนะ

จากการศึกษาและทดลองเปรียบเทียบประสิทธิภาพอัลกอริธึม Feature Selection ในการแบ่งกลุ่มข้อมูลนั้น จะสามารถสรุปผลการดำเนินงานได้ดังนี้

5.1 สรุปผลการศึกษา

ในการทำ Feature Selection โดยใช้อัลกอริธึม Relief-F, Fast Correlation-Based Filter และ Las Vegas Filter นั้น Relief-F มีประสิทธิภาพที่ดีที่สุดเมื่อพิจารณาถึงความถูกต้องในการนำไปแบ่งกลุ่มข้อมูลและความสามารถในการลดจำนวนแอททริบิวต์ของข้อมูลไปพร้อมๆกัน ในขณะที่ Fast Correlation-Based Filter นั้นจะมีประสิทธิภาพรองลงมา ส่วน Las Vegas Filter นั้นประสิทธิภาพไม่แน่นอน ขึ้นอยู่กับชนิดของข้อมูลที่ใช้งาน โดยทั้งนี้การทำ Feature Selection นั้นมีแนวโน้มที่จะทำให้ความถูกต้องในการแบ่งกลุ่มข้อมูลลดลงได้เมื่อเปรียบเทียบกับการแบ่งกลุ่มข้อมูลตามปกติที่ไม่มีการทำ Feature Selection

5.2 ประโยชน์ที่ได้รับจากการศึกษา

1. ทำให้รู้จักหลักการทำงานของการทำ Feature Selection ในการแบ่งกลุ่มข้อมูล
2. ทำให้มีความเข้าใจในขั้นตอนการทำงานของอัลกอริธึม Relief-F, Fast Correlation-Based Filter และ Las Vegas Filter ดีมากขึ้น
3. ทำให้ทราบถึงข้อดีข้อเสียของแต่ละอัลกอริธึในการทำ Feature Selection รวมทั้งข้อจำกัดในการใช้งาน
4. ช่วยพัฒนาทักษะในการพัฒนาโปรแกรมในการแบ่งกลุ่มข้อมูลด้วยโครงข่ายประสาทเทียม

5.3 ข้อเสนอแนะ

ในการทดลองนี้ มีการกำหนดค่า Threshold ของแต่ละอัลกอริธึมที่ใช้ในการทำ Feature Selection นั้นออกเป็นช่วงๆ ซึ่งค่า Threshold นั้นอาจจะไม่เหมาะสมกับข้อมูลแต่ละประเภททำให้ได้ผลลัพธ์ที่ยังไม่ถูกต้องหรือเหมาะสมเท่าที่ควร ถ้ามีการทดลองเปลี่ยนค่า Threshold ของแต่ละอัลกอริธึมให้เหมาะสมกับชุดข้อมูลที่ใช้ในแต่ละชุด อาจจะทำให้ได้ผลลัพธ์ที่ดีขึ้นหรือแตกต่างจากการทดลองนี้ ในขณะเดียวกันการกำหนดจำนวนรอบในการสุ่มของอัลกอริธึม

Relief-F และ Las Vegas Filter ในการทดลองนี้ยังมีค่าที่ตายตัว ทำให้อาจจะไม่เหมาะสมกับชุดข้อมูลที่ใช้ทุกชุดทำให้ได้ผลลัพธ์ที่ยังไม่ถูกต้องเหมาะสมได้เช่นกัน ซึ่งถ้ามีการทดลองเปลี่ยนค่าจำนวนรอบในการสุ่มของแต่ละอัลกอริธึมใหม่ให้เหมาะสมกับชุดข้อมูลที่ใช้ในแต่ละชุด อาจจะทำให้ได้ผลลัพธ์ที่ดีขึ้นหรือแตกต่างจากการทดลองนี้ได้เช่นกัน

บรรณานุกรม

Adit. **Neural Network Algorithms.** 2009. [Online] Available: http://www.adit.co.uk/html/programming_a_neural_networn.html.

Azofra, A. et al. 2004. "A Feature Set Measure Based on Relief" University of Cordoba.

Bins, J. C. 2000. "Feature Selection Form Huge Feature Sets in The Context of Computer Vision" Ph.D.Thesis of Colorado State University.

Guyon, I. and Elisseeff, A. 2003. "An Introduction to Variable and Feature Selection" **Journal of Machine Learning Research 3**. 1157-1182.

Han, J. and Kamber, M. 2001. **Data Mining : Concepts and Techniques.** San Francisco: Morgan Kanfmann.

Liu, H. and Yu, L. April 2005. "Toward Integrating Feature Selection Algorithms for Classification and Clustering. San Francisco" **IEEE Transactions on Knowledge and Data Engineering.Vol. 17. No. 4.**

Robnik, M. and Kononenko, I. 2003. "Theoretical and Empirical Analysis of ReliefF and RReliefF" **Machine Learning Journal.**

Temple University. 2009. **Building Classification Models: ID3 and C4.5.** [Online] Available: http://www.cis.temple.edu/~ingargio/cis587/readings/id3-c45.html.

University of Bozen-Bolzano. 2009. **ID3 Classification Algorithm.** [Online] Available: http://www.inf.unibz.it/dis/teaching/DWDM07/reports/5/id3.pdf.

ประวัติผู้เขียน

ชื่อ-นามสกุล	กฤตมุข ลีศิริชัยกุล
วัน เดือน ปีเกิด	28 ตุลาคม 2526
วุฒิการศึกษา	วิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์
สถานที่สำเร็จการศึกษา	จุฬาลงกรณ์มหาวิทยาลัย
ปีที่สำเร็จการศึกษา	2549