เหมืองข้อมูล

DATA MINING

สุณิสา ตันติธรรมผาติ

อัณศยา ศิริแสง

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2557

เหมืองข้อมูล

DATA MINING

สุณิสา ตันติธรรมผาติ
อัณศยา ศิริแสง

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2557

ปริญญานิพนธ์ปีการศึกษา 2557

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เหมืองข้อมูล

DATA MINING

ผู้จัดทำ

1. นางสาวสุณิสา ตันติธรรมผาติ รหัสนักศึกษา 54011389
2. นางสาวอัณศยา ศิริแสง รหัสนักศึกษา 54011538

_____ อาจารย์ที่ปรึกษา

(ดร. วรวัฒน์ ลิ้มโภคา)

เหมืองข้อมูล

นางสาวสุณิสา ตันติธรรมผาติ 54011389
นางสาวอัณศยา ศิริแสง 54011538
ดร.วรวัฒน์ ลิ้มโภคา อาจารย์ที่ปรึกษา
ปีการศึกษา 2557

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้มีวัตถุประสงค์เพื่อศึกษากระบวนการในการทำเหมืองข้อมูล (Data mining) และเปรียบเทียบประสิทธิภาพของซอฟต์แวร์ที่เป็นที่นิยมใช้ในการทำเหมืองข้อมูล 3 ตัว ได้แก่ เวก้า (Weka), ราปิดมายเนอร์ (RapidMiner) และอาร์ คาต้ามายนิ่ง (R Data mining) ทำการทดลองโดยใช้รูปแบบการทำเหมืองข้อมูลแบบการจำแนกข้อมูล (Classification) กับชุดข้อมูล 7 ชุด ที่มีลักษณะแต่กต่างกัน โดยทำการจำแนกข้อมูลเหล่านี้ด้วยอัลกอริทึม 3 ตัวที่เป็นที่นิยมซึ่งได้แก่ อัลกอริทึมนาอีฟเบย์ (Naïve Bayes), อัลกอริทึมต้นไม้ตัดสินใจ (Decision Tree) และอัลกอริทึมอดาบูส (AdaBoost) โดยอัลกอริทึมแต่ละตัวจะมีวิธีการคำนวณในการสร้างโมเดลจากชุดข้อมูลเรียนรู้ที่แตกต่างกันและเมื่อนำชุดข้อมูลชุดเดียวกันมาทดสอบกับแต่ละอัลกอริทึมจะทำให้เราทราบว่าชุดข้อมูลประเภทใดเหมาะที่จะใช้อัลกอริทึมใดเพื่อให้ได้ประสิทธิภาพในการจำแนกข้อมูลที่ดีที่สุด ซึ่งพบว่าซอฟต์แวร์แต่ละตัวมีประสิทธิภาพในการสร้างโมเดลสำหรับทำนายได้ใกล้เคียงกัน ไม่มีซอฟต์แวร์ใดที่มีผลโดดเด่นไปกว่าซอฟต์แวร์อื่น จึงสามารถสรุปได้ว่าผลเปอร์เซ็นต์ความถูกต้องของโมเดลนั้นขึ้นอยู่กับแต่ละอัลกอริทึมที่ผู้ใช้เลือกใช้และขึ้นอยู่กับลักษณะที่ต่างกันของชุดข้อมูล เช่น ลักษณะของชุดข้อมูล, จำนวนอินสแตนซ์และจำนวนแอตทริบิวต์ในชุดข้อมูล เป็นต้น

ในการทดลองมีการแบ่งชุดข้อมูลออกเป็นชุดข้อมูลเรียนรู้และชุดข้อมูลทดสอบจากชุดข้อมูลเดิม โดยมีวัตถุประสงค์เพื่อศึกษาว่าการแบ่งชุดข้อมูลเรียนรู้ต่อชุดข้อมูลทดสอบในอัตราเท่าใดจะทำให้เกิดการสร้างโมเดลที่มีประสิทธิภาพในการจำแนกได้ถูกต้องมากกว่า โดยการทดลองมีการแบ่งชุดข้อมูลทั้งหมด 3 แบบคือ หนึ่งแบ่งเป็นชุดข้อมูลเรียนรู้ 70 เปอร์เซ็นต์และชุดข้อมูลทดสอบ 30 เปอร์เซ็นต์ สองแบ่งเป็นชุดข้อมูลเรียนรู้ 80 เปอร์เซ็นต์และชุดข้อมูลทดสอบ 20 เปอร์เซ็นต์ และสามแบ่งเป็นชุดข้อมูลเรียนรู้ 90 เปอร์เซ็นต์และชุดข้อมูลทดสอบ 10 เปอร์เซ็นต์ จากการทดลองสามารถสรุปได้ว่าการแบ่งชุดข้อมูลแบบแบ่งเป็นชุดข้อมูลเรียนรู้ 90 เปอร์เซ็นต์และชุดข้อมูลทดสอบ 10 เปอร์เซ็นต์นั้นมีประสิทธิภาพในการนำสร้างโมเดลสำหรับจำแนกข้อมูลมากที่สุด ทั้งนี้กระบวนการทำเหมืองข้อมูลเป็นวิธีการหนึ่งที่ทำให้สามารถนำข้อมูลที่มีอยู่อย่างมากมายในธุรกิจหรือกิจกรรมประเภทต่างๆ มาใช้ให้เกิดประโยชน์สูงสุดได้ ซึ่งเป็นสิ่งที่น่าสนใจและมีประโยชน์อย่างยิ่งในปัจจุบัน

DATA MINING

Ms. Sunisa Tantitumpati 54011389

Ms. Ansaya Sirisang 54011538

Dr. Voravat Limpoka Advisor

Academic Year 2014

ABSTRACT

The purpose of this report is to study about the data mining processes and to compare the performances between three popular data mining software, for example, Weka, RapidMiner and R Data Mining. To compare the performances of this three softwares, we use three popular classification algorithms, such as Naïve Bayes, Decision tree and AdaBoost with seven datasets, which each have several characteristics. Each algorithm has a different solution to build a predictive model from each dataset. When we use different algorithms with same dataset. We will know which algorithms is suitable for that dataset for the best efficiency of prediction. This experiment makes us know that each software has similar correctly classified percentage. It does not has any software which is outstanding more than others. And we can conclude that the correctly classified percentage depends on different algorithms and different dataset characteristics, such as the characteristic of dataset, amount of instance, amount of attribute in the dataset and etc.

In this experiment, We have Split the original dataset into training set and test set by the different three ways as follows the first is 70% of training set and 30% of test set, the second is 80% of training set and 20% of test set and the last is 90% of training and 10% of test set. We do this for finding which way gives the best of predictive model and we can summarize that 90% of training and 10% of test set is the best. Data mining is the process of analyzing data from different perspectives and summarizing a big data into useful information. It is very interesting useful.

กิตติกรรมประกาศ

คณะผู้จัดทำขอขอบพระคุณอาจารย์ที่ปรึกษาโครงงาน ดร.วรวัฒน์ ลิ้มโภคาที่คอยให้ความสนใจและใส่ใจสอบถามถึงความคืบหน้าของโครงงานอย่างสม่ำเสมอ อีกทั้งยังคอยให้คำแนะนำและให้ความช่วยเหลือในด้านต่างๆ ตลอดมา นอกจากนี้ยังขอขอบพระคุณคุณศิราณี จรัสวชิรกุล และคุณชานนท์ ทรัพย์สำราญที่ให้ความช่วยเหลือและแนะนำที่ดีกับผู้จัดทำ

ขอขอบพระคุณสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง และสถาบันการศึกษาในอดีต ที่ให้โอกาสดีๆทางการศึกษาแก่ข้าพเจ้า

ขอขอบคุณเพื่อนๆ ในสาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่เป็นกำลังใจให้กันตลอดการทำงาน รวมทั้งให้คำปรึกษา และช่วยเหลือกันในด้านต่างๆตลอดระยะเวลาที่ผ่านมา

สำหรับคุณงามความดีอันใดที่เกิดจากรายงานเล่มนี้คณะผู้จัดทำขอมอบให้บิดามารดาซึ่งเป็นที่รักและเคารพยิ่งตลอดจนครูอาจารย์ที่เคารพทุกท่านที่ได้ประสิทธิ์ประสาทวิชาความรู้และประสบการณ์ที่ดีแก่คณะผู้จัดทำ

นางสาว สุณิสา ตันติธรรมผาติ
นางสาว อัณศยา ศิริแสง

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

สารบัญรูป

สารบัญรูป (ต่อ)

สารบัญรูป (ต่อ)

รูป หน้า

สารบัญรูป (ต่อ)

รูป หน้า

สารบัญรูป (ต่อ)

สารบัญรูป (ต่อ)

สารบัญรูป (ต่อ)

รูป หน้า

สารบัญรูป (ต่อ)

รูป หน้า

สารบัญรูป (ต่อ)

สารบัญรูป (ต่อ)

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มาของโครงงาน

ในปัจจุบันทั้งในวงการของธุรกิจประเภทต่างๆ ทั้งธนาคาร ห้างสรรพสินค้า หรือในวงการทางด้านการแพทย์ ทางด้านวิทยาศาสตร์ การวิจัยค้นคว้าทดลองต่างต้องมีการเก็บข้อมูลเพื่อใช้ในการขับเคลื่อนหรือพัฒนาองค์กรให้มีความก้าวหน้าเติบโตต่อไป ก่อให้เกิดการแข่งขันทางด้านธุรกิจซึ่งข้อมูลที่แต่ละในองค์กรมีอยู่จะเป็นเครื่องมือสำคัญในการตัดสินใจการวางแผนในการดำเนินธุรกิจต่อไป แต่เนื่องจากการเก็บข้อมูลในธุรกิจประเภทต่างๆ นั้นอาจมีการเก็บข้อมูลที่สำคัญมาอย่างยาวนาน หรือเป็นการเก็บข้อมูลที่เยอะมากๆ ทำให้มีข้อมูลในระบบที่เยอะจนเกินไป ไม่สามารถค้นหาข้อมูลที่สำคัญเจอ หรืออาจทำให้เกิดการค้นหาข้อมูลเจอแต่ล่าช้าทำให้สูญเสียโอกาสทางธุรกิจไปอย่างน่าเสียดายได้ ซึ่งกระบวนการทำเหมืองข้อมูลเป็นวิธีการที่จะช่วยให้สามารถค้นเจอรูปแบบความสัมพันธ์ของข้อมูลที่มีในระบบได้หรือที่เรียกกันว่า “Knowledge” จากข้อมูลที่มีอยู่ในระบบอย่างมหาศาล เปรียบเทียบได้กับการทำเหมืองแร่ ซึ่งเราจะต้องทำการสกัดแร่จากดินที่มีอยู่อย่างมหาศาลเพื่อให้ได้แร่ที่มีค่า แร่ในที่นี้ก็เปรียบได้กับข้อมูลที่เราต้องการนำไปใช้ประโยชน์นั่นเอง ซึ่งในปัจจุบันนี้การทำเหมืองข้อมูลกำลังได้รับความสนใจเป็นอย่างมาก ทั้งนี้ยังมีเครื่องมือในการทำเหมืองข้อมูลหลายๆ อย่างที่ถูกพัฒนาขึ้นมาเพื่อการทำเหมืองข้อมูล ซึ่งเครื่องมือแต่ละตัวก็จะมีความสามารถ วิธีการใช้งาน และอาจได้รับประสิทธิภาพที่แตกต่างๆ กัน การเลือกใช้เครื่องมือในการทำเหมืองข้อมูลให้เหมาะสมกับชุดข้อมูลและการนำไปใช้จึงเป็นสิ่งสำคัญอีกส่วนหนึ่งที่จะทำให้เราได้รูปแบบความสัมพันธ์ของข้อมูลที่มีประสิทธิภาพ นอกจากนี้ในชุดข้อมูลแต่ละตัวยังมีอัลกอริทึมที่ใช้ในการเรียนรู้อีกมากมาย การเลือกใช้อัลกอริทึมให้เหมาะสมกับข้อมูลแต่ละประเภทจึงเป็นสิ่งสำคัญที่จะส่งผลให้เราได้รูปแบบความสัมพันธ์ที่ดีหรือไม่ดีก็ได้

1.2 วัตถุประสงค์ของโครงงาน

วัตถุประสงค์ของรายงานฉบับนี้จัดทำเพื่อศึกษาและทดลองการทำเหมืองข้อมูล รวมถึงศึกษาการใช้งานซอฟต์แวร์ต่างๆ ที่เกี่ยวข้องกับการทำเหมืองข้อมูล ศึกษาและทดลองให้เห็นถึงประสิทธิภาพและการทำงานของแต่ละซอฟต์แวร์ รวมไปถึงการศึกษาและทดลองใช้งานแต่ละอัลกอริทึมที่มีในซอฟต์แวร์แต่ละตัว เพื่อดูว่าข้อมูลประเภทใดเหมาะสมจะใช้กับอัลกอริทึมไหน เพื่อสามารถนำไปเลือกใช้ได้อย่างเหมาะสมและได้โมเดลที่มีประสิทธิภาพมากที่สุด

1.3 ขอบเขตของโครงงาน

เป็นการศึกษาประสิทธิภาพการทำงานของซอฟต์แวร์ในการทำเหมืองข้อมูล โดยเปรียบเทียบประสิทธิภาพของแต่ละอัลกอริทึมที่มีในซอฟต์แวร์ทั้งสามตัว โดยการควบคุมข้อมูลให้เหมือนกัน เพื่อศึกษาว่าหากเป็นอัลกอริทึมเดียวกันแต่ซอฟต์แวร์ต่างกันผลลัพธ์ที่ได้นั้นจะมีความเหมือนหรือแตกต่างกันมากน้อยเพียงใด อีกทั้งในซอฟต์แวร์แค่ละตัวยังมีอัลกอริทึมที่หลากหลายให้ผู้ใช้เลือกใช้ โดยขอบเขตของการศึกษาทดลองในโครงงานนี้จะทำการทดลองเพื่อหาความเหมาะสมของแต่ละอัลกอริทึมเทียบกับชุดข้อมูลลักษณะต่างๆ ว่าแต่ละอัลกอริทึมนั้นเหมาะสมกับข้อมูลลักษณะใด นอกจากนี้ในการทดสอบประสิทธิภาพของโมเดลที่ได้จากการเรียนรู้ของเครื่องมือ ผู้ใช้ยังสามารถทำการตั้งค่ารูปแบบในการทดสอบได้ ทั้งนี้คณะผู้จัดทำจะทำการศึกษารูปแบบการทดสอบที่มีการแบ่งข้อมูลส่วนหนึ่งเป็นชุดข้อมูลเรียนรู้ และอีกส่วนหนึ่งเป็นชุดข้อมูลทดสอบว่าหากมีการแบ่งชุดข้อมูลในอัตราที่แตกต่างกันประสิทธิภาพของโมเดลที่ได้จะยังคงมีความเหมือนกันหรือแตกต่างกันมากน้อยเพียงใด การแบ่งชุดข้อมูลแบบใดจะทำให้ได้โมเดลที่มีประสิทธิภาพมากที่สุด

1.4 วิธีการดำเนินการ

1) ศึกษาเกี่ยวกับการทำเหมืองข้อมูล ความหมาย ความสำคัญ และประโยชน์ที่คาดว่าจะได้รับจากกระบวนการทำเหมืองข้อมูล การนำไปใช้ประโยชน์ในด้านต่างๆ
2) ศึกษาเกี่ยวกับเครื่องมือที่ใช้ในการทำเหมืองข้อมูลที่นิยมใช้กันมากที่สุดในปัจจุบัน ซึ่งจะให้ความสนใจเฉพาะประเภทที่เป็นโอเพ่นซอส (Open source) รูปแบบการใช้งาน การรองรับสกุลไฟล์ข้อมูล วิธีการใช้งาน และการรายงานผลข้อมูล รวมไปถึงองค์ประกอบการตั้งค่าส่วนต่างๆ ภายในซอฟต์แวร์
3) ศึกษาเกี่ยวกับอัลกอริทึมต่างๆที่มีในซอฟต์แวร์แต่ละตัว รวมทั้งหาข้อมูลเกี่ยวกับอัลกอริทึมที่ได้รับความนิยม ศึกษาการทำงานของแต่ละอัลกอริทึมเพื่อความเข้าใจมากยิ่งขึ้นในการนำมาใช้งานการปรับแต่งค่าพารามิเตอร์ต่างๆ ในแต่ละอัลกอริทึม
4) ทำการวางแผนรูปแบบการทดลองโดยแบ่งเป็น การหาความเหมาะสมของแต่ละอัลกอริทึมกับชุดข้อมูลแต่ละรูปแบบ การหาประสิทธิภาพของการแบ่งชุดข้อมูลที่แตกต่างกัน และการหาประสิทธิภาพของซอฟต์แวร์แต่ละตัว
5) ทำการทดลองตามแผนการทดลองที่กำหนดไว้ ศึกษา เปรียบเทียบ และวิเคราะห์ผลลัพธ์ที่ได้จากการทดลอง เพื่อหาข้อสรุปในแต่ละประเด็น

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1) ได้รับความรู้ความเข้าใจเกี่ยวกับกระบวนการในการทำเหมืองข้อมูล ประโยชน์และการนำการทำเหมืองข้อมูลไปใช้จริงในด้านต่างๆ
2) ได้รับความรู้ความเข้าใจและทักษะเกี่ยวกับวิธีการใช้งานของซอฟต์แวร์ในการทำเหมืองข้อมูล คือ Weka, RapidMiner และ R Data Mining
3) ได้รับความรู้ความเข้าใจเกี่ยวกับการทำงานของอัลกอริทึมในแต่ละซอฟต์แวร์
4) ได้รับความรู้ความเข้าใจในการใช้อัลกอริทึม Decision tree, Naïve Bayes, OneR, ZeroR, AdaBoost และ IBk รวมถึงการตั้งค่าพารามิเตอร์ต่างๆ ของแต่ละอัลกอริทึม
5) ทำให้เข้าใจถึงกระบวนการในการทำเหมืองข้อมูล ทั้งขั้นตอนการเรียนรู้ข้อมูลเพื่อหาความสัมพันธ์ของข้อมูลเพื่อสร้างโมเดล ขั้นตอนในการวัดประสิทธิภาพของโมเดลด้วยการนำชุดข้อมูลที่เกี่ยวข้องมาทำการทดสอบ
6) ได้รับความรู้ความเข้าใจในการอ่านค่าผลการทดลองในส่วนต่างๆ และในรูปแบบต่างๆ
7) ได้รับความรู้ความเข้าใจในการเลือกใช้อัลกอริทึมและซอฟต์แวร์ให้สอดคล้องและเหมาะสมกับชุดข้อมูลที่จะนำมาดำเนินการทำเหมืองข้อมูล

1.6 ส่วนประกอบของปริญญานิพนธ์

รายงานฉบับนี้ได้แบ่งเนื้อหาออกเป็น 5 บทด้วยกันคือ

บทที่ 1 บทนำ กล่าวถึงความสำคัญและที่มาของโครงงาน วัตถุประสงค์ของโครงงาน ขอบเขตของโครงงาน วิธีการดำเนินการ ประโยชน์ที่คาดว่าจะได้รับและส่วนประกอบของรายงาน

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง กล่าวถึงทฤษฎีพื้นฐานต่างๆ ที่เกี่ยวข้อง ซึ่งมีการนำมาใช้ในโครงงาน ทั้งซอฟต์แวร์ที่ใช้ และอัลกอริทึมที่เกี่ยวข้อง

บทที่ 3 การวิเคราะห์และการออกแบบ กล่าวถึงชุดข้อมูลที่ใช้ในการทดลอง แนวทางการออกแบบการทดลอง และชุดเครื่องมือที่ใช้ในการทดลอง

บทที่ 4 การทดลองและผลการทดลอง กล่าวถึงการดำเนินการทดลอง การควบคุมชุดข้อมูล การแบ่งชุดข้อมูล การเปลี่ยนค่าทดลองการสุ่ม อธิบายรายละเอียดการทดลองทำเหมืองข้อมูลด้วยซอฟต์แวร์ต่างๆ การทดลองด้วยการแบ่งชุดข้อมูลที่ต่างกัน การทดลองเพื่อศึกษาอัลกอริทึมที่เหมาะสมกับข้อมูลแต่ละประเภท การทดลองเพื่อเปรียบเทียบประสิทธิภาพของซอฟต์แวร์

บทที่ 5 บทสรุปและข้อเสนอแนะ กล่าวถึงการวิเคราะห์ผลที่ได้จากการทดลอง ประสิทธิภาพของซอฟต์แวร์ จุดแข็งจุดอ่อนของซอฟต์แวร์ อัลกอริทึมที่เหมาะสมกับข้อมูลแต่ละประเภท อัตราส่วนของชุดข้อมูลเรียนรู้และชุดข้อมูลทดสอบที่เหมาะสม ปัญหาอุปสรรคและการแก้ไขปัญหา รวมถึงแนวทางการพัฒนาต่อ

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

2.1 เหมืองข้อมูล

เนื่องจากในปัจจุบันเป็นยุคที่ข้อมูลข่าวสารและสารสนเทศมีความสำคัญ การเผยแพร่ข้อมูลข่าวสารหรือการคิดหาแรงจูงใจต่างๆ ที่ตรงกับความต้องการและพฤติกรรมของลูกค้าจึงเป็นสิ่งจำเป็น การทำเหมืองข้อมูลถือเป็นอีกหนึ่งวิธีที่สำคัญในการศึกษาพฤติกรรมของลูกค้า ช่วยในการค้นหาและทำนายความต้องการของลูกค้าทำให้องค์กรสามารถตอบสนองต่อความต้องการของลูกค้าได้อย่างตรงประเด็น เกิดความพึงพอใจของลูกค้า และทำให้องค์กรประสบความสำเร็จมากขึ้น

2.1.1 ความหมายและเทคนิคในการทำเหมืองข้อมูล

การทำเหมืองข้อมูล (Data mining) หรือการค้นหาความรู้ในฐานข้อมูล (Knowledge Discovery in Databases: KDD) คือกระบวนการที่กระทำกับข้อมูลจำนวนมากเพื่อค้นหารูปแบบและความสัมพันธ์ที่ซ่อนอยู่ในชุดข้อมูลนั้น โดยอาศัยหลักการทางสถิติ, การจดจำรูปแบบ (Pattern recognition), การเรียนรู้ของเครื่องมือ (Machine learning) และหลักการทางคณิตศาสตร์ เป็นต้น ในปัจจุบันการทำเหมืองข้อมูลได้ถูกนำไปประยุกต์ใช้ในงานหลายประเภท ทั้งในด้านธุรกิจที่ช่วยในการตัดสินใจของผู้บริหาร ในด้านวิทยาศาสตร์และการแพทย์รวมทั้งในด้านเศรษฐกิจและสังคม

เทคนิคในการทำเหมืองข้อมูล

1) กฎความสัมพันธ์ (Association rule) เป็นกระบวนการหนึ่งในการทำ Data Mining ที่ได้รับความนิยมมาก โดยจะใช้ในการหาความสัมพันธ์ของข้อมูลสองชุดหรือมากกว่าสองชุดขึ้นไปภายในกลุ่มข้อมูลที่มีขนาดใหญ่ ในการหากฎความสัมพันธ์นั้นจะมีขั้นตอนวิธีการหาหลายวิธีด้วยกัน แต่ขั้นตอนวิธีที่เป็นที่รู้จักและใช้อย่างแพร่หลายคือขั้นตอนวิธี Apriori การใช้กฎความสัมพันธ์ จะช่วยแสดงความสัมพันธ์ของเหตุการณ์หรือวัตถุที่เกิดขึ้นพร้อมกัน ตัวอย่างของการประยุกต์ใช้กฎเชื่อมโยง เช่น การวิเคราะห์ข้อมูลการขายสินค้า โดยเก็บข้อมูลจากระบบ ณ จุดขาย (POS) หรือร้านค้าออนไลน์แล้วพิจารณาสินค้าที่ผู้ซื้อมักจะซื้อพร้อมกัน เช่น ถ้าพบว่าคนที่ซื้อเทปวิดีโอมักจะซื้อเทปกาวด้วย ร้านค้าก็อาจจะจัดร้านให้สินค้าสองอย่างอยู่ใกล้กัน เพื่อเพิ่มยอดขาย หรืออาจจะพบว่าหลังจากคนซื้อหนังสือ ก แล้ว มักจะซื้อหนังสือ ข ด้วย ก็สามารถนำความรู้นี้ไปแนะนำผู้ที่กำลังจะซื้อหนังสือ ก ได้

2) การจำแนกประเภทข้อมูล (Data classification) เป็นการทำนายประเภทของวัตถุจากคุณสมบัติต่างๆ ของวัตถุ ซึ่งจะมีการสร้างฟังก์ชันเชื่อมโยง ระหว่างคุณสมบัติของวัตถุกับประเภทของวัตถุจากตัวอย่างของข้อมูลหรือเรียกว่า Training set แล้วจึงใช้ฟังก์ชันนี้ทำนายประเภทของวัตถุที่ไม่เคยพบ เครื่องมือหรือขั้นตอนวิธีที่ใช้สำหรับการแบ่งประเภทข้อมูลเช่น โครงข่ายประสาทเทียม ต้นไม้ตัดสินใจ หรือเรียกอีกอย่างว่าเป็นการหากฎเพื่อระบุประเภทของวัตถุจากคุณสมบัติของวัตถุ เช่น หาความสัมพันธ์ระหว่างผลการตรวจร่างกายต่างๆ กับการเกิดโรค โดยใช้ข้อมูลผู้ป่วยและการวินิจฉัยของแพทย์ที่เก็บไว้ เพื่อนำมาช่วยวินิจฉัยโรคของผู้ป่วย หรือการวิจัยทางการแพทย์ ในทางธุรกิจจะใช้เพื่อดูคุณสมบัติของผู้ที่จะก่อหนี้ดีหรือหนี้เสีย เพื่อประกอบการพิจารณาการอนุมัติเงินกู้ เป็นต้น

3) การแบ่งกลุ่มข้อมูล (Data clustering) เป็นวิธีการวิเคราะห์ข้อมูลโดยจะแบ่งชุดข้อมูล (มักจะเป็นเวกเตอร์) ออกเป็นกลุ่ม (Cluster) นำข้อมูลที่มีคุณลักษณะเหมือนกัน หรือคล้ายกันจัดไว้ในกลุ่มเดียวกัน ขั้นตอนวิธีที่ใช้ในการแบ่งกลุ่มจะอาศัยความเหมือน (Similarity) หรือ ความใกล้ชิด (Proximity) โดยคำนวณจากการวัดระยะระหว่างเวกเตอร์ของข้อมูลเข้า โดยใช้การวัดระยะแบบต่างๆ เช่น การวัดระยะแบบยูคลิด (Euclidean distance) การวัดระยะแบบแมนฮัตตัน (Manhattan distance) การวัดระยะแบบเชบิเชฟ (Chebychev distance) การแบ่งกลุ่มข้อมูลจะแตกต่างจากการแบ่งประเภทข้อมูล (Classification) โดยจะแบ่งกลุ่มข้อมูลจากความคล้าย โดยไม่มีการกำหนดประเภทของข้อมูลไว้ก่อน จึงกล่าวได้ว่าการแบ่งกลุ่มข้อมูล เป็นการเรียนรู้แบบไม่มีผู้สอน (ไม่มี Training set) เช่น การแบ่งกลุ่มผู้ป่วยที่เป็นโรคเดียวกันตามลักษณะอาการ เพื่อนำไปใช้ประโยชน์ในการวิเคราะห์หาสาเหตุของโรค โดยพิจารณาจากผู้ป่วยที่มีอาการคล้ายคลึงกัน

4) การสร้างมโนภาพ (Visualization) เป็นการสร้างภาพคอมพิวเตอร์กราฟิกที่สามารถนำเสนอข้อมูลมากมายอย่างครบถ้วนแทนการใช้ข้อความนำเสนอข้อมูลที่มากมาย เราอาจพบข้อมูลที่ซ่อนเร้นเมื่อดูข้อมูลชุดนั้นด้วยจินตทัศน์

2.2 เวก้า (Weka)

ซอฟต์แวร์ Weka (Waikato Environment for Knowledge Analysis) เริ่มพัฒนาขึ้นในปี 1997 โดยมหาวิทยาลัย Waikato ประเทศนิวซีแลนด์ เป็นซอฟต์แวร์ Open Source ที่อยู่ภายใต้ GNU General

Public License โดยซอฟต์แวร์เวก้าถูกพัฒนามาจากภาษาจาวาทั้งหมด ซึ่งพัฒนาขึ้นเพื่อใช้ในงาน Machine learning และ Data mining โดยเฉพาะ สามารถใช้งานได้บนหลายระบบปฏิบัติการ ได้แก่ Windows, Mac OSX และ Linux เป็นต้น ซอฟต์แวร์เวก้าเป็นเครื่องมือที่ทำงานในด้านเหมืองข้อมูลที่รวบรวมอัลกอริทึมไว้มากมาย ซึ่ง อัลกอริทึมสามารถเลือกใช้งานได้โดยตรงจากอัลกอริทึมที่มีมาให้ หรือจากอัลกอริทึมที่เขียนเพิ่มเป็นชุดเครื่องมือเพิ่มเติม

2.2.1 ความสามารถของซอฟต์แวร์เวก้า

หน้าต่างของซอฟต์แวร์เวก้าประกอบด้วย Panel ต่างๆ ซึ่งช่วยสนับสนุนการทำเหมืองข้อมูล ได้แก่

1) Preprocess panel เป็น Panel สำหรับทำการเตรียมข้อมูลก่อนจะนำไปทำ Data mining โดยผู้ใช้งานจำเป็นที่จะต้องทำการ Add data ที่จะทำ Data mining เข้ามาก่อนใน Panel นี้ จึงจะสามารถนำไปดำเนินการต่างๆ ต่อไปได้
2) Classify panel เป็น Panel ที่ใช้ในการจำแนกประเภทของข้อมูล (Classification) และการทำนายค่าของข้อมูลใหม่โดยอาศัยข้อมูลเก่า นอกจากนี้ยังสามารถเลือกให้แสดงข้อมูลอื่นๆ เพิ่มเติมหรือเลือกไม่ให้แสดงข้อมูลตามที่เราต้องการได้ โดยการเลือก More option… และปรับเปลี่ยนตามที่เราต้องการ
3) Cluster panel เป็น Panel สำหรับการแบ่งกลุ่มข้อมูลจากความคล้ายคลึง (Similarity) ผู้ใช้สามารถกำหนดหรือปรับค่าใดๆ ตามฟังก์ชันการทำ Clustering กับชุดข้อมูลปัจจุบันได้
4) Associate panel เป็น Panel ที่ใช้สำหรับหารูปแบบของข้อมูลที่เกิดร่วมกันบ่อยๆ โดยผู้ใช้สามารถหากฎความสัมพันธ์ของข้อมูลผ่านการใช้ Weka associator
5) Select attributes panel เป็น Panel ที่ใช้คัดเลือกแอตทริบิวต์ที่สำคัญ ซึ่ง Panel นี้จะช่วยให้ผู้ใช้สามารถกำหนดและปรับค่าต่างๆ โดยใช้ตัวช่วยประเมินผลและเครื่องมือในการค้นหาของเวก้าในการเลือกแอตทริบิวต์ที่สำคัญของชุดข้อมูลนั้น
6) Visualize panel เป็น Panel ที่แสดงการพล็อตเมทริกซ์แบบกระจาย (Scatter plot matrix) ของชุดข้อมูลปัจจุบัน โดยขนาดของแต่ละเซลล์และขนาดของจุดบนพื้นที่กราฟสามารถปรับได้ตามความต้องการของผู้ใช้ที่ด้านล่างของหน้าต่าง โดย Panel นี้สามารถแสดงผลจากใช้ Panel ทุกๆ Panel ที่กล่าวมาข้างต้น

2.2.2 ข้อดีของซอฟต์แวร์เวก้า

- เป็นซอฟต์แวร์ที่สามารถดาวน์โหลดได้ฟรี
- สามารถทำงานได้บนหลายระบบปฏิบัติการ

- มีการเตรียมข้อมูลและเทคนิคในการสร้างแบบจำลองที่ครอบคลุม
- มี User interface ที่ง่ายต่อการใช้งาน
- มีอัลกอริทึมที่เป็นที่นิยมใช้ในการทำเหมืองข้อมูลให้เลือกใช้อย่างครบถ้วนและสามารถเขียนฟังก์ชันเพิ่มเข้าไปในซอฟต์แวร์เองได้

2.3 ราปิดมายเนอร์ (RapidMiner)

ราปิดมายเนอร์ก่อตั้งขึ้นที่ประเทศเยอรมนีในปี ค.ศ. 2006 โดยใช้ชื่อ Rapid-I ต่อมาในปี ค.ศ. 2013 ได้เปลี่ยนชื่อบริษัทเป็น RapidMiner และย้ายบริษัทมาอยู่ที่บอสตัน ประเทศสหรัฐอเมริกา โดยมีผลิตภัณฑ์หลักคือ RapidMiner Studio โดยราปิดมายเนอร์ได้รับการจัดอันดับให้อยู่ในกลุ่ม Leader สำหรับซอฟต์แวร์ที่ใช้ในการวิเคราะห์ข้อมูลโดย Gartner ราปิดมายเนอร์สามารถใช้งานได้บนหลายระบบปฏิบัติการ ได้แก่ Windows, Mac OSX และ Linux เป็นต้น นอกจากนี้ยังมีอัลกอริทึมให้เลือกมากมายและครบถ้วนตามอัลกอริทึมที่ได้รับความนิยมในการทำเหมืองข้อมูล และยังสามารถเพิ่มคุณสมบัติพิเศษที่เรียกว่า RapidMiner Extensions ได้ตามความต้องการ

2.3.1 ความสามารถของซอฟต์แวร์ราปิดมายเนอร์

ราปิดมายเนอร์เป็นซอฟต์แวร์ที่ให้บริการในด้านการวิเคราะห์ขั้นสูง รวมทั้งการวิเคราะห์การทำนาย, การทำเหมืองข้อมูลและการทำ Text mining ซึ่งราปิดมายเนอร์จะมีความสามารถในการวิเคราะห์ข้อมูลจากฐานข้อมูลหรือจากข้อความที่มีขนาดใหญ่ได้ โดยราปิดมายเนอร์จะเหมาะสำหรับผู้ใช้ทุกคนเนื่องจากใช้งานง่าย และไม่จำเป็นต้องเขียนโค้ดซอฟต์แวร์ใดๆ ราปิดมายเนอร์มี User Interface (UI) เป็นแบบกราฟิกซึ่งมีประสิทธิภาพและเหมาะกับกระบวนการวิเคราะห์ หรือหากผู้ใช้งานไม่ถนัดการใช้งานในโหมดกราฟิก สามารถเลือกใช้งานในโหมด Batch แทนได้ ชุดข้อมูลที่ราปิดมายเนอร์ยอมรับมีหลากหลายสกุลไฟล์ โดยค่าพื้นฐานจะเป็นไฟล์ .CSV (Comma Separated Value) แต่เนื่องจากในการทำโครงงานนี้เป็นการใช้ไฟล์ .ARFF ในการทดลอง จึงจำเป็นต้องทำการลงทะเบียนอีเมล์เพื่อขอ License ของเวอร์ชันที่มีความสามารถมากขึ้นจึงใช้งานได้ตามปกติ สำหรับความสามารถของราปิดมายเนอร์นั้น ได้ครอบคลุมเทคนิคต่างๆ ของการทำเหมืองข้อมูลไว้อย่างครบถ้วน ได้แก่

1) Association rules เป็นการวิเคราะห์เพื่อหากฎความสัมพันธ์ของข้อมูล
2) Clustering เป็นการแบ่งกลุ่มข้อมูล ได้แก่ การแบ่งข้อมูลที่มีลักษณะคล้ายๆ กันให้อยู่กลุ่มเดียวกัน และแบ่งข้อมูลที่มีลักษณะแตกต่างกันมากๆ ให้อยู่คนละกลุ่มกัน โดยแต่ละกลุ่มจะเรียกว่า คลัสเตอร์ (Cluster) ในการแบ่งกลุ่มข้อมูลนั้น ทำโดยการจัดข้อมูลให้อยู่ในกลุ่มต่างๆ โดยมีการวัดค่าความคล้ายคลึง (Similarity) หรือค่า

ระยะห่าง (Distance) ระหว่างข้อมูลแต่ละตัว โดยค่าระยะห่างที่นิยมใช้ เช่น ระยะห่างยูคลิเดียน (Euclidean distance) เป็นต้น อัลกอริทึมที่นิยมใช้ในการทำ Clustering ได้แก่ K-Means เป็นต้น

3) Classification สำหรับการจำแนกข้อมูล ซึ่งมีอัลกอริทึมที่เป็นที่นิยมในการทำ Classification ไว้อย่างครบถ้วน เช่น Decision tree, Naïve Bayes, AdaBoost, K-Nearest Neighbors เป็นต้น นอกจากนี้ยังมีลักษณะการ Test ให้เลือกใช้อย่างครบถ้วน เช่น การใช้ Supplied test set หรือ Cross-validation เป็นต้น

2.3.2 ข้อดีของซอฟต์แวร์ราปิดมายเนอร์

- เป็นซอฟต์แวร์ที่สามารถดาวน์โหลดได้ฟรีในเวอร์ชัน Starter ซึ่งมีความครอบคลุมในการใช้งาน
- สามารถทำงานได้บนหลายระบบปฏิบัติการ
- มีการเตรียมข้อมูลและเทคนิคในการสร้างแบบจำลองที่ครอบคลุม
- มี User interface แบบกราฟิกโหมด ซึ่งง่ายต่อการใช้งานและสามารถเลือกใช้เป็น Batch mode ก็ได้
- มีอัลกอริทึมที่เป็นที่นิยมใช้ในการทำเหมืองข้อมูลให้เลือกใช้อย่างครบถ้วนและสามารถเขียนฟังก์ชันเพิ่มเข้าไปในซอฟต์แวร์เองได้
- ครอบคลุมความสามารถทั้งหมดที่เวก้ามี

2.4 อาร์ ดาต้ามายนิ่ง (R Data Mining)

อาร์เป็นซอฟต์แวร์ที่ให้บริการแบบไม่เสียค่าใช้จ่าย โดยใช้ในงานด้านการคำนวณทางด้านสถิติศาสตร์และการแสดงผลกราฟิกด้านสถิติศาสตร์ด้วย อาร์สามารถรองรับการใช้งานได้กว่า 6,000 แพ็กเกจในปัจจุบันโดยการดาวน์โหลดผ่าน CRAN และนอกจากนี้ยังมีแมนนวลให้อ่านทำความเข้าใจกับทุกๆ แพ็กเกจ ในปัจจุบันอาร์เป็นที่นิยมทั้งในการด้านการศึกษาและอุตสาหกรรม ซึ่งอาร์ได้รับการโหวตให้เป็นอันดับ 1 บน KDnuggets 2014 สำหรับการโหวตภาษาที่ได้รับความนิยมสูงสุดในกระบวนการวิเคราะห์ข้อมูลและการทำเหมืองข้อมูล ซึ่งเป็นอันดับ 1 ในปี 2011, 2012 และ 2013 ด้วย การใช้งานของอาร์เป็นการเขียนคำสั่งเรียกใช้แพ็กเกจได้ตามที่ผู้ใช้ต้องการ รวมไปถึงการเรียกใช้คำสั่งแสดงผลเป็นกราฟ หรือเป็นแผนผังต้นไม้ ซึ่งอาร์จะค่อนข้างเข้าใจยากสำหรับผู้เริ่มต้นใช้งาน แต่เนื่องจากความสามารถของอาร์มีความครอบคลุมในทุกๆ ด้านของการทำเหมืองข้อมูล และมีทุกอัลกอริทึมในการทำเหมืองข้อมูล ทำให้อาร์เป็นที่นิยมอย่างมากในปัจจุบัน อาร์จะ

เปิดโอกาสให้ผู้ใช้พัฒนาแพ็กเกจของอาร์และแบ่งปันให้กับผู้ใช้คนอื่นๆ ได้อย่างสะดวก ทำให้อาร์ได้รับความนิยมในหลากหลายประเทศ

2.4.1 ความสามารถของซอฟต์แวร์อาร์ ดาต้ามายนิ่ง

อาร์มีความสามารถในการทำเหมืองข้อมูลในทุกๆ เทคนิค ได้แก่

1) Classification การจำแนกข้อมูล ผู้ใช้สามารถ Install แพ็กเกจเพิ่มเติมได้เพื่อเรียกใช้อัลกอริทึมที่ต้องการในการทำ Classification โดยอาร์มีอัลกอริทึมที่ครอบคลุมการจำแนกข้อมูลทุกอัลกอริทึม ตัวอย่างเช่น หากผู้ใช้ต้องการใช้อัลกอริทึม Decision tree สามารถ Install แพ็กเกจ Party ซึ่งเป็นแพ็กเกจที่มีฟังก์ชันของอัลกอริทึม Decision tree จากนั้นผู้ใช้สามารถศึกษาการใช้งานฟังก์ชันต่างๆ ในแพ็กเกจ Party ได้โดยศึกษาจากหน้าเว็บไซต์ของ CRAN ซึ่งมีรายละเอียดและตัวอย่างการใช้ฟังก์ชันอย่างครบถ้วนและเข้าใจง่าย
2) Clustering การแบ่งกลุ่มข้อมูล ผู้ใช้สามารถทำการ Install แพ็กเกจที่ใช้สำหรับแบ่งข้อมูลได้ตามต้องการ โดยอาร์มีอัลกอริทึมที่ใช้สำหรับแบ่งกลุ่มข้อมูลให้เลือกใช้อย่างครบถ้วน ซึ่งอัลกอริทึมที่เป็นที่นิยมได้แก่ K-means และ K-medoids เป็นต้น
3) Association rule การทำเหมืองข้อมูลโดยการหากฎความสัมพันธ์ โดยอาร์จะประกอบไปด้วยอัลกอริทึมสำหรับหากฎความสัมพันธ์ที่ได้รับความนิยมอย่างครบถ้วน เช่น Apriori() และ Eclat()

2.4.2 ข้อดีของซอฟต์แวร์อาร์ ดาต้ามายนิ่ง

- เป็นซอฟต์แวร์ที่สามารถดาวน์โหลดได้ฟรี และสามารถดาวน์โหลดแพ็กเกจเพิ่มเติมตามความต้องการได้ฟรีเช่นกัน
- สามารถทำงานได้บนหลายระบบปฏิบัติการ
- มีแมนนวลอธิบายในการใช้งานทุกๆ แพ็กเกจ
- มีอัลกอริทึมที่เป็นที่นิยมใช้ในการทำเหมืองข้อมูลให้เลือกใช้อย่างครบถ้วนและสามารถเขียนแพ็กเกจเพิ่มเติมเองได้ รวมถึงสามารถแบ่งปันแพ็กเกจที่เขียนขึ้นเองให้แก่ผู้ใช้คนอื่นได้
- ครอบคลุมความสามารถทั้งหมดที่เวก้ามี

2.5 อัลกอริทึมนาอีฟเบย์ (Naïve Bayes Algorithm)

Naïve Bayes classifiers เป็นอัลกอริทึมในกลุ่มของ Probabilistic classifiers หรือจัดว่าเป็นโมเดลการคัดแยกประเภทข้อมูลที่ใช้หลักความน่าจะเป็นซึ่งอยู่บนพื้นฐานของ Bayes' theorem และสมมติฐานที่ให้การเกิดของเหตุการณ์ต่างๆ เป็นอิสระต่อกัน (Independence)

Naïve Bayes เป็นเทคนิคอย่างง่ายในการจำแนกข้อมูล ซึ่งไม่ได้เป็นอัลกอริทึมสำหรับสร้างโมเดลจาก Training set เท่านั้น แต่เป็นอัลกอริทึมที่จะสมมติค่าของตัวแปรอิสระขึ้นมาและคำนวณหาคลาสให้กับตัวแปรนั้น โดยข้อดีที่สำคัญของ Naïve Bayes คือต้องการข้อมูลที่เป็น Training set ไม่มาก ก็สามารถทำการสร้างโมเดลสำหรับจำแนกข้อมูล (Classification) ได้อย่างแม่นยำ

สมการของ Bayes theorem ซึ่งมีพื้นฐานมาจาก Probabilistic model มีลักษณะดังนี้

$$P(C_k|x) = \frac{\mathrm{P}(C_k)\times \mathrm{P}(x|C_k)}{\mathrm{P}(x)} \qquad (2.1)$$

โดยสมการของ Bayes จะมี 3 ส่วนที่สำคัญคือ

- Posterior probability หรือ $P(C_k|x)$ คือ ความน่าจะเป็นที่ข้อมูลที่มีแอตทริบิวต์เป็น x จะมีคลาสเป็น C_k
- Likelihood หรือ $P(x|C_k)$ คือ ความน่าจะเป็นที่ข้อมูล Training set ที่มีคลาส C_k และมีแอตทริบิวต์ x โดยที่ $x = x_1 \cap x_2 \ldots \cap x_n$ โดยที่ n คือจำนวนแอตทริบิวต์ใน Training set
- Prior probability หรือ $P(C_k)$ คือความน่าจะเป็นของคลาส C_k

แต่เนื่องจากการที่แอตทริบิวต์ $x = x_1 \cap x_2 \ldots \cap x_n$ ที่เกิดขึ้นใน Training set อาจจะมีจำนวนน้อยมากหรือไม่มีรูปแบบของแอตทริบิวต์แบบนี้เกิดขึ้นเลย ดังนั้นจึงได้ใช้หลักการที่ว่าแต่ละแอตทริบิวต์เป็น Independent ต่อกันทำให้สามารถเปลี่ยนสมการของ $P(x|C_k)$ ได้เป็น

$$P(x|C_k) = P(x_1|C_k) \times P(x_2|C_k) \ldots P(x_n|C_k) \qquad (2.2)$$

โดยในการสร้างโมเดล Naïve Bayes สามารถนำข้อมูล Training set มาคำนวณตามสมการข้างต้นจะเกิดเป็นโมเดลที่มีประสิทธิภาพขึ้น

2.6 อัลกอริทึมต้นไม้ตัดสินใจ (Decision tree Algorithm)

Decision tree algorithm หรือเรียกอีกอย่างว่า Decision tree learning เป็นการใช้แผนภาพต้นไม้เป็นโมเดลสำหรับทำนาย ซึ่งเป็น Classification tree ชนิดหนึ่ง โดยแผนภาพต้นไม้นี้ ใบ (Leaves) จะแสดงคลาสและกิ่งก้าน (Branches) จะแสดงการเชื่อมต่อของคลาสเหล่านั้น Decision tree เป็นที่นิยมอย่างมากในการทำเหมืองข้อมูล โดยมีวัตถุประสงค์คือการสร้างโมเดลที่สามารถทำนายค่าของตัวแปรเป้าหมายโดยมีพื้นฐานจากหลายๆ ตัวแปรก่อนหน้านี้ การเรียนรู้ของ Decision tree เป็นการเรียนรู้โดยการจำแนกข้อมูล (Classification) โดยการจำแนกออกเป็นคลาสต่างๆ โดยใช้คุณลักษณะ (Attribute) ของข้อมูลในการจำแนก

ส่วนประกอบของผลลัพธ์ของ Decision tree

- โนดภายใน (Internal node) คือคุณลักษณะต่างๆ ของข้อมูลซึ่งเมื่อข้อมูลใดๆ ตกลงมาที่โนด จะใช้คุณลักษณะนี้เป็นตัวตัดสินใจว่าข้อมูลจะไปในทิศทางใด โดยโนดภายในที่เป็นจุดเริ่มต้นของต้นไม้นี้จะเรียกว่า ราก (Root)
- กิ่ง (Branch) เป็นค่าของคุณลักษณะในโนดภายในที่แตกกิ่งนี้ออกมา
- โนดใบ (Leaf node) คือกลุ่มต่างๆ ซึ่งเป็นผลลัพธ์ในการจำแนกประเภทข้อมูล

ขั้นตอนการสร้าง Decision tree จาก Training set เพื่อใช้จำแนกข้อมูล ทำได้ดังนี้

1) เลือกแอตทริบิวต์ที่ทำหน้าที่เป็น Root node โดยเกณฑ์ที่ช่วยตัดสินใจในการเลือก Root node คือทดลองเลือกแอตทริบิวต์แต่ละตัวมาทำหน้าที่เป็น Root node แล้วหาค่า Gain ซึ่งเป็นค่าที่ใช้บอกว่าแอตทริบิวต์ที่ทำหน้าที่เป็น Root node สามารถจำแนกข้อมูลได้ดีมากน้อยเพียงใด โดยจะทำการเลือกแอตทริบิวต์ที่มีค่า Gain สูงที่สุดเป็น Root node โดยการคำนวณหาค่า Gain ทำได้จากสมการนี้

$$Gain(x) = info(T) - info_x(T) \quad (2.3)$$

T คือ เซตของ Training set

X คือ แอตทริบิวต์ที่ถูกเลือกให้เป็นตัวจำแนกข้อมูล

Info (T) คือ ฟังก์ชันที่ระบุปริมาณข้อมูลที่ต้องการเพื่อให้สามารถจำแนกคลาสที่ต้องการได้

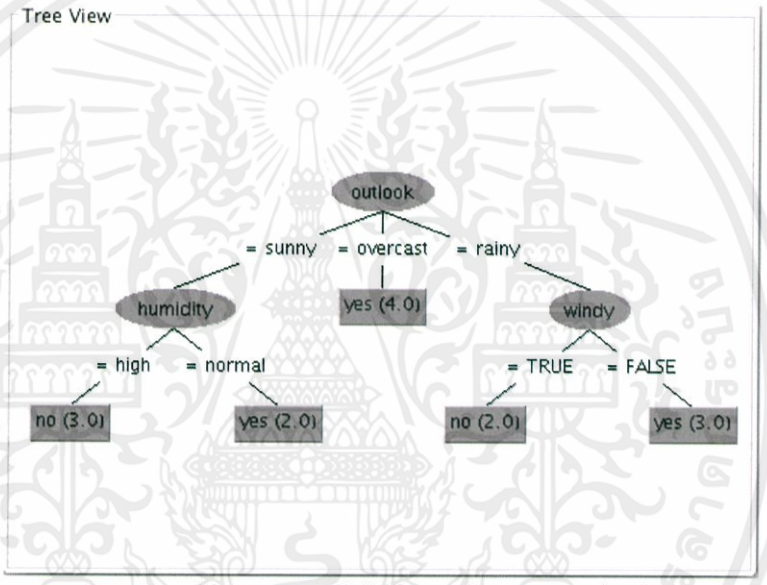
$Info_x$ (T) คือ ฟังก์ชันที่ระบุปริมาณข้อมูลที่ต้องการเพื่อการจำแนกคลาสของข้อมูลโดยใช้แอตทริบิวต์ X เป็นตัวตรวจสอบเพื่อแยกข้อมูล

โดยค่า Info (T) สามารถเรียกได้อีกอย่างหนึ่งว่า Entropy มีสมการการคำนวณดังนี้

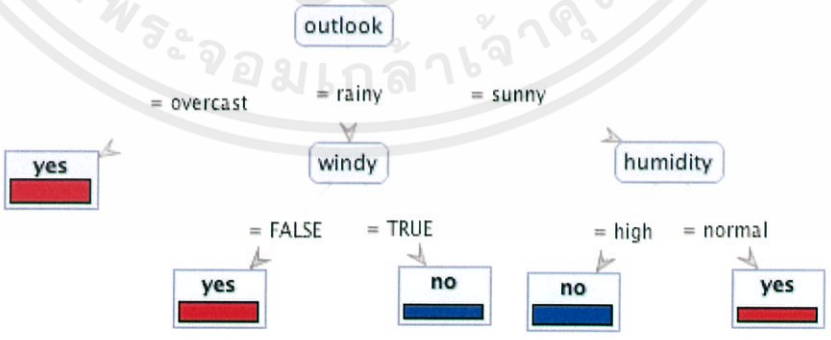
$$entropy(P_1, P_2, \dots, P_n) = -P_1 log P_1 - P_2 log P_2 - \cdots - P_n log P_n \quad (2.4)$$

2) หลังจากได้แอตทริบิวต์ที่เป็น Root node จะทำการเลือกแอตทริบิวต์ที่เหลือมาเป็นโนดต่อๆ ไป โดยทำการคำนวณด้วยสมการเดิม ซึ่งแต่ละกิ่งจะได้ค่าโนดถัดไปเป็นแอตทริบิวต์คนละตัว ทำเช่นนี้ไปเรื่อยๆ

3) กระบวนการสร้าง Decision tree จะสิ้นสุดเมื่อ Leaf node เป็นกลุ่มของข้อมูลคลาสเดียวกันทั้งหมด



รูป 2.1 ตัวอย่างแผนภาพต้นไม้ตัดสินใจจากซอฟต์แวร์เวก้า



รูป 2.2 ตัวอย่างแผนภาพต้นไม้ตัดสินใจจากซอฟต์แวร์ราปิดมายเนอร์

2.7 อัลกอริทึมอดาบูส (AdaBoost Algorithm)

วิธีการอดาบูสมีหลักการคือการปรับค่าน้ำหนักของชุดข้อมูล Training set ที่เรียนรู้ได้ยากหรือจำแนกประเภทข้อมูลไม่ถูกต้อง โดยการสร้างตัวจำแนกขึ้นมาในแต่ละรอบ ดังนี้

1) กำหนดให้ S_i คือตัวจำแนกประเภทข้อมูล โดยที่ $i = \{0, 1, 2, \ldots, n\}$
2) เริ่มต้นด้วยการสร้างตัวจำแนก S_0 จากข้อมูล Training set ตัวจำแนกนี้อาจไม่มีความถูกต้องมากก็ได้ สิ่งที่นำมาพิจารณาเพิ่มเติมจากตัวจำแนก S_0 คือข้อมูลใดบ้างใน Training set ที่ตัวจำแนก S_0 จำแนกข้อมูลไม่ถูกต้อง
3) สร้างตัวจำแนก S_1 ขึ้นมา โดยเพิ่มค่าน้ำหนักให้กับข้อมูลที่จำแนกประเภทไม่ถูกต้องด้วยตัวจำแนก S_0 เมื่อสร้างตัวจำแนก S_1 แล้วก็ทำการเพิ่มค่าน้ำหนักให้กับชุดข้อมูล Training set ที่จำแนกประเภทข้อมูลไม่ถูกต้องอีกครั้งและสร้างเป็นตัวจำแนก S_2 ต่อไป
4) ทำซ้ำกระบวนการเดิมอีกครั้งจนถึงตัวจำแนก S_n จึงหยุดทำการเทรน

ซึ่งจะพบว่าวิธีการนี้ การสร้างตัวจำแนกในรอบหลังๆ จะมีเป้าหมายคือพยายามปรับค่าน้ำหนักของข้อมูล Training set ที่จำแนกประเภทข้อมูลไม่ถูกต้อง เมื่อนำตัวจำแนกทุกตัวตั้งแต่ S_0 จนกระทั่งถึง S_n ที่สร้างขึ้นมาใช้ร่วมกัน สิ่งที่คาดว่าจะได้รับคือได้ตัวจำแนกประเภทข้อมูลที่มีความถูกต้องมากกว่าการใช้ตัวจำแนกประเภทข้อมูลแค่ตัวใดตัวหนึ่ง โดยอดาบูสจะเข้าไปช่วยทำให้โมเดลที่เกิดจากการจำแนกด้วยอัลกอริทึมปกตินั้น มีประสิทธิภาพมากขึ้น และทำให้การจำแนกมีความถูกต้องมากยิ่งขึ้น

2.8 อัลกอริทึมซีโร่อาร์ (ZeroR Algorithm)

ซีโร่อาร์เป็นวิธีการที่ง่ายที่สุดในการจำแนกข้อมูล โดยอาศัยการสนใจแค่เป้าหมายและไม่สนใจการทำนายอื่นๆ ซีโร่อาร์จะสร้างโมเดลจากการสนใจเพียงคลาสหลักของชุดข้อมูล แม้ซีโร่อาร์จะไม่มีการสร้างโมเดลทำนายจากการสนใจแอตทริบิวต์ทั้งหมด แต่ก็เป็นประโยชน์สำหรับการตั้งเป็นมาตรฐาน (Baseline) สำหรับอัลกอริทึมอื่นๆ ในการสร้างโมเดลสำหรับทำนาย

2.9 อัลกอริทึมดิซีชั่นสตัมพ์ (DecisionStump Algorithm)

ดิซีชั่นสตัมพ์เป็น Machine learning model ที่ประกอบด้วยต้นไม้ตัดสินใจแบบ One-level คือเป็นต้นไม้ตัดสินใจที่มี 1 Internal node (Root) ซึ่งเชื่อมต่อไปยังใบต่างๆ ดิซีชั่นสตัมพ์จะทำให้เกิดการทำนายแค่ค่าเดียวเท่านั้น สามารถเรียกอีกอย่างหนึ่งว่า 1-rules ดิซีชั่นสตัมพ์มักถูกใช้เป็นส่วนประกอบหรือที่เรียกว่า Weak learning หรือ Base learner ในการทำ Machine learning ประเภท Boosting และ Bagging เป็นต้น

2.10 อัลกอริทึมวันอาร์ (OneR Algorithm)

OneR ย่อมาจาก "One Rule" ซึ่งแปลว่าง่ายและถูกต้อง วันอาร์เป็นอัลกอริทึมที่ใช้สำหรับจำแนกข้อมูลโดยสร้างกฎสำหรับทำนายข้อมูลขึ้นมาหนึ่งกฎจากแต่ละข้อมูล จากนั้นเลือกกฎที่มีข้อผิดพลาดน้อยที่สุดมาใช้กับทั้งหมด ตัวอย่างเช่น

Which one is the best predictor ?

Outlook	Temp	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	High	True	No

Frequency Tables

★		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3

		Play Golf	
		Yes	No
Temp.	Hot	2	2
	Mild	4	2
	Cool	3	1

		Play Golf	
		Yes	No
Humidity	High	3	4
	Normal	6	1

		Play Golf	
		Yes	No
Windy	False	6	2
	True	3	3

รูป 2.3 ตัวอย่างการทำงานของอัลกอริทึมวันอาร์

โดยจากรูปสามารถอธิบายได้ดังนี้ จาก Frequency Tables ของ Outlook

ถ้า outlook = sunny แล้ว play = no จะเกิด 2 errors ใน 5 records

ถ้า outlook = overcast แล้ว play = yes จะเกิด 0 errors ใน 4 records

ถ้า outlook = rainy แล้ว play = yes จะเกิด 2 errors ใน 5 records

รวมแล้ว Outlook จะมี 4 errors จาก 14 records ซึ่งถือเป็นการเกิด Error น้อยที่สุด วันอาร์จึงเลือก Outlook เป็น Decisive attribute

2.11 รูปแบบไฟล์เออาร์เอฟเอฟ (ARFF Format)

ARFF ย่อมาจาก Attribute-Relation File Format ซึ่งเป็น ASCII text file ที่อธิบายข้อมูลของ Instance และแอตทริบิวต์ ARFF ถูกสร้างขึ้นโดยคณะวิทยาศาสตร์คอมพิวเตอร์ มหาวิทยาลัย Waikato เพื่อใช้งานกับซอฟต์แวร์เวก้า

ARFF จะแบ่งออกเป็น 2 ส่วน ได้แก่

1) ส่วน Header ซึ่งประกอบไปด้วยส่วนของรายละเอียดของชุดข้อมูล โดย Header จะมีคำอธิบายของชุดข้อมูล, รายการของแอตทริบิวต์ และ คลาส ตัวอย่างของส่วน Header เป็นดังนี้

ตัวอย่าง 2.1 ส่วน Header ของไฟล์ ARFF

```
% 1. Title: Iris Plants Database
% 2. Sources:
%      (a) Creator: R.A. Fisher
%      (b) Donor: Michael Marshall
%      (c) Date: July, 1988
@RELATION iris
@ATTRIBUTE sepallength    NUMERIC
@ATTRIBUTE sepalwidth     NUMERIC
@ATTRIBUTE petallength    NUMERIC
@ATTRIBUTE petalwidth     NUMERIC
@ATTRIBUTE class {Iris-setosa,Iris-versicolor,Iris-virginica}
```

จากตัวอย่างข้างต้น สามารถอธิบายได้ดังนี้

- บรรทัดที่ขึ้นต้นด้วย % จะเป็นการ Comment ไม่มีผลอะไรในชุดข้อมูล ส่วนใหญ่จะมีเอาไว้เพื่อบอกที่มาของชุดข้อมูล รวมถึงจำนวน Instance และอธิบายรายละเอียดของแต่ละแอตทริบิวต์
- บรรทัดที่ขึ้นต้นด้วย @RELATION จะบ่งบอกว่าเป็นบรรทัดแรกของไฟล์ ARFF นั้น
- บรรทัดที่ขึ้นต้นด้วย @ATTRIBUTE เป็นการแสดงรายการของแอตทริบิวต์ที่มีอยู่ในชุดข้อมูลนั้น โดยจะตามด้วยลักษณะข้อมูลของแอตทริบิวต์นั้น ในตัวอย่างจะเป็น NUMERIC ซึ่งเป็นตัวเลข แต่หากข้อมูลนั้นไม่ใช่ตัวเลขจะกำหนดเป็น {'value', 'value', 'value', ...} โดยจำนวนบรรทัดของการกำหนดค่าแอตทริบิวต์ จะมีจำนวนบรรทัดตามจำนวนของแอตทริบิวต์และคลาส

2) ส่วน Data เป็นส่วนที่แสดงข้อมูลของชุดข้อมูล ARFF โดยมีตัวอย่างดังนี้

ตัวอย่าง 2.2 ส่วน Data ของไฟล์ ARFF

```
@DATA
5.1,3.5,1.4,0.2,Iris-setosa
4.9,3.0,1.4,0.2,Iris-setosa
4.7,3.2,1.3,0.2,Iris-setosa
4.6,3.1,1.5,0.2,Iris-setosa
5.0,3.6,1.4,0.2,Iris-setosa
5.4,3.9,1.7,0.4,Iris-setosa
4.6,3.4,1.4,0.3,Iris-setosa
5.0,3.4,1.5,0.2,Iris-setosa
4.4,2.9,1.4,0.2,Iris-setosa
4.9,3.1,1.5,0.1,Iris-setosa
```

จากตัวอย่างข้างต้นในแต่ละบรรทัดจะถูกเรียกว่าหนึ่ง Instance โดยแต่ละคอมม่าจะเป็นค่าในแอตทริบิวต์ที่ได้กำหนดไว้ในส่วนของ Header และปิดท้ายด้วยคลาสของข้อมูลใน Instance นั้น

บทที่ 3
การวิเคราะห์และการออกแบบ

3.1 ชุดข้อมูลที่ใช้ในการวิเคราะห์

ในการทดลองศึกษาเกี่ยวกับกระบวนการทำเหมืองข้อมูลในประเด็นต่างๆ โดยใช้ซอฟต์แวร์เกี่ยวกับการทำเหมืองข้อมูล ได้แก่ Weka, RapidMiner และ R Data Mining หลังจากกระบวนการดาวน์โหลดและติดตั้งซอฟต์แวร์แต่ละตัวแล้ว จะพบว่ามีซอฟต์แวร์ Weka ที่มีการให้ชุดข้อมูลมาด้วยอยู่แล้ว และสามารถเปิดอ่านได้ด้วยโปรแกรมทั่วไป เช่น Notepad++ ซึ่งจะมีส่วนช่วยให้เราสามารถศึกษาเกี่ยวกับรายละเอียดของข้อมูลได้เข้าใจมากยิ่งขึ้น ซึ่งข้อมูลเหล่านี้จะอยู่ในรูปแบบของสกุลไฟล์ ARFF ซึ่งได้รับการรองรับในซอฟต์แวร์ทุกๆ ตัวที่เราจะทำการศึกษา โดยชุดข้อมูลที่เราทำการคัดเลือกนำมาใช้ในการทดลอง ประกอบด้วยชุดข้อมูลทั้งหมด 7 ชุด ซึ่งจะมีความแตกต่างกันทั้งในด้านขนาดของข้อมูล แอตทริบิวต์ และลักษณะของข้อมูล

3.1.1 ดอกไอริส (Iris.arff)

เป็นการเก็บข้อมูลความกว้าง ความยาว ของกลีบเลี้ยงและกลีบดอก เพื่อจำแนกว่าเป็นดอกไอริสชนิดใด โดยจะประกอบด้วยแอตทริบิวต์ทั้งหมด 5 แอตทริบิวต์ แต่ละแอตทริบิวต์จะประกอบด้วยค่าต่างๆ ดังนี้

- sepallength ความยาวกลีบเลี้ยง (cm) ประกอบด้วยค่าที่เป็นตัวเลข มีค่าอยู่ในช่วง 4.3-7.9
- sepalwidth ความกว้างกลีบเลี้ยง (cm) ประกอบด้วยค่าที่เป็นตัวเลข มีค่าอยู่ในช่วง 2.0-4.4
- petallength ความยาวกลีบดอก (cm) ประกอบด้วยค่าที่เป็นตัวเลข มีค่าอยู่ในช่วง 1.0-6.9
- petalwidth ความกว้างกลีบดอก (cm) ประกอบด้วยค่าที่เป็นตัวเลข มีค่าอยู่ในช่วง 0.1-2.5
- class จำแนกว่าเป็นดอกไอริสชนิดใด ประกอบด้วย 3 ประเภท ดังนี้ iris-setosa, iris-versicolor และ iris-virginica

ชุดข้อมูลดอกไอริสจะประกอบด้วยข้อมูลทั้งหมด 150 Instances

Relation: iris

No.	sepallength Numeric	sepalwidth Numeric	petallength Numeric	petalwidth Numeric	class Nominal
1	5.1	3.5	1.4	0.2	Iris-setosa
2	4.9	3.0	1.4	0.2	Iris-setosa
3	4.7	3.2	1.3	0.2	Iris-setosa
4	4.6	3.1	1.5	0.2	Iris-setosa
5	5.0	3.6	1.4	0.2	Iris-setosa
6	5.4	3.9	1.7	0.4	Iris-setosa
7	4.6	3.4	1.4	0.3	Iris-setosa
8	5.0	3.4	1.5	0.2	Iris-setosa
9	4.4	2.9	1.4	0.2	Iris-setosa
10	4.9	3.1	1.5	0.1	Iris-setosa
11	5.4	3.7	1.5	0.2	Iris-setosa
12	4.8	3.4	1.6	0.2	Iris-setosa
13	4.8	3.0	1.4	0.1	Iris-setosa
14	4.3	3.0	1.1	0.1	Iris-setosa
15	5.8	4.0	1.2	0.2	Iris-setosa
16	5.7	4.4	1.5	0.4	Iris-setosa
17	5.4	3.9	1.3	0.4	Iris-setosa
18	5.1	3.5	1.4	0.3	Iris-setosa
19	5.7	3.8	1.7	0.3	Iris-setosa
20	5.1	3.8	1.5	0.3	Iris-setosa
21	5.4	3.4	1.7	0.2	Iris-setosa
22	5.1	3.7	1.5	0.4	Iris-setosa
23	4.6	3.6	1.0	0.2	Iris-setosa
24	5.1	3.3	1.7	0.5	Iris-setosa
25	4.8	3.4	1.9	0.2	Iris-setosa
26	5.0	3.0	1.6	0.2	Iris-setosa
27	5.0	3.4	1.6	0.4	Iris-setosa
28	5.2	3.5	1.5	0.2	Iris-setosa
29	5.2	3.4	1.4	0.2	Iris-setosa
30	4.7	3.2	1.6	0.2	Iris-setosa

รูป 3.1 ตัวอย่างชุดข้อมูลดอกไอริส

3.1.2 กระจก (Glass.arff)

เป็นชุดข้อมูลเกี่ยวกับองค์ประกอบทางเคมีต่างๆ ที่ใช้ในการทำแก้วประเภทต่างๆ กัน โดยใช้ข้อมูลองค์ประกอบทางเคมีที่เรามีเพื่อทำนายวัสดุว่าวัสดุที่มีองค์ประกอบแบบนี้เป็นกระจกประเภทใด โดยจะประกอบด้วยแอตทริบิวต์ทั้งหมด 10 แอตทริบิวต์ แต่ละแอตทริบิวต์จะประกอบด้วยค่าต่างๆ ดังนี้

- RI ค่าการหักเห ประกอบด้วยค่าที่เป็นตัวเลขมีค่าอยู่ในช่วง 1.511-1.534
- Na โซเดียม (หน่วยวัดเป็นร้อยละของน้ำหนักที่ใช้ในการออกไซด์) ประกอบด้วยค่าที่เป็นตัวเลข มีค่าอยู่ในช่วง 10.73-17.38
- Mg แมกนีเซียม (หน่วยวัดเป็นร้อยละของน้ำหนักที่ใช้ในการออกไซด์) ประกอบด้วยค่าที่เป็นตัวเลข มีค่าอยู่ในช่วง 0.0-4.49
- Al อลูมิเนียม (หน่วยวัดเป็นร้อยละของน้ำหนักที่ใช้ในการออกไซด์) ประกอบด้วยค่าที่เป็นตัวเลข มีค่าอยู่ในช่วง 0.29-3.5
- Si ซิลิกอน (หน่วยวัดเป็นร้อยละของน้ำหนักที่ใช้ในการออกไซด์) ประกอบด้วยค่าที่เป็นตัวเลข มีค่าอยู่ในช่วง 69.81-75.41
- K โพแทสเซียม (หน่วยวัดเป็นร้อยละของน้ำหนักที่ใช้ในการออกไซด์) ประกอบด้วยค่าที่เป็นตัวเลข มีค่าอยู่ในช่วง 0.0-6.21
- Ca แคลเซียม (หน่วยวัดเป็นร้อยละของน้ำหนักที่ใช้ในการออกไซด์) ประกอบด้วยค่าที่เป็นตัวเลข มีค่าอยู่ในช่วง 5.43-16.19

- Ba แบเรียม (หน่วยวัดเป็นร้อยละของน้ำหนักที่ใช้ในการออกไซด์) ประกอบด้วยค่าที่เป็นตัวเลข มีค่าอยู่ในช่วง 0.0-3.15
- Fe เหล็ก (หน่วยวัดเป็นร้อยละของน้ำหนักที่ใช้ในการออกไซด์) ประกอบด้วยค่าที่เป็นตัวเลข มีค่าอยู่ในช่วง 0.0-0.51
- Type ชนิดของกระจก แบ่งเป็นทั้งหมด 7 ชนิด ได้แก่ build wind float, build wind non-float, vehic wind float, vehic wind non-float, containers, tableware และheadlamps
- ชุดข้อมูลกระจกจะประกอบด้วยข้อมูลทั้งหมด 214 Instances

Relation: Glass

No.	RI Numeric	Na Numeric	Mg Numeric	Al Numeric	Si Numeric	K Numeric	Ca Numeric	Ba Numeric	Fe Numeric	Type Nominal
1	1.51793	12.79	3.5	1.12	73.03	0.64	8.77	0.0	0.0	build wind float
2	1.51643	12.16	3.52	1.35	72.89	0.57	8.53	0.0	0.0	vehic wind float
3	1.51793	13.21	3.48	1.41	72.64	0.59	8.43	0.0	0.0	build wind float
4	1.51299	14.4	1.74	1.54	74.55	0.0	7.59	0.0	0.0	tableware
5	1.53393	12.3	0.0	1.0	70.16	0.12	16.19	0.0	0.24	build wind non-float
6	1.51655	12.75	2.85	1.44	73.27	0.57	8.79	0.11	0.22	build wind non-float
7	1.51779	13.64	3.65	0.65	73.0	0.06	8.93	0.0	0.0	vehic wind float
8	1.51837	13.14	2.84	1.28	72.85	0.55	9.07	0.0	0.0	build wind float
9	1.51545	14.14	0.0	2.68	73.39	0.08	9.07	0.61	0.05	headlamps
10	1.51789	13.19	3.9	1.3	72.33	0.55	8.44	0.0	0.28	build wind non-float
11	1.51625	13.36	3.58	1.49	72.72	0.45	8.21	0.0	0.0	build wind non-float
12	1.51743	12.2	3.25	1.16	73.55	0.62	8.9	0.0	0.24	build wind non-float
13	1.52223	13.21	3.77	0.79	71.99	0.13	10.02	0.0	0.0	build wind float
14	1.52121	14.03	3.76	0.58	71.79	0.11	9.65	0.0	0.0	vehic wind float
15	1.51665	13.14	3.45	1.76	72.48	0.6	8.38	0.0	0.17	vehic wind float
16	1.51707	13.48	3.48	1.71	72.52	0.62	7.99	0.0	0.0	build wind non-float
17	1.51719	14.75	0.0	2.0	73.02	0.0	8.53	1.59	0.08	headlamps
18	1.51629	12.71	3.33	1.49	73.28	0.67	8.24	0.0	0.0	build wind non-float
19	1.51994	13.27	0.0	1.76	73.03	0.47	11.32	0.0	0.0	containers
20	1.51811	12.96	2.96	1.43	72.92	0.6	8.79	0.14	0.0	build wind non-float
21	1.52152	13.05	3.65	0.87	72.22	0.19	9.85	0.0	0.17	build wind float
22	1.52475	11.45	0.0	1.88	72.19	0.81	13.24	0.0	0.34	build wind non-float
23	1.51841	12.93	3.74	1.11	72.28	0.64	8.96	0.0	0.22	build wind non-float
24	1.51754	13.39	3.66	1.19	72.79	0.57	8.27	0.0	0.11	build wind float

รูป 3.2 ตัวอย่างชุดข้อมูลกระจก

3.1.3 มะเร็งเต้านม (Breast-cancer.arff)

เป็นชุดข้อมูลทางการแพทย์เกี่ยวกับสุขภาพของคนไข้ เพื่อทำนายว่าจะเป็นมะเร็งเต้านมหรือไม่ โดยจะประกอบด้วยแอตทริบิวต์ทั้งหมด 10 แอตทริบิวต์ แต่ละแอตทริบิวต์จะประกอบด้วยค่าต่างๆ ดังนี้

- age อายุ แบ่งเป็นช่วงอายุ 10-19, 20-29, 30-39, 40-49, 50-59, 60-69, 70-79, 80-89 และ 90-99
- menopause ประกอบด้วยค่า lt40, ge40 และ premeno
- tumor-size ประกอบด้วยช่วง 0-4, 5-9, 10-14, 15-19, 20-24, 25-29, 30-34, 35-39, 40-44, 45-49, 50-54 และ 55-59

- inv-nodes ประกอบด้วยช่วง 0-2, 3-5, 6-8, 9-11, 12-14, 15-17, 18-20, 21-23, 24-26, 27-29, 30-32, 33-35 และ 36-39
- node-caps ประกอบด้วยค่า yes และ no
- deg-malig ประกอบด้วยค่า 1, 2 และ 3
- breast ประกอบด้วยค่า left และ right
- breast-quad ประกอบด้วยค่า left_up, left_low, right_up, right_low และ central
- irradiat ประกอบด้วยค่า yes และ no
- Class แบ่งเป็น 2 ประเภท คือ ประเภทที่เป็นมะเร็งเต้านม (recurrence-events) และไม่เป็นมะเร็งเต้านม (no-recurrence-events)

ชุดข้อมูลมะเร็งเต้านมจะประกอบด้วยข้อมูลทั้งหมด 286 Instances ซึ่งในบาง Instance จะมีค่าในบางแอตทริบิวต์ที่ไม่มีข้อมูล

Relation: breast-cancer

No.	age Nominal	menopause Nominal	tumor-size Nominal	inv-nodes Nominal	node-caps Nominal	deg-malig Nominal	breast Nominal	breast-quad Nominal	irradiat Nominal	Class Nominal
1	40-49	premeno	15-19	0-2	yes	3	right	left_up	no	recurrence-events
2	50-59	ge40	15-19	0-2	no	1	right	central	no	no-recurrence-events
3	50-59	ge40	35-39	0-2	no	2	left	left_low	no	recurrence-events
4	40-49	premeno	35-39	0-2	yes	3	right	left_low	yes	no-recurrence-events
5	40-49	premeno	30-34	3-5	yes	2	left	right_up	no	recurrence-events
6	50-59	premeno	25-29	3-5	no	2	right	left_up	yes	no-recurrence-events
7	50-59	ge40	40-44	0-2	no	3	left	left_up	no	no-recurrence-events
8	40-49	premeno	10-14	0-2	no	2	left	left_up	no	no-recurrence-events
9	40-49	premeno	0-4	0-2	no	2	right	right_low	no	no-recurrence-events
10	40-49	ge40	40-44	15-17	yes	2	right	left_up	yes	no-recurrence-events
11	50-59	premeno	25-29	0-2	no	2	left	left_low	no	no-recurrence-events
12	60-69	ge40	15-19	0-2	no	2	right	left_up	no	no-recurrence-events
13	50-59	ge40	30-34	0-2	no	1	right	central	no	no-recurrence-events
14	50-59	ge40	25-29	0-2	no	2	right	left_up	no	no-recurrence-events
15	40-49	premeno	25-29	0-2	no	2	left	left_low	yes	recurrence-events
16	30-39	premeno	20-24	0-2	no	3	left	central	no	no-recurrence-events
17	50-59	premeno	10-14	3-5	no	1	right	left_up	no	no-recurrence-events
18	60-69	ge40	15-19	0-2	no	2	right	left_up	no	no-recurrence-events
19	50-59	premeno	40-44	0-2	no	2	left	left_up	no	no-recurrence-events
20	50-59	ge40	20-24	0-2	no	3	left	left_up	no	no-recurrence-events
21	50-59	lt40	20-24	0-2		1	left	left_low	no	recurrence-events
22	60-69	ge40	40-44	3-5	no	2	right	left_up	yes	no-recurrence-events
23	50-59	ge40	15-19	0-2	no	2	right	left_low	no	no-recurrence-events
24	40-49	premeno	10-14	0-2	no	1	right	left_up	no	no-recurrence-events

รูป 3.3 ตัวอย่างชุดข้อมูลมะเร็งเต้านม

3.1.4 เบาหวาน (Diabetes.arff)

เป็นชุดข้อมูลทางการแพทย์เกี่ยวกับสุขภาพของคนไข้ เพื่อทำนายว่าจะเป็นโรคเบาหวานหรือไม่ โดยจะประกอบด้วยแอตทริบิวต์ทั้งหมด 9 แอตทริบิวต์ แต่ละแอตทริบิวต์จะประกอบด้วยค่าต่างๆ ดังนี้

- preg จำนวนครั้งของการตั้งครรภ์ ประกอบด้วยค่าที่เป็นตัวเลข มีค่าอยู่ในช่วง 0-17
- plas ความเข้มข้นของกลูโคสพลาสม่า ประกอบด้วยค่าที่เป็นตัวเลข มีค่าอยู่ในช่วง 0-199

- pres ความดันเลือด (mm Hg) ประกอบด้วยค่าที่เป็นตัวเลข มีค่าอยู่ในช่วง 0-122
- skin ความหนาของผิวหนัง (mm) ประกอบด้วยค่าที่เป็นตัวเลข มีค่าอยู่ในช่วง 0-99
- insu ระดับอินซูลิน (mu U/ml) ประกอบด้วยค่าที่เป็นตัวเลข มีค่าอยู่ในช่วง 0-846
- mass ดัชนีมวลกาย (kg/m^2) ประกอบด้วยค่าที่เป็นตัวเลข มีค่าอยู่ในช่วง 0-67.1
- pedi ค่าจากฟังก์ชั่น DPF หาค่าความน่าจะเป็นจากเครือญาติ ประกอบด้วยค่าที่เป็นตัวเลข มีค่าอยู่ในช่วง 0.078-2.42
- age อายุ (ปี) ประกอบด้วยค่าที่เป็นตัวเลข มีค่าอยู่ในช่วง 21-81
- class แบ่งเป็น 2 ประเภท คือ เป็นโรคเบาหวาน (tested_positive) และไม่เป็นโรคเบาหวาน (tested_negative)

ชุดข้อมูลเบาหวานจะประกอบด้วยข้อมูลทั้งหมด 768 Instances

Relation: pima_diabetes

No.	preg Numeric	plas Numeric	pres Numeric	skin Numeric	insu Numeric	mass Numeric	pedi Numeric	age Numeric	class Nominal
1	6.0	148.0	72.0	35.0	0.0	33.6	0.627	50.0	tested_positive
2	1.0	85.0	66.0	29.0	0.0	26.6	0.351	31.0	tested_negative
3	8.0	183.0	64.0	0.0	0.0	23.3	0.672	32.0	tested_positive
4	1.0	89.0	66.0	23.0	94.0	28.1	0.167	21.0	tested_negative
5	0.0	137.0	40.0	35.0	168.0	43.1	2.288	33.0	tested_positive
6	5.0	116.0	74.0	0.0	0.0	25.6	0.201	30.0	tested_negative
7	3.0	78.0	50.0	32.0	88.0	31.0	0.248	26.0	tested_positive
8	10.0	115.0	0.0	0.0	0.0	35.3	0.134	29.0	tested_negative
9	2.0	197.0	70.0	45.0	543.0	30.5	0.158	53.0	tested_positive
10	8.0	125.0	96.0	0.0	0.0	0.0	0.232	54.0	tested_positive
11	4.0	110.0	92.0	0.0	0.0	37.6	0.191	30.0	tested_negative
12	10.0	168.0	74.0	0.0	0.0	38.0	0.537	34.0	tested_positive
13	10.0	139.0	80.0	0.0	0.0	27.1	1.441	57.0	tested_negative
14	1.0	189.0	60.0	23.0	846.0	30.1	0.398	59.0	tested_positive
15	5.0	166.0	72.0	19.0	175.0	25.8	0.587	51.0	tested_positive
16	7.0	100.0	0.0	0.0	0.0	30.0	0.484	32.0	tested_positive
17	0.0	118.0	84.0	47.0	230.0	45.8	0.551	31.0	tested_positive
18	7.0	107.0	74.0	0.0	0.0	29.6	0.254	31.0	tested_positive
19	1.0	103.0	30.0	38.0	83.0	43.3	0.183	33.0	tested_negative
20	1.0	115.0	70.0	30.0	96.0	34.6	0.529	32.0	tested_positive
21	3.0	126.0	88.0	41.0	235.0	39.3	0.704	27.0	tested_negative
22	8.0	99.0	84.0	0.0	0.0	35.4	0.388	50.0	tested_negative
23	7.0	196.0	90.0	0.0	0.0	39.8	0.451	41.0	tested_positive
24	9.0	119.0	80.0	35.0	0.0	29.0	0.263	29.0	tested_positive

รูป 3.4 ตัวอย่างชุดข้อมูลเบาหวาน

3.1.5 ถั่วเหลือง (Soybean.arff)

เป็นชุดข้อมูลเกี่ยวกับลักษณะที่เกิดขึ้นกับต้นถั่วเหลือง เพื่อทำนายว่าต้นถั่วเหลืองนี้เป็นโรคอะไร โดยจะประกอบด้วยแอตทริบิวต์ทั้งหมด 36 แอตทริบิวต์ แต่ละแอตทริบิวต์จะประกอบด้วยค่าต่างๆ ดังนี้

- date เดือนที่เกิดลักษณะดังกล่าว ประกอบด้วยค่า april, may, june, july, august, september และ october
- plant-stand ประกอบด้วยค่า normal และ lt-normal
- precip ประกอบด้วยค่า lt-norm, norm และ gt-norm

- temp ประกอบด้วยค่า lt-norm, norm และ gt-norm
- hail ประกอบด้วยค่า yes และ no
- crop-hist ประกอบด้วยค่า diff-lst-year, same-lst-yr, same-lst-two-yrs และ same-lst-sev-yrs
- area-damaged ประกอบด้วยค่า scattered, low-areas, upper-areas และ whole-field
- severity ประกอบด้วยค่า minor, pot-severe และ severe
- seed-tmt ประกอบด้วยค่า none, fungicide และ other
- germination ประกอบด้วยค่าแบ่งเป็นช่วง 90-100, 80-89 และ lt-80
- plant-growth ประกอบด้วยค่า norm และ abnorm
- leaves ประกอบด้วยค่า norm และ abnorm
- leafspots-halo ประกอบด้วยค่า absent, yellow-halos และ no-yellow-halos
- leafspots-marg ประกอบด้วยค่า w-s-marg, no-w-s-marg และ dna
- leafspot-size ประกอบด้วยค่า lt-1/8, gt-1/8 และ dna
- leaf-shread ประกอบด้วยค่า absent และ present
- leaf-malf ประกอบด้วยค่า absent และ present
- leaf-mild ประกอบด้วยค่า absent, upper-surf และ lower-surf
- stem ประกอบด้วยค่า norm และ abnorm
- lodging ประกอบด้วยค่า yes และ no
- stem-cankers ประกอบด้วยค่า absent, below-soil, above-soil และ above-sec-nde
- canker-lesion ประกอบด้วยค่า dna, brown, dk-brown-blk และ tan
- fruiting-bodies ประกอบด้วยค่า absent และ present
- external-decay ประกอบด้วยค่า absent, firm-and-dry และ watery
- mycelium ประกอบด้วยค่า absent และ present
- int-discolor ประกอบด้วยค่า none, brown และ black
- sclerotia ประกอบด้วยค่า absent และ present
- fruit-pods ประกอบด้วยค่า norm, diseased, few-present และ dna
- fruit-spots ประกอบด้วยค่า absent, colored, brown-w/blk-specks, distort และdna
- seed ประกอบด้วยค่า norm และ abnorm
- mold-growth ประกอบด้วยค่า absent และ present
- seed-discolor ประกอบด้วยค่า absent และ present
- seed-size ประกอบด้วยค่า norm และ lt-norm
- shriveling ประกอบด้วยค่า absent และ present

- roots ประกอบด้วยค่า norm, rotted และ galls-cysts
- class แบ่งประเภทของโรคในต้นถั่วเหลืองออกเป็น 19 โรค คือ diaporthe-stem-canker, charcoal-rot, rhizoctonia-root-rot, phytophthora-rot, brown-stem-rot, powdery-mildew, downy-mildew, brown-spot, bacterial-blight, bacterial-pustule, purple-seed-stain, anthracnose, phyllosticta-leaf-spot, alternarialeaf-spot, frog-eye-leaf-spot, diaporthe-pod-&-stem-blight, cyst-nematode, 2-4-d-injur และ herbicide-injury

ชุดข้อมูลถั่วเหลืองจะประกอบด้วยข้อมูลทั้งหมด 683 Instances

Relation: soybean

#	date	plant-s	precip	temp	hail	crop-hist	area-da	severity	seed-tr	germin	plant-g	leaves	leafspo	leaf	leaf-m	leaf-shr	leaf-ml	leaf-ml	stem	lodging	stem-can	canker	fruiting-	extern	mycelia	int-dis	sclerotia	fruit-p	fruit	seed	mold-g	seed-di	seed-s	shrivel	roots	class
	Nominal	Nom	Nominal	Nomin	Nom	Nominal	Nom	Nominal	Nominal	Nom	Nom	Nomin	Nom		ti	Nomin	Nomin	Nomin	Nomin	Nomin	Nom	Nomi	Nom	Nom	Nom	Nom	Nominal	Nom	Nom	Nom	Nom	Nom	Nomi	Nomi	Nominal
1	octo...	nor...	gt-n...	norm	yes	same-l...	low-a...	pot-se...	none	90-100	abn...	abn...	absent	dna	dna	absent	absent	absent	abn...	no	above...	brown	present	firm-...	absent	none	absent	norm	dna	norm	absent	absent	norm	absent	norm	diaporth...
2	august	nor...	gt-n...	norm	yes	same-l...	scatt...	severe	fungi...	80-89	abn...	abn...	absent	dna	dna	absent	absent	absent	abn...	yes	above...	brown	present	firm-...	absent	none	absent	norm	dna	norm	absent	absent	norm	absent	norm	diaporth...
3	july	nor...	gt-n...	norm	yes	same-l...	scatt...	severe	fungi...	lt-80	abn...	abn...	absent	dna	dna	absent	absent	absent	abn...	yes	above...	dna	present	firm-...	absent	none	absent	norm	dna	norm	absent	absent	norm	absent	norm	diaporth...
4	july	nor...	gt-n...	norm	yes	same-l...	scatt...	severe	none	80-89	abn...	abn...	absent	dna	dna	absent	absent	absent	abn...	yes	above...	dna	present	firm-...	absent	none	absent	norm	dna	norm	absent	absent	norm	absent	norm	diaporth...
5	octo...	nor...	gt-n...	norm	yes	same-l...	scatt...	pot-se...	none	lt-80	abn...	abn...	absent	dna	dna	absent	absent	absent	abn...	yes	above...	brown	present	firm-...	absent	none	absent	norm	dna	norm	absent	absent	norm	absent	norm	diaporth...
6	sept...	nor...	gt-n...	norm	yes	same-l...	scatt...	pot-se...	none	80-89	abn...	abn...	absent	dna	dna	absent	absent	absent	abn...	yes	above...	dna	present	firm-...	absent	none	absent	norm	dna	norm	absent	absent	norm	absent	norm	diaporth...
7	sept...	nor...	gt-n...	norm	yes	same-l...	scatt...	pot-se...	fungi...	90-100	abn...	abn...	absent	dna	dna	absent	absent	absent	abn...	no	above...	brown	present	firm-...	absent	none	absent	norm	dna	norm	absent	absent	norm	absent	norm	diaporth...
8	august	nor...	gt-n...	norm	no	same-l...	scatt...	pot-se...	none	lt-80	abn...	abn...	absent	dna	dna	absent	absent	absent	abn...	yes	above...	brown	present	firm-...	absent	none	absent	norm	dna	norm	absent	absent	norm	absent	norm	diaporth...
9	octo...	nor...	gt-n...	norm	yes	same-l...	scatt...	pot-se...	fungi...	80-89	abn...	abn...	absent	dna	dna	absent	absent	absent	abn...	yes	above...	brown	present	firm-...	absent	none	absent	norm	dna	norm	absent	absent	norm	absent	norm	diaporth...
10	august	nor...	gt-n...	norm	yes	same-l...	scatt...	severe	none	lt-80	abn...	abn...	absent	dna	dna	absent	absent	absent	abn...	yes	above...	brown	present	firm-...	absent	none	absent	norm	dna	norm	absent	absent	norm	absent	norm	diaporth...
11	octo...	nor...	lt-norm	gt-...	yes	same-l...	whole...	pot-se...	fungi...	90-100	abn...	abn...	absent	dna	dna	absent	absent	absent	abn...	yes	absent	tan	absent	absent	absent	black	present	norm	dna	norm	absent	absent	norm	absent	norm	charcoal...
12	august	nor...	lt-norm	norm	no	same-l...	whole...	pot-se...	fungi...	80-89	abn...	abn...	absent	dna	dna	absent	absent	absent	abn...	no	absent	tan	absent	absent	absent	black	present	norm	dna	norm	absent	absent	norm	absent	norm	charcoal...
13	july	nor...	lt-norm	norm	yes	same-l...	upper...	pot-se...	none	90-100	abn...	abn...	absent	dna	dna	absent	absent	absent	abn...	yes	absent	tan	absent	absent	absent	black	present	norm	dna	norm	absent	absent	norm	absent	norm	charcoal...
14	octo...	nor...	lt-norm	norm	no	same-l...	whole...	pot-se...	fungi...	90-100	abn...	abn...	absent	dna	dna	absent	absent	absent	abn...	yes	absent	tan	absent	absent	absent	black	present	norm	dna	norm	absent	absent	norm	absent	norm	charcoal...
15	octo...	nor...	lt-norm	gt-...	yes	same-l...	whole...	pot-se...	fungi...	80-89	abn...	abn...	absent	dna	dna	absent	absent	absent	abn...	yes	absent	tan	absent	absent	absent	black	present	norm	dna	norm	absent	absent	norm	absent	norm	charcoal...
16	sept...	nor...	lt-norm	gt-...	no	same-l...	whole...	pot-se...	fungi...	lt-80	abn...	abn...	absent	dna	dna	absent	absent	absent	abn...	yes	absent	tan	absent	absent	absent	black	present	norm	dna	norm	absent	absent	norm	absent	norm	charcoal...
17	octo...	nor...	lt-norm	gt-...	no	diff-lst...	upper...	pot-se...	none	90-100	abn...	abn...	absent	dna	dna	absent	absent	absent	abn...	no	absent	tan	absent	absent	absent	black	present	norm	dna	norm	absent	absent	norm	absent	norm	charcoal...
18	august	nor...	lt-norm	norm	yes	same-l...	upper...	pot-se...	none	80-89	abn...	abn...	absent	dna	dna	absent	absent	absent	abn...	yes	absent	tan	absent	absent	absent	black	present	norm	dna	norm	absent	absent	norm	absent	norm	charcoal...
19	july	nor...	lt-norm	gt-...	yes	same-l...	upper...	pot-se...	none	lt-80	abn...	abn...	absent	dna	dna	absent	absent	absent	abn...	yes	absent	tan	absent	absent	absent	black	present	norm	dna	norm	absent	absent	norm	absent	norm	charcoal...
20	sept...	nor...	lt-norm	gt-...	no	same-l...	upper...	pot-se...	none	lt-80	abn...	abn...	absent	dna	dna	absent	absent	absent	abn...	yes	absent	tan	absent	absent	absent	black	present	norm	dna	norm	absent	absent	norm	absent	norm	charcoal...
21	may	lt-r-...	gt-n...	lt-n...	yes	same-l...	low-a...	severe	none	lt-80	abn...	norm	absent	dna	dna	absent	absent	absent	abn...	yes	below...	brown	absent	firm-...	pres...	none	absent	dna	dna	norm	absent	absent	norm	absent	norm	rhizocto...
22	may	lt-r-...	gt-n...	lt-n...	yes	same-l...	low-a...	severe	none	80-89	abn...	norm	absent	dna	dna	absent	absent	absent	abn...	yes	below...	brown	absent	firm-...	absent	none	absent	dna	dna	norm	absent	absent	norm	absent	norm	rhizocto...
23	july	nor...	gt-n...	lt-n...	no	same-l...	low-a...	severe	none	80-89	abn...	norm	absent	dna	dna	absent	absent	absent	abn...	no	below...	brown	absent	firm-...	pres...	none	absent	dna	dna	norm	absent	absent	norm	absent	norm	rhizocto...
24	april	lt-r-...	gt-n...	lt-n...	yes	diff-lst...	low-a...	cot-se...	fungi...	lt-80	abn...	norm	absent	dna	dna	absent	absent	absent	abn...	yes	below...	brown	absent	firm-...	absent	none	absent	dna	dna	norm	absent	absent	norm	absent	norm	rhizocto...

รูป 3.5 ตัวอย่างชุดข้อมูลถั่วเหลือง

3.1.6 การลงคะแนนเสียงเลือกตั้ง (Vote.arff)

เป็นชุดข้อมูลเกี่ยวกับการเลือกตั้งของประเทศอเมริกาสองพรรคที่แข่งขันกัน โดยจะมีการเก็บข้อมูลจากประชาชนว่าชอบนโยบายใด ไม่ชอบนโยบายใดบ้าง และสุดท้ายประชาชนเลือกพรรคใด โดยจะประกอบด้วยแอตทริบิวต์ทั้งหมด 17 แอตทริบิวต์ แต่ละแอตทริบิวต์จะประกอบด้วยค่าต่างๆ ดังนี้

- handicapped-infants ประกอบด้วยค่า n และ y
- water-project-cost-sharing ประกอบด้วยค่า n และ y
- adoption-of-the-budget-resolution ประกอบด้วยค่า n และ y
- physician-free-freeze ประกอบด้วยค่า n และ y
- el-salvador-aid ประกอบด้วยค่า n และ y
- religious-groups-in-schools ประกอบด้วยค่า n และ y
- anti-satellite-test-ban ประกอบด้วยค่า n และ y
- aid-to-nicaraguan-contras ประกอบด้วยค่า n และ y

- mx-missile ประกอบด้วยค่า n และ y
- immigration ประกอบด้วยค่า n และ y
- synfuels-corporation-cutback ประกอบด้วยค่า n และ y
- education-spending ประกอบด้วยค่า n และ y
- superfund-right-to-use ประกอบด้วยค่า n และ y
- crime ประกอบด้วย n และ y
- duty-free-exports ประกอบด้วยค่า n และ y
- export-administration-act-south-africa ประกอบด้วยค่า n และ y
- Class แบ่งออกเป็น 2 พรรค คือ democrat และ republican

ชุดข้อมูลการลงคะแนนเสียงเลือกตั้งจะประกอบด้วยข้อมูลทั้งหมด 435 Instances ซึ่งในบาง Instance จะมีค่าในบางแอตทริบิวต์ที่ไม่มีข้อมูล

Relation: vote

No.	han	wate	adc	phy	el-s	reli	anti	aid	mx-	imm	syn	edu	supe	crir	dut	exp	Class
									Nc	Nc				Non			Nominal
1	n	y	n	y	y	y	n	n	n	y		y	y	y	n	y	republican
2	n	y	n	y	y	y	n	n	n	n	n	y	y	y	n		republican
3		y	y		y	y	n	n	n	n	y	n	y	y	n	n	democrat
4	n	y	y	n		y	n	n	n	n	y	n	y	n	n	y	democrat
5	y	y	y	n	y	y	n	n	n	n	y		y	y	y	y	democrat
6	n	y	y	n	y	y	n	n	n	n	n	n	y	y	y	y	democrat
7	n	y	n	y	y	y	n	n	n	n	n	n		y	y	y	democrat
8	n	y	n	y	y	y	n	n	n	n	n	n	y	y		y	republican
9	n	y	n	y	y	y	n	n	n	n	n	y	y	y	n	y	republican
10	y	y	y	n	n	n	y	y	y	n	n	n	n	n			democrat
11	n	y	n	y	y	n	n	n	n	n			y	y	n	n	republican
12	n	y	n	y	y	y	n	n	n	n	y		y	y			republican
13	n	y	y	n	n	n	y	y	y	n	n	n	y	n			democrat
14	y	y	y	n	n	y	y	y		y	y		n	n	y		democrat
15	n	y	n	y	y	y	n	n	n	n	n	y			n		republican
16	n	y	n	y	y	y	n	n	n	y	n	y	y		n		republican
17	y	n	y	n	n	y	n	y		y	y	y		n	n	y	democrat
18	y		y	n	n	n	y	y	y	n	n	n	y	n	y	y	democrat
19	n	y	n	y	y	y	n	n	n	n	n		y	y	n	n	republican
20	y	y	y	n	n	n	y	y	y	n	y	n	n	n	y	y	democrat
21	y	y	y	n	n		y	y	n	n	y	n	n	n	y	y	democrat
22	y	y	y	n	n	n	y	y	y	n	n	n			y	y	democrat
23	y		y	n	n	n	y	y	y	n	n		n	n	y	y	democrat
24	y	y	y	n	n	n	y	y	y	n	n	n	n	n	y	y	democrat

รูป 3.6 ตัวอย่างชุดข้อมูลการลงคะแนนเสียงเลือกตั้ง

3.1.7 สินเชื่อ (Credit-g.arff)

เป็นชุดข้อมูลเกี่ยวกับข้อมูลทั่วๆไปต่างๆ ของลูกค้า และข้อมูลการทำธุรกรรมในอดีตของลูกค้าของธนาคารแห่งหนึ่งในเยอรมัน เพื่อจำแนกว่าเป็นลูกค้าที่ดีขององค์กรหรือไม่ โดยจะประกอบด้วยแอตทริบิวต์ทั้งหมด 21 แอตทริบิวต์ แต่ละแอตทริบิวต์จะประกอบด้วยค่าต่างๆ ดังนี้

- checking_status ประกอบด้วยค่า <0, 0<=x<200, >=200 และ no checking
- duration ประกอบด้วยค่าที่เป็นตัวเลข มีค่าอยู่ในช่วง 4.0-72.0

- credit_history ประกอบด้วยค่า no credits/all paid, all paid, existing paid, delayed previously และ critical/other existing credit
- purpose ประกอบด้วยค่า new car, used car, furniture/equipment, radio/tv, domestic appliance, repairs, education, vacation, retraining, business และ other
- credit_amount ประกอบด้วยค่าที่เป็นตัวเลข มีค่าอยู่ในช่วง 250-18424
- savings_status ประกอบด้วยค่า <100, 100<=x<500, 500<=x<1000, >=1000 และ no known saving
- employment ประกอบด้วยค่า unemployed, <1, 1<=x<4, 4<=x<7 และ >=7
- installment_commitment ประกอบด้วยค่าที่เป็นตัวเลข มีค่าอยู่ในช่วง 1.0-4.0
- personal_status ประกอบด้วยค่า male div/sep, female div/dep/mar, male single, male mar/wid และ female single
- other_parties ประกอบด้วยค่า none, co applicant และ guarantor
- residence_since ประกอบด้วยค่าที่เป็นตัวเลข มีค่าอยู่ในช่วง 1.0-4.0
- property_magnitude ประกอบด้วยค่า real estate, life insurance, car, และ no known property
- age ประกอบด้วยค่าที่เป็นตัวเลข มีค่าอยู่ในช่วง 19.0-75.0
- other_payment_plants ประกอบด้วยค่า back, stores และ none
- housing ประกอบด้วยค่า rent, own และ for free
- existing_credits ประกอบด้วยค่าที่เป็นตัวเลข มีค่าอยู่ในช่วง 1.0-4.0
- job ประกอบด้วยค่า unemp/unskilled non res, unskilled resident, skilled และ high qualif/self emp/mgmt.
- num_dependents ประกอบด้วยค่าที่เป็นตัวเลข มีค่าอยู่ในช่วง 1.0-2.0
- own_telephone ประกอบด้วยค่า none และ yes
- foreign_worker ประกอบด้วยค่า yes และ no
- class แบ่งลูกค้าเป็น 2 ประเภท คือ good และ bad

ชุดข้อมูลสินเชื่อจะประกอบด้วยข้อมูลทั้งหมด 1000 Instances

Relation: german_credit

No	checking_sta Nominal	dura Num	credit_history Nominal	purpose Nominal	credit_ Nur	savings_status Nominal	employment Nominal	insta	personal_status Nominal	othe Nc	resi	property_magnitude Nominal	age Num	othe	housing Nominal	exis	job Nominal	num	own_ N	fore f	class Nominal
1	('(0	6.0	critical/other existing credit	radio/tv	1169.0	no known savings	)=7		4.0 male single	none	4.0	real estate	67.0	none	own	2.0	skilled	1.0	yes	yes	good
2	0(=X(200	48.0	existing paid	radio/tv	5951.0	('(100	1(=X(4		2.0 female div/dep/mar	none	2.0	real estate	22.0	none	own	1.0	skilled	1.0	none	yes	bad
3	no checking	12.0	critical/other existing credit	education	2096.0	('(100	4(=X(7		2.0 male single	none	3.0	real estate	49.0	none	own	1.0	unskilled resident	2.0	none	yes	good
4	('(0	42.0	existing paid	furnitu...	7882.0	('(100	4(=X(7		2.0 male single	gu...	4.0	life insurance	45.0	none	for free	1.0	skilled	2.0	none	yes	good
5	('(0	24.0	delayed previously	new car	4870.0	('(100	1(=X(4		3.0 male single	none	4.0	no known property	53.0	none	for free	2.0	skilled	2.0	none	yes	bad
6	no checking	36.0	existing paid	education	9055.0	no known savings	1(=X(4		2.0 male single	none	4.0	no known property	35.0	none	for free	1.0	unskilled resident	2.0	yes	yes	good
7	no checking	24.0	existing paid	furnitu...	2835.0	500(=X(1000	)=7		3.0 male single	none	4.0	life insurance	53.0	none	own	1.0	skilled	1.0	none	yes	good
8	0(=X(200	36.0	existing paid	used car	6948.0	('(100	1(=X(4		2.0 male single	none	2.0	car	35.0	none	rent	1.0	high qualif/self emp/mgmt	1.0	yes	yes	good
9	no checking	12.0	existing paid	radio/tv	3059.0	)=1000	4(=X(7		2.0 male div/sep	none	4.0	real estate	61.0	none	own	1.0	unskilled resident	1.0	none	yes	good
10	0(=X(200	30.0	critical/other existing credit	new car	5234.0	('(100	unemployed		4.0 male mar/wid	none	2.0	car	28.0	none	own	2.0	high qualif/self emp/mgmt	1.0	none	yes	bad
11	0(=X(200	12.0	existing paid	new car	1295.0	('(100	('(1		3.0 female div/dep/mar	none	1.0	car	25.0	none	rent	1.0	skilled	1.0	none	yes	bad
12	('(0	48.0	existing paid	business	4308.0	('(100	('(1		3.0 female div/dep/mar	none	4.0	life insurance	24.0	none	rent	1.0	skilled	1.0	none	yes	bad
13	0(=X(200	12.0	existing paid	radio/tv	1567.0	('(100	1(=X(4		1.0 female div/dep/mar	none	1.0	car	22.0	none	own	1.0	skilled	1.0	yes	yes	good
14	('(0	24.0	critical/other existing credit	new car	1199.0	('(100	)=7		4.0 male single	none	4.0	car	60.0	none	own	2.0	unskilled resident	1.0	none	yes	bad
15	('(0	15.0	existing paid	new car	1403.0	('(100	1(=X(4		2.0 female div/dep/mar	none	4.0	car	28.0	none	rent	1.0	skilled	1.0	none	yes	good
16	('(0	24.0	existing paid	radio/tv	1282.0	100(=X(500	1(=X(4		4.0 female div/dep/mar	none	2.0	car	32.0	none	own	1.0	unskilled resident	1.0	none	yes	bad
17	no checking	24.0	critical/other existing credit	radio/tv	2424.0	no known savings	)=7		4.0 male single	none	4.0	life insurance	53.0	none	own	2.0	skilled	1.0	none	yes	good
18	('(0	30.0	no credits/all paid	business	8072.0	no known savings	('(1		2.0 male single	none	3.0	car	25.0	bank	own	3.0	skilled	1.0	none	yes	good
19	0(=X(200	24.0	existing paid	used car	125...	('(100	)=7		4.0 female div/dep/mar	none	2.0	no known property	44.0	none	for free	1.0	high qualif/self emp/mgmt	1.0	yes	yes	bad
20	no checking	24.0	existing paid	radio/tv	3430.0	500(=X(1000	)=7		3.0 male single	none	2.0	car	31.0	none	own	1.0	skilled	2.0	yes	yes	good
21	no checking	9.0	critical/other existing credit	new car	2134.0	('(100	1(=X(4		4.0 male single	none	4.0	car	48.0	none	own	3.0	skilled	1.0	yes	yes	good
22	('(0	6.0	existing paid	radio/tv	2647.0	500(=X(1000	1(=X(4		2.0 male single	none	3.0	real estate	44.0	none	rent	1.0	skilled	2.0	none	yes	good
23	('(0	10.0	critical/other existing credit	new car	2241.0	('(100	('(1		1.0 male single	none	3.0	real estate	48.0	none	rent	2.0	unskilled resident	2.0	none	no	good
24	0(=X(200	12.0	critical/other existing credit	used car	1804.0	100(=X(500	('(1		3.0 male single	none	4.0	life insurance	44.0	none	own	1.0	skilled	1.0	none	yes	good

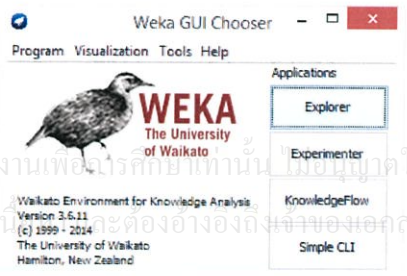
รูป 3.7 ตัวอย่างชุดข้อมูลสินเชื่อ

3.2 เครื่องมือที่ใช้ในการทำเหมืองข้อมูล

เนื่องจากการทำเหมืองข้อมูลกำลังได้รับความเป็นที่นิยมมากขึ้นเรื่อยๆ ทั้งในต่างประเทศและในประเทศไทย ทำให้มีการพัฒนาเครื่องมือที่ใช้ในการทำเหมืองข้อมูลออกมาเป็นจำนวนมาก มีทั้งแบบที่ต้องซื้อหามาใช้หรือสามารถนำมาใช้ได้ฟรี ทั้งแบบเต็มรูปแบบหรือแบบทดลองใช้งาน แต่ซอฟต์แวร์ที่นิยมใช้กันมากที่สุดติดอันดับและเป็นแบบที่สามารถนำมาใช้ได้ฟรี เป็นซอฟต์แวร์ที่เราจะเลือกนำมาใช้ในการศึกษาทดลองประกอบด้วยซอฟต์แวร์ 3 ซอฟต์แวร์คือ เวก้า (Weka) ราปิดมายเนอร์ (RapidMiner) และอาร์ ดาต้ามายนิ่ง (R Data Mining) ซึ่งการใช้งาน และวิธีใช้งานของแต่ละซอฟต์แวร์มีความแตกต่างกันทั้งทางด้านหน้าต่างการใช้งานสำหรับผู้ใช้และการออกแบบการใช้งาน

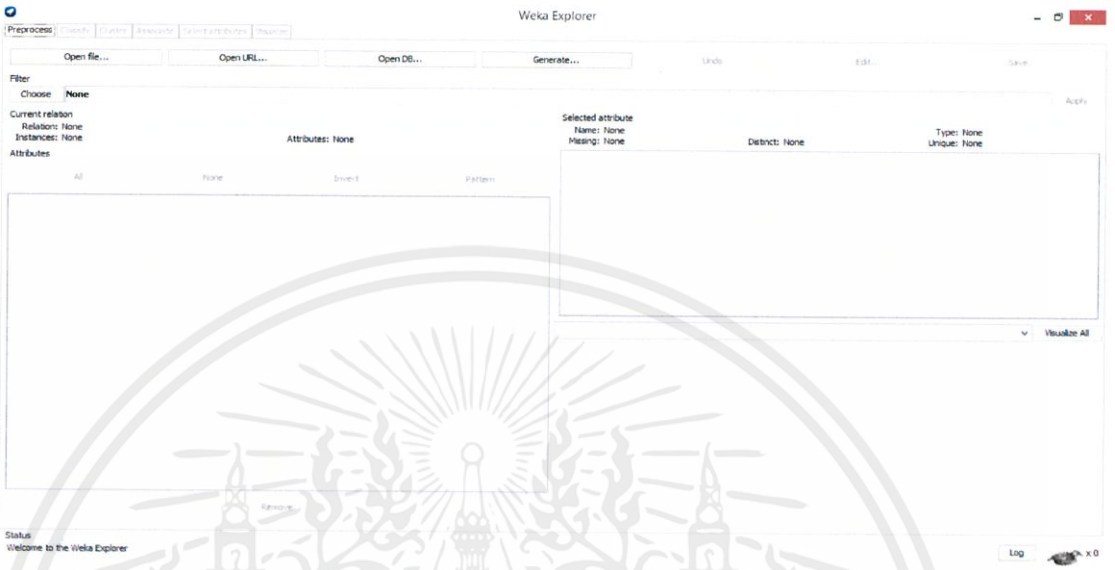
3.2.1 การสร้างโมเดลทำนายค่าในซอฟต์แวร์เวก้า

สำหรับหน้าต่างการใช้งานของซอฟต์แวร์เวก้าเมื่อดับเบิ้ลคลิกตัวไอคอนการใช้งานขึ้นมาจะปรากฏหน้าต่างดังรูป



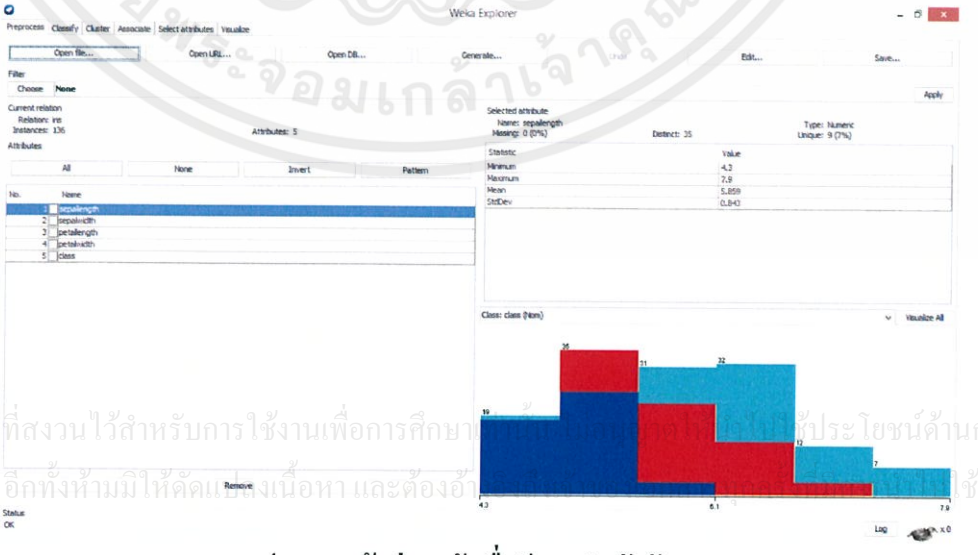
รูป 3.8 หน้าต่างใช้งานของเวก้า

โดยส่วนที่เราจะใช้งานจะเป็นส่วนของหน้าต่าง Explorer ซึ่งเป็นส่วนที่ใช้งานง่ายที่สุดสำหรับผู้เริ่มใช้งาน เมื่อทำการเลือกหน้าต่าง Explorer จะปรากฏหน้าต่างดังรูป



รูป 3.9 หน้าต่าง Explorer ของโปรแกรม Weka

ในขั้นตอนเริ่มแรกเราจะต้องทำการเลือกชุดข้อมูลที่จะนำมาทำการเรียนรู้เข้าสู่ซอฟต์แวร์ก่อน โดยสามารถนำข้อมูลเข้าได้โดยคลิก Open file… ในหน้า Preprocess ซึ่งในหน้าของ Preprocess เป็นส่วนสำหรับให้ผู้ใช้ทำการนำเข้าข้อมูล แก้ไขข้อมูล และบันทึกข้อมูล รวมไปถึงการคัดกรองข้อมูล ซึ่งรวมๆแล้วหมายถึงเป็นการเตรียมข้อมูลให้พร้อมก่อนการนำไปทำเหมืองข้อมูลนั่นเอง เมื่อเลือกข้อมูลเข้ามาได้แล้วในหน้าต่างการใช้งานจะปรากฏข้อมูลดังรูป



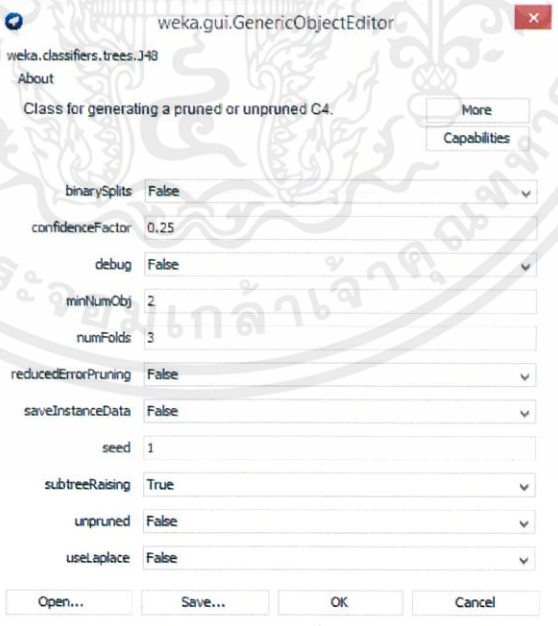
รูป 3.10 หน้าต่างเวก้าเมื่อมีการนำเข้าข้อมูล

หลังจากนำเข้าข้อมูลเรียบร้อยแล้วจะสังเกตุเห็นว่าในหน้าต่างส่วนอื่นๆ เราสามารถใช้งานได้แล้ว หลังจากนั้นเราจะทำการใช้งานในส่วนของหน้าต่าง Classify จะสังเกตุเห็นว่าในหน้าต่างนี้จะมีส่วนสำหรับให้ผู้ใช้ทำการเลือกอัลกอริทึมที่จะใช้ในการสร้างโมเดลทำนายค่า โดยคลิกเลือกที่ปุ่ม Choose จะพบว่ามีอัลกอริทึมในการทำ Classifier หลากหลายกลุ่มในแต่ละกลุ่มก็จะมีอัลกอริทึมแตกต่างกันไป โดยในการทดลองเราจะเลือกใช้อัลกอริทึมดังต่อไปนี้

- อัลกอริทึมในกลุ่ม bayes เลือกใช้ Naïve Bayes
- อัลกอริทึมในกลุ่ม lazy เลือกใช้ IBk
- อัลกอริทึมในกลุ่ม meta เลือกใช้ AdaBoostM1
- อัลกอริทึมในกลุ่ม rules เลือกใช้ ZeroR และ OneR
- อัลกอริทึมในกลุ่ม trees เลือกใช้ J48

เมื่อทำการเลือกอัลกอริทึมที่จะใช้งานแล้วจะปรากฏชื่ออัลกอริทึมที่เราเลือกในช่องหลังปุ่ม Choose ผู้ใช้สามารถทำการปรับหรือตั้งค่าพารามิเตอร์ต่างๆ ของอัลกอริทึมนั้นได้โดยคลิกที่ชื่ออัลกอริทึม

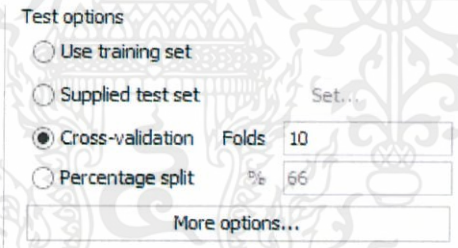
รูป 3.11 อัลกอริทึมที่เลือกใช้งาน



รูป 3.12 การปรับตั้งค่าพารามิเตอร์

โดยในการทดลองในด้านต่างๆ อาจมีการปรับตั้งค่าพารามิเตอร์แตกต่างกันไปตามความเหมาะสม หลังจากทำการปรับตั้งค่าพารามิเตอร์ต่างๆ เรียบร้อยแล้ว ในหน้าต่างนี้ยังมีส่วนสำหรับให้ผู้ใช้ทำการปรับแต่งตั้งค่าการทดสอบได้ว่าต้องการทดสอบแบบใด แบ่งเป็น

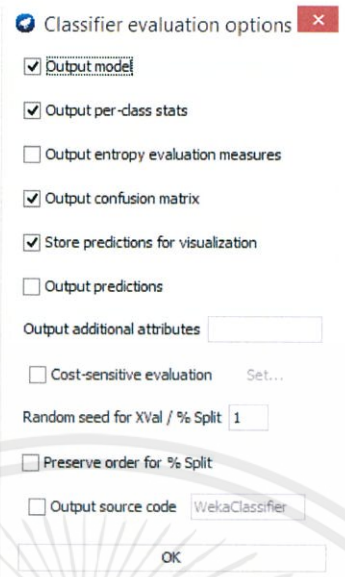
- Use training set ใช้ชุดข้อมูลทดสอบเดียวกันกับชุดข้อมูลเรียนรู้ ซึ่งผลลัพธ์ที่ได้จากการทดสอบมันจะมีเปอร์เซ็นต์ความถูกต้องสูงที่สุด
- Supplied test set ผู้ใช้สามารถทำการกำหนดชุดข้อมูลทดสอบได้เอง โดยสามารถนำเข้าชุดข้อมูลทดสอบเข้าได้ในส่วนนี้ ผลลัพธ์ที่ได้ขึ้นอยู่กับชุดข้อมูลที่นำมาทดสอบ
- Cross-validation เป็นการแบ่งชุดข้อมูลจากชุดข้อมูลเรียนรู้ออกเป็นส่วนๆตามจำนวนที่ผู้ใช้กำหนด โดยมี 1 ส่วนเป็นชุดข้อมูลทดสอบ และส่วนที่เหลือเป็นชุดข้อมูลเรียนรู้ แล้วทำการเรียนรู้ข้อมูล ทำซ้ำเช่นนี้ไปเรื่อยๆ โดยการเปลี่ยนชุดข้อมูลทดสอบจนข้อมูลทุกส่วนเคยเป็นชุดข้อมูลทดสอบหมด เลือกโมเดลที่มีประสิทธิภาพมากที่สุด และนำไปทดสอบอีกครั้ง
- Percentage split เป็นการกำหนดว่าจากชุดข้อมูลเรียนรู้ที่ผู้ใช้นำเข้ามานั้น จะแบ่งเป็นชุดข้อมูลเรียนรู้กี่ส่วน และที่เหลือแบ่งเป็นชุดข้อมูลทดสอบ



รูป 3.13 หน้าต่างในส่วน Test options

ในการตั้งค่าส่วน Test option นี้ จะมีการกำหนดค่าแตกต่างกันไปตามการใช้งานในแต่ละด้านตามความเหมาะสม

ในส่วนของการแสดงผลผู้ใช้สามารถเลือกข้อมูลที่จะแสดงผลได้โดยคลิก More option… และทำการตั้งค่าคลิกเครื่องหมายถูกในส่วนที่ต้องการให้แสดง



รูป 3.14 การตั้งค่าการแสดงผล

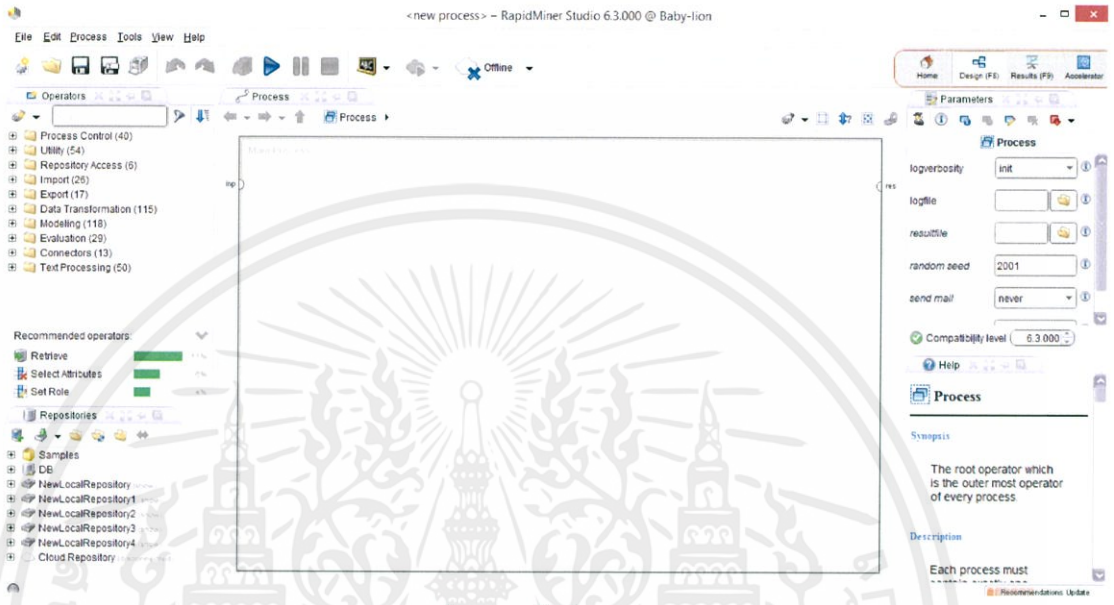
3.2.2 การสร้างโมเดลทำนายค่าในซอฟต์แวร์ราปิดมายเนอร์

การใช้งานซอฟต์แวร์ราปิดมายเนอร์ จะมีหน้าต่างการใช้งานเริ่มต้นเมื่อดับเบิ้ลคลิกตัวไอคอนใช้งาน ดังรูป

รูป 3.15 หน้าต่างเริ่มต้นใช้งานซอฟต์แวร์ราปิดมายเนอร์

หากเป็นผู้ใช้งานเริ่มต้นสามารถเลือกใช้งานแบบ Tutorials ได้ซึ่งจะมีการสอนการใช้งานตัวซอฟต์แวร์แบบทีละขั้นตอน หากต้องการทำเหมืองข้อมูลด้วยวิธีที่ง่ายและรวดเร็วที่สุดผู้ใช้สามารถ

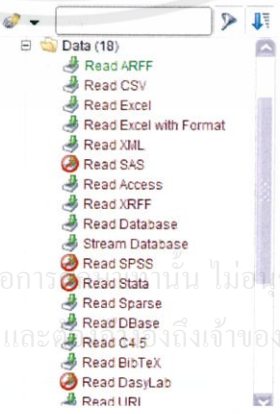
เลือก Accelerator ได้ หรือสามารถเลือกเริ่มการทำเหมืองข้อมูลใหม่ได้โดยคลิก New Process หรือหากผู้ใช้งานต้องการเรียกใช้โมเดลที่เคยทำสามารถเรียกใช้ได้จาก Open โดยในส่วนของการทดลองนี้จะทำการคลิกเลือก New Process ซึ่งจะแสดงผลหน้าต่างดังรูป



รูป 3.16 หน้าต่าง Design

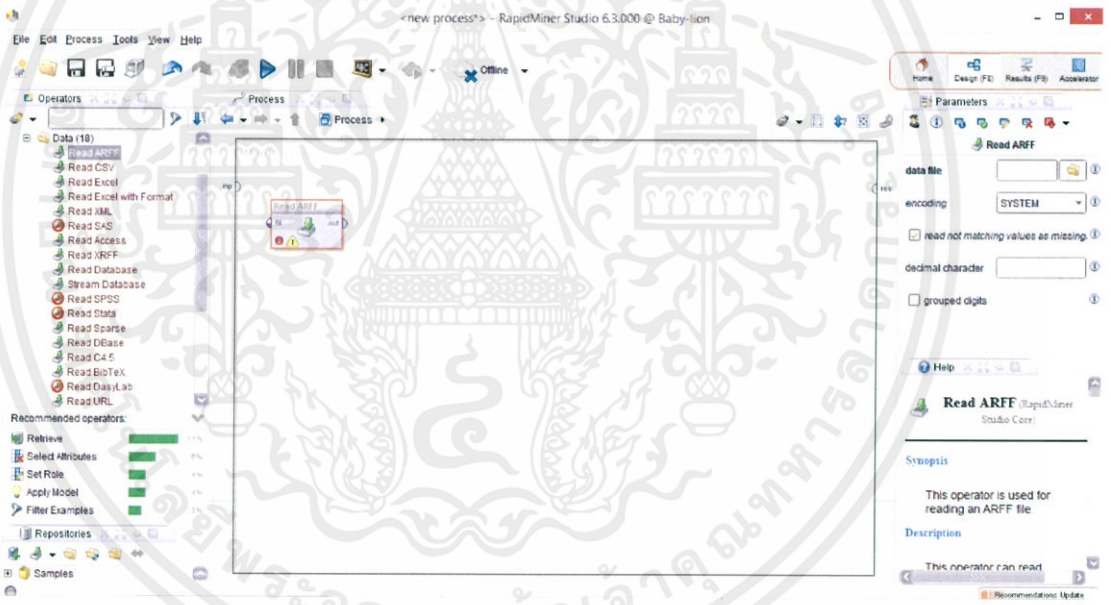
ซึ่งการใช้งานซอฟต์แวร์ราปิดมายเนอร์จะคล้ายกับการต่อวงจร ซึ่งตัวดำเนินการต่างๆ จะอยู่ในรูปของโมดูล ซึ่งผู้ใช้สามารถเลือกโมดูลมาต่อกันได้จากทางฝั่งซ้ายมือ จะเห็นว่ามีโมดูลสำหรับการดำเนินการต่างๆ อย่างมากมาย โดยแบ่งไว้เป็นหมวดหมู่

ในขั้นตอนเริ่มต้นเราจะต้องทำการนำเข้าข้อมูลที่ใช้ในการเรียนรู้เข้าสู่กระบวนการก่อน โดยโมดูลสำหรับการนำเข้าข้อมูลจะอยู่ในหมวดหมู่ Import>Data จะปรากฏโมดูลสำหรับการนำเข้าข้อมูลหลากหลายแบบซึ่งรองรับหลายสกุลไฟล์ ดังรูป



รูป 3.17 โมดูลสำหรับการนำเข้าข้อมูล

จะสังเกตเห็นว่าในบางโมดูลมีวงกลมสีแดงวงไว้ หมายถึงว่าเราจะไม่สามารถใช้งานโมดูลนั้นๆ ได้ เนื่องจากซอฟต์แวร์ราปิดมายเนอร์จะมีการแบ่งระดับผู้ใช้งานออกเป็นระดับต่างๆ ซึ่งแต่ละระดับจะมีความสามารถเข้าถึงการใช้งานได้แตกต่างกัน ผู้ใช้ในระดับเริ่มต้นจะเป็นเพียงระดับ Starter Edition และระดับที่สองผู้ใช้จะต้องทำการลงทะเบียนอีเมล์เพื่อทำการขอ License เพื่อการเข้าใช้งานเพิ่มเติม ซึ่ง 1 อีเมล์จะทำการขอ License ได้เพียงหนึ่งครั้งเท่านั้น และจำกัดเวลาในการใช้งาน เรียกว่าเป็นระดับ Professional Trial Edition เมื่อ license หมดอายุ ผู้ใช้งานก็จะกลับมาสู่ระดับ Starter Edition ตามเดิม โดยในการทดลองผู้ทดลองจะใช้ข้อมูลสกุลไฟล์ ARFF ในการทดลอง ซึ่งในระดับ Starter Edition ไม่สามารถใช้งานโมดูลสำหรับนำเข้าข้อมูลสกุลไฟล์ ARFF ได้ ผู้ใช้จึงต้องทำการลงทะเบียนอีเมล์เพื่อขอ License เมื่อทำการกรอก License เรียบร้อยแล้ว จะสังเกตเห็นว่าในบางโมดูลที่เคยใช้งานไม่ได้ ผู้ใช้จะสามารถใช้งานได้แล้ว รวมถึงโมดูล Read ARFF ที่เราต้องการใช้งานด้วย เมื่อสามารถใช้งานโมดูลได้แล้วให้เราทำการคลิกที่ชื่อโมดูลนั้นแล้วลากมาว่างในพื้นที่สำหรับต่อ Process ดังรูป



รูป 3.18 การเลือกใช้โมดูล Read ARFF

หลังจากนั้นให้ผู้ใช้ทำการคลิกที่โมดูลในพื้นที่ต่อ Process ซึ่งทางด้านขวามือจะขึ้นข้อมูลสำหรับการตั้งค่าพารามิเตอร์และการเลือกไฟล์ ผู้ใช้สามารถทำการเลือกไฟล์ที่ต้องการได้จากไอคอนรูปไฟล์ ในช่อง data file จากนั้นโปรแกรมจะให้เราทำการเลือกไฟล์

หลังจากการนำเข้าข้อมูลได้แล้วในขั้นตอนต่อไป เนื่องจากเราจะทำการทำเหมืองข้อมูลด้านการทำนายข้อมูล ดังนั้นในซอฟต์แวร์ราปิดมายเนอร์เราจะต้องทำการตั้งค่าแอตทริบิวต์ที่เราต้องการจะทำนายค่า ซึ่งในส่วนของซอฟต์แวร์เวก้าจะถือว่าแอตทริบิวต์สุดท้ายเป็นแอตทริบิวต์ที่เราต้องการทำนาย แต่ในซอฟต์แวร์ราปิดมายเนอร์แอตทริบิวต์ที่เราต้องการทำนายจะสามารถอยู่ในตำ

แหน่งที่เท่าใดก็ได้ตามแต่ผู้ใช้กำหนด ซึ่งโมดูลสำหรับการตั้งค่านี้มีชื่อว่าโมดูล Set Role ซึ่งอยู่ในหมวดหมู่ Data Transformation>Name and Role Modification ทำการลากโมดูลวางลงบนพื้นที่สำหรับ Process ต่อจากโมดูล Read ARFF และทำการต่อ Output ของโมดูล Read ARFF กับ Input ของโมดูล Set Role และทำการตั้งค่าพารามิเตอร์ทางด้านขวาของหน้าต่างใช้งาน โดยในช่อง Attribute name ใส่ชื่อแอตทริบิวต์ที่เราต้องการทำการตั้งค่า และในช่อง Target role ทำการเลือกค่าเป็น Label

หลังจากนั้นเราจะทำการเลือกใช้โมดูลอัลกอริทึมสำหรับการทำนายค่าจากโมดูลในหมวดหมู่ Modeling>Classification and Regression ซึ่งจะมีการแบ่งอัลกอริทึมออกเป็นอีกหลายหมวดหมู่เช่นเดียวกับซอฟต์แวร์เว็ก้า ซึ่งในการทดลองเราจะทำการเลือกใช้อัลกอริทึมต่างๆ ดังต่อไปนี้

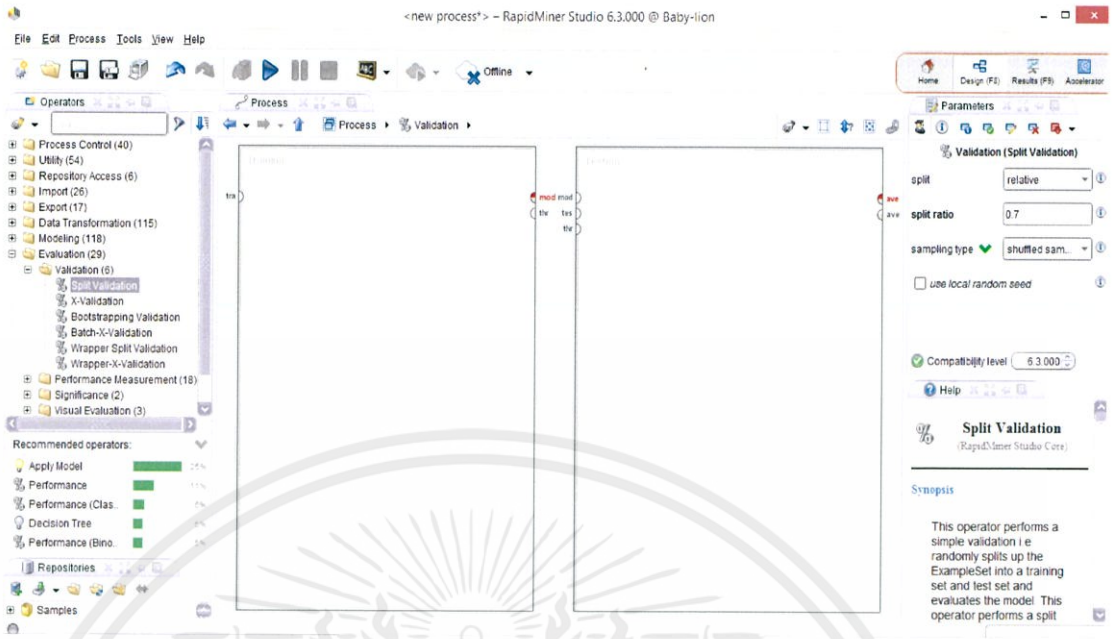
- อัลกอริทึมในกลุ่ม Bayesian Modeling เลือกใช้ Naïve Bayes
- อัลกอริทึมในกลุ่ม Tree Induction เลือกใช้ Decision Tree
- อัลกอริทึมในกลุ่ม Meta Modeling เลือกใช้ AdaBoost

ทำการเลือกโมดูลอัลกอริทึมที่ต้องการใช้งานลงในพื้นที่ Process แต่เนื่องจากในซอฟต์แวร์ราปิดมายเนอร์ ผู้ใช้จะต้องทำการเลือกรูปแบบในการทดสอบก่อน เนื่องจากในบางรูปแบบผู้ใช้จะต้องทำการนำโมดูลอัลกอริทึมที่ต้องใช้นั้นไว้เป็นส่วนของ Subprocess ในโมดูลการทดสอบ ซึ่งในซอฟต์แวร์ราปิดมายเนอร์ก็มีรูปแบบการทดสอบหลากหลายรูปแบบให้ผู้ใช้สามารถทำการเลือกใช้ได้ตามความเหมาะสม ยกตัวอย่างดังนี้

- Split Validation ทำการแบ่งข้อมูลส่วนหนึ่งเป็นข้อมูลเรียนรู้ และส่วนที่เหลือเป็นชุดข้อมูลทดสอบ
- X-Validation ทำงานเหมือนกับ Cross-validation ของซอฟต์แวร์เว็ก้าคือ ทำการแบ่งชุดข้อมูลจากชุดข้อมูลเรียนรู้ออกเป็นส่วนๆตามจำนวนที่ผู้ใช้กำหนด โดยมี 1 ส่วนเป็นชุดข้อมูลทดสอบ และส่วนที่เหลือเป็นชุดข้อมูลเรียนรู้ แล้วทำการเรียนรู้ข้อมูล ทำซ้ำเช่นนี้ไปเรื่อยๆ โดยการเปลี่ยนชุดข้อมูลทดสอบจนข้อมูลทุกส่วนเคยเป็นชุดข้อมูลทดสอบหมด เลือกโมเดลที่มีประสิทธิภาพมากที่สุด และนำไปทดสอบอีกครั้ง

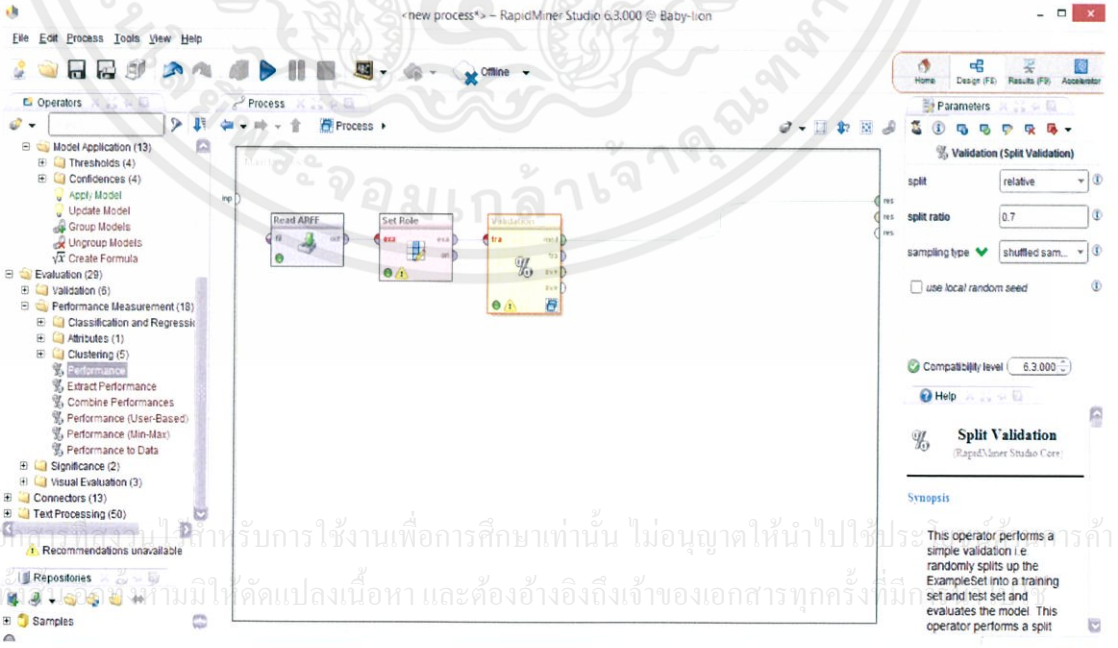
ซึ่งนอกเหนือจากนี้ หากผู้ใช้ต้องการทำการสร้างชุดข้อมูลทดสอบแบบ User training set หรือ Supplied test set ผู้ใช้สามารถทำการเลือกใช้โมดูล Read ARFF ตัวที่สองและเลือกนำเข้าไฟล์นั้นๆได้เลยเช่นกัน ในการตั้งค่าหรือต่อโมดูลทำหรับการทดสอบและวัดประสิทธิภาพของโมเดลสามารถทำได้ในโมดูลสำหรับการทดสอบเลยดังนี้

สำหรับโมดูลในการเลือกรูปแบบการทดสอบจะอยู่ในหมวดหมู่ Evaluation>Validation ในขั้นแรกผู้ใช้ทำการเลือกรูปแบบการทดสอบและลากโมดูลวางลงบนพื้นที่สำหรับ Process และสามารถทำการตั้งค่าพารามิเตอร์ต่างๆ ได้ทางด้านขวา จากนั้นทำการดับเบิ้ลคลิกโมดูลนั้นเพื่อทำการต่อโมดูล Subprocess ย่อยภายใน จะปรากฏหน้าต่างดังรูป

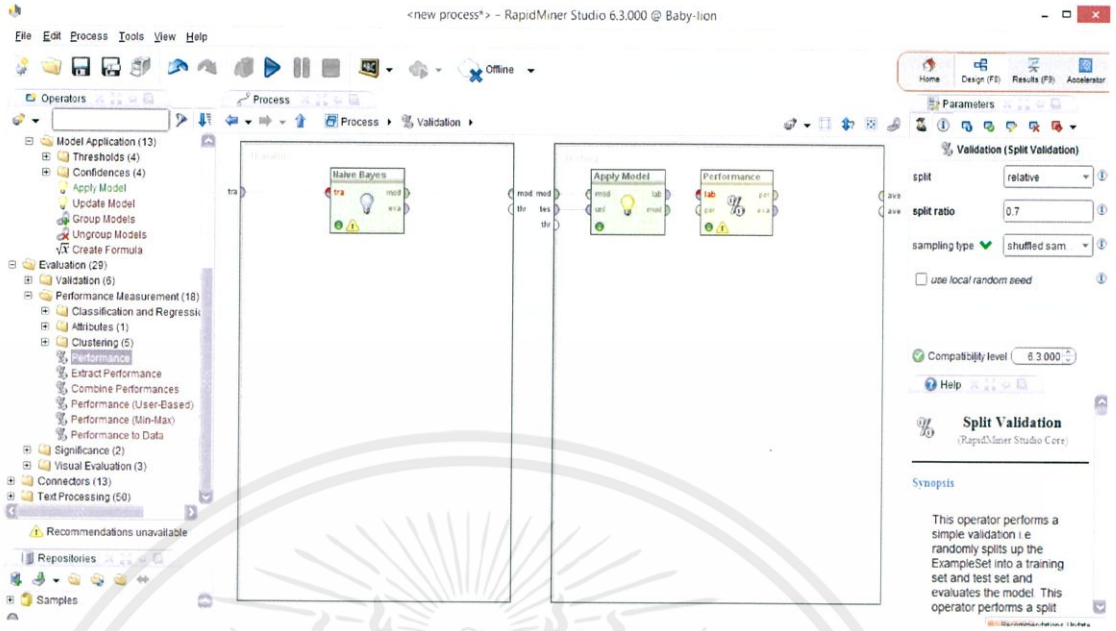


รูป 3.19 หน้าต่าง Subprocess ภายในโมดูล Split validation

ในส่วนพื้นที่ Training เราจะทำการเลือกวางโมดูลอัลกอริทึมสำหรับการเรียนรู้ลงไป และในส่วน Testing เราจะทำการวาง โมดูล Apply Model ซึ่งอยู่ในหมวดหมู่ Modeling>Model Application และโมดูล Performance ซึ่งอยู่ในหมวดหมู่ Evaluation>Performance Measurement สำหรับโมดูล Apply Model เราจะใช้สำหรับการทดสอบโมเดลที่ได้จากการเรียนรู้ และโมดูล Performance เราจะใช้ทำการวัดประสิทธิภาพของการทดสอบ สามารถต่อตามขั้นตอนข้างต้นได้ดังรูป

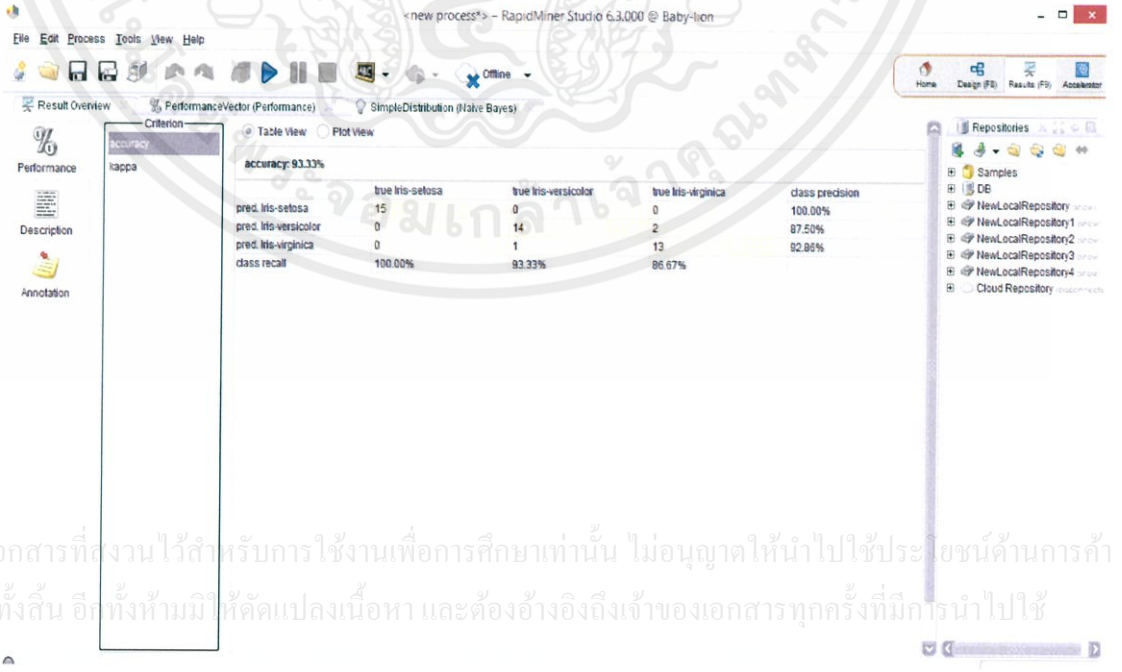


รูป 3.20 หน้าต่างในส่วน Main Process



รูป 3.21 หน้าต่างในส่วน Subprocess

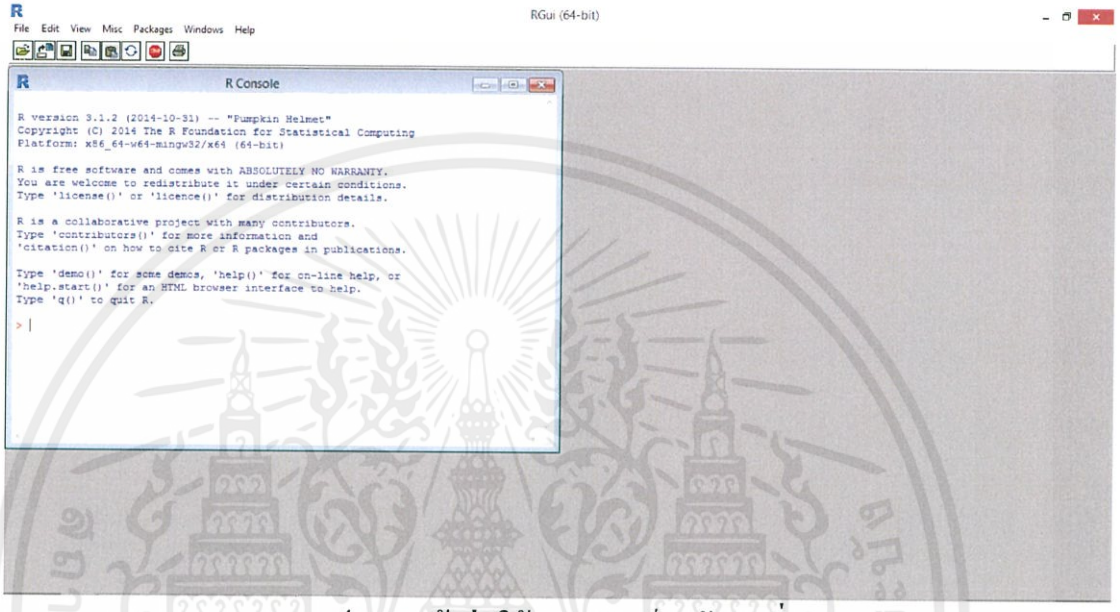
ซึ่งการต่อโมดูลดังกล่าวข้างต้นจะเห็นได้ว่าเป็นการต่อโมดูลที่ค่อนข้างจะไม่ซับซ้อนมากนัก ทั้งนี้ผู้ใช้งานสามารถทำการออกแบบ Process ที่มีความซับซ้อนมากยิ่งขึ้นได้ตามความเหมาะสมของการนำไปใช้ เมื่อทำการต่อโมดูลครบทุกส่วนและทำการตั้งค่าพารามิเตอร์เรียบร้อยแล้ว ผู้ใช้ทำการกดรัน Process จะปรากฏหน้าต่างแสดงผลตามที่ผู้ใช้ต่อโมดูลไว้ สำหรับตัวอย่างข้างต้นจะแสดงผลลัพธ์ดังรูป



รูป 3.22 หน้าต่างแสดงผล

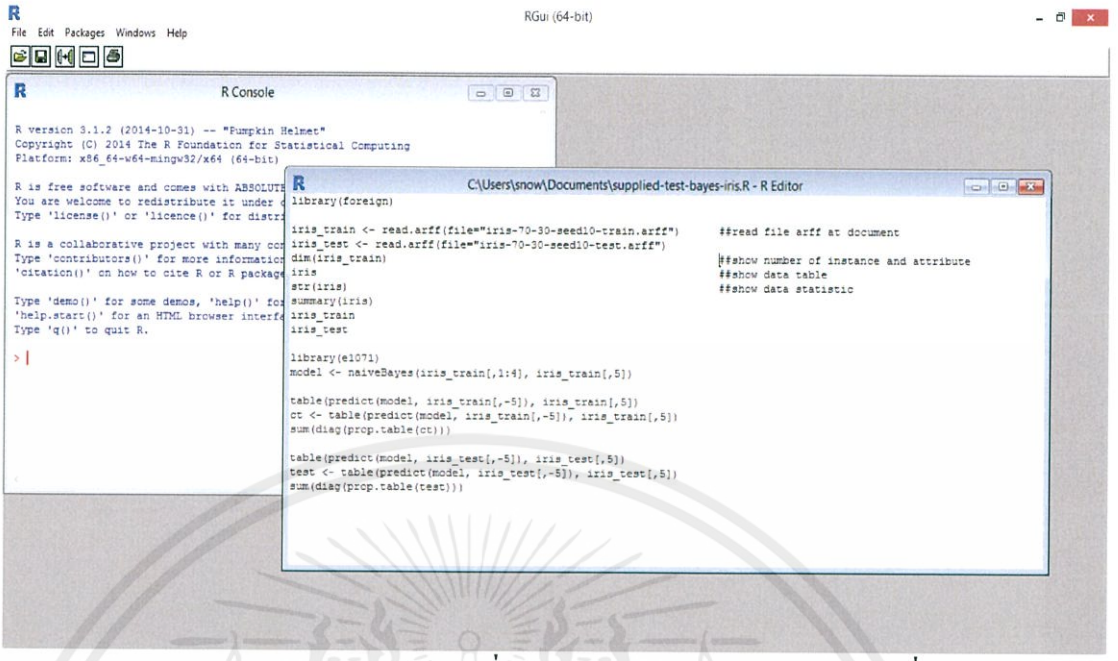
3.2.3 การสร้างโมเดลทำนายค่าในซอฟต์แวร์อาร์ ดาต้ามายนิ่ง

สำหรับการทำเหมืองข้อมูลด้วยอาร์ ดาต้ามายนิ่งนั้น หน้าต่างสำหรับผู้ใช้งานจะไม่เป็น GUI เหมือนซอฟต์แวร์ 2 ตัวที่ผ่านมา แต่จะเป็นการเขียนคำสั่งและรันคำสั่งที่ละบรรทัดคล้าย Command line ดังรูป



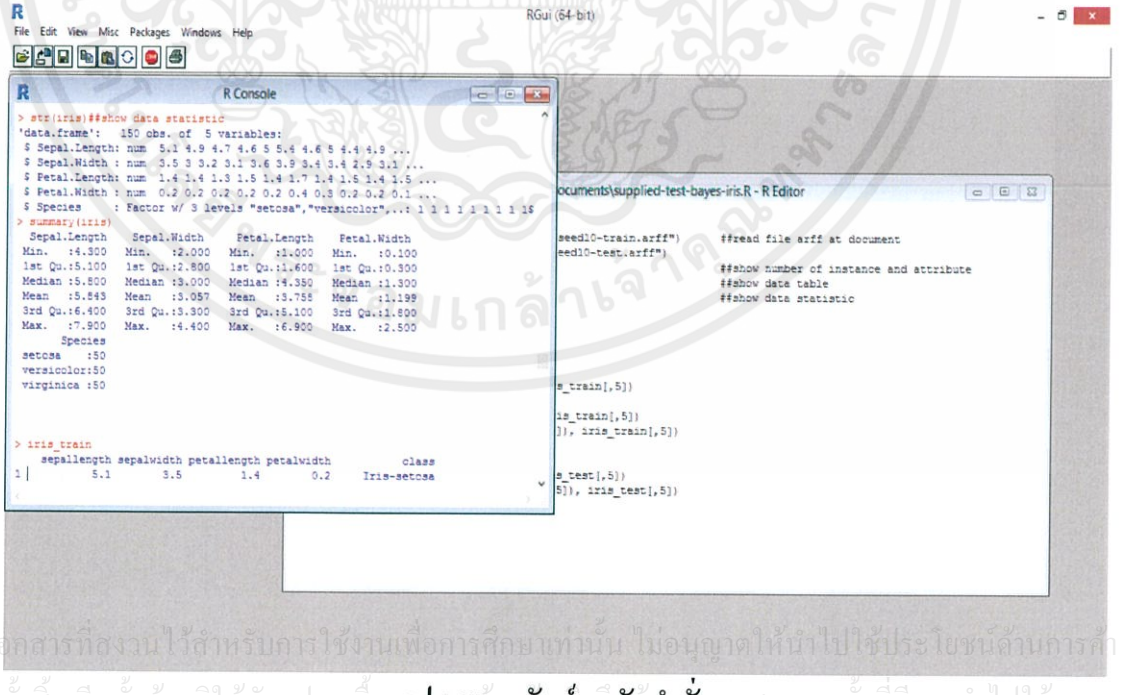
รูป 3.23 หน้าต่างใช้งานของอาร์ ดาต้ามายนิ่ง

ซึ่งจะปรากฏหน้าต่าง R Console สำหรับการรันคำสั่ง สำหรับวิธีการใช้งานผู้ใช้สามารถทำการสร้าง Script เพื่อทำการเขียนคำสั่งและบันทึกคำสั่งเก็บไว้ใช้งานได้ โดยทำการคลิกที่ File>New script จะปรากฏหน้าต่าง R Editor ดังรูป หลังจากนั้นผู้ใช้สามารถทำการเขียนคำสั่งให้มีการนำเข้าข้อมูล ประมวลผลข้อมูล และแสดงผลลัพธ์ข้อมูลลงใน R Editor นี้ได้ ซึ่งการเขียนคำสั่งต่างๆ นั้นจะต้องทำการศึกษาวิธีการเขียนคำสั่งต่างๆอย่างถูกต้องต่อไป บางคำสั่งอาจจะต้องมีการติดตั้งไลบรารีเฉพาะของคำสั่งนั้นๆ ดังรูป



รูป 3.24 ตัวอย่างการเขียนคำสั่งการทำเหมืองข้อมูลในอาร์ ดาต้ามายนิ่ง

เมื่อเราทำการเขียนโค้ดคำสั่งเรียบร้อยแล้ว ต่อไปจะเป็นขั้นตอนวิธีการรันคำสั่งที่เราเขียน สามารถทำได้โดยการคลิกเมาส์ลากครอบคำสั่งในบรรทัดที่เราต้องการรัน และคลิกไอคอน Run line or selection หลังจากนั้นผลลัพธ์การรันจะแสดงในหน้าต่าง R Console ดังรูป



รูป 3.25 ผลลัพธ์การรันคำสั่ง

ซึ่งในส่วนของการแสดงผลหรือการกำหนดค่าพารามิเตอร์ต่างๆ นั้นผู้ใช้จะต้องทำการศึกษาคำสั่งนั้นๆ เป็นกรณีไปตามแต่ผู้ใช้ต้องการ

3.3 การออกแบบการทดลองอัลกอริทึมที่เหมาะสมกับชุดข้อมูลแต่ละแบบ

ในการทดลองศึกษาหาความเหมาะสมในการเลือกใช้อัลกอริทึมกับแต่ละชุดข้อมูลแต่ละงานที่เรามีอยู่ เพื่อให้ได้การเรียนรู้ที่มีประสิทธิภาพมากที่สุด โดยจากการศึกษาซอฟต์แวร์ที่ใช้ในการทำเหมืองข้อมูลทั้ง 3 ซอฟต์แวร์ ได้แก่ เวก้า (Weka), ราปิดมายเนอร์ (RapidMiner) และอาร์ คาต้ามายนิ่ง (R Data mining) พบว่าซอฟต์แวร์เวก้าจะมีอัลกอริทึมหนึ่งที่สามารถใช้เป็นมาตรฐาน (Baseline) ได้ ซึ่งก็คืออัลกอริทึมซีโร่อาร์ (ZeroR) ซึ่งเป็นผลลัพธ์เปอร์เซ็นต์ความถูกต้องขั้นต่ำสุดที่ควรจะได้จากการทดลองในกรณีนั้นๆ ตัวอย่างเช่นหากในการทดลองกับชุดข้อมูลดอกไอริส (Iris.arff)ด้วยอัลกอริทึมซีโร่อาร์ และใช้วิธีการทดสอบแบบใช้ข้อมูลทดสอบเดียวกันกับชุดข้อมูลเรียนรู้ (Use training set) แล้วได้เปอร์เซ็นต์ความถูกต้องเท่ากับ 33.33 เปอร์เซ็นต์ การเลือกใช้อัลกอริทึมอื่นๆ ที่จะนำมาใช้สร้างโมเดลควรจะได้ผลลัพธ์เปอร์เซ็นต์ความถูกต้องที่มากกว่าค่าที่ได้จากค่ามาตรฐานนั่นเอง แต่ในซอฟต์แวร์ราปิดมายเนอร์และอาร์ คาต้ามายนิ่งจากการศึกษาวิธีการใช้งานรวมถึงคู่มือต่างๆ ยังไม่พบว่ามีอัลกอริทึมใดที่สามารถใช้เป็นมาตรฐานได้ ดังนั้นในการศึกษาวิเคราะห์อัลกอริทึมที่เหมาะสมกับข้อมูลแต่ละรูปแบบจึงจะใช้ซอฟต์แวร์เวก้าในการทดลอง โดยขั้นตอนในการทดลองจะทำการหาค่ามาตรฐานก่อนเป็นอันดับแรก เมื่อได้ค่ามาตรฐานที่ต้องการของแต่ละชุดข้อมูลแล้วขั้นตอนต่อไปจะทำการเลือกใช้อัลกอริทึมที่สนใจหรือเป็นที่นิยมใช้งานมาทำการเรียนรู้เพื่อสร้างโมเดล และสังเกตโมเดลที่ได้กับเปอร์เซ็นต์ความถูกต้องที่ได้จากการทดสอบ หากอัลกอริทึมใดที่ได้ค่าความถูกต้องน้อยกว่าค่ามาตรฐานเราจะไม่นำอัลกอริทึมนั้นมาใช้กับชุดข้อมูลดังกล่าว ในส่วนของอัลกอริทึมที่มีค่าความถูกต้องมากกว่าค่ามาตรฐานให้นำอัลกอริทึมเหล่านั้นมาเปรียบเทียบเพื่อหาอัลกอริทึมที่เหมาะสมกับชุดข้อมูลต่อไป โดยในการทดลองจะใช้ชุดข้อมูลทั้ง 7 ชุด ที่มีความแตกต่างกันทั้งในด้านขนาดของข้อมูล จำนวนแอตทริบิวต์ และประเภทของข้อมูล กับอัลกอริทึมที่นิยมในหมวดหมู่ที่แตกต่างกันทั้งหมด 6 อัลกอริทึม ดังนี้

- Rules>ZeroR
- Bayes>NaiveBayes
- Lazy>IBk
- Meta>AdaBoostM1
- Rules>OneR
- Trees>J48

ทำการเปรียบเทียบผลลัพธ์ที่ได้ในแต่ละชุดข้อมูล และทำการวิเคราะห์ผล

3.4 การออกแบบการทดลองเปรียบเทียบการแบ่งชุดข้อมูลที่แตกต่างกัน

ในการทดลองในด้านการเปรียบเทียบประสิทธิภาพของการแบ่งชุดข้อมูลที่แตกต่างกัน เป็นส่วนของการตั้งค่าในส่วนของการทดสอบโมเดล ในการเลือกรูปแบบการทดสอบแบบ Split Validation ผู้ใช้สามารถทำการกำหนดอัตราส่วนระหว่างชุดข้อมูลเรียนรู้กับชุดข้อมูลทดสอบได้อย่างอิสระ ดังนั้นในส่วนนี้จะทำการทดลองหาว่าหากมีการแบ่งชุดข้อมูลเรียนรู้ต่อชุดข้อมูลทดสอบในอัตราส่วนที่ไม่เท่ากันผลลัพธ์จะมีค่าเท่ากันหรือแตกต่างกันอย่างไร หากแตกต่างกันแบ่งในอัตราส่วนเท่าไหร่จึงจะเหมาะสมที่สุด

ในการทดลองนี้จะทำการทดลองในทั้ง 3 ซอฟต์แวร์ และทดลองด้วยอัลกอริทึม 3 อัลกอริทึม ได้แก่ Naïve Bayes, Decision Tree และ AdaBoost กับชุดข้อมูลทั้ง 7 ชุด ด้วยการใช้ Random seed ที่แตกต่างกัน 10 ครั้ง โดยการแบ่งชุดข้อมูลเราจะทำการแบ่งชุดข้อมูลออกเป็น 3 แบบด้วยกัน คือ

- แบ่งเป็นชุดข้อมูลเรียนรู้ 90% และชุดข้อมูลทดสอบ 10%
- แบ่งเป็นชุดข้อมูลเรียนรู้ 80% และชุดข้อมูลทดสอบ 20%
- แบ่งเป็นชุดข้อมูลเรียนรู้ 70% และชุดข้อมูลทดสอบ 30%

หลังจากนั้นทำการหาค่าเฉลี่ยในแต่ละชุดข้อมูลแต่ละอัลกอริทึมในแต่ละรูปแบบการแบ่งชุดข้อมูล เพื่อนำผลลัพธ์ที่ได้มาทำการเปรียบเทียบวิเคราะห์ต่อไป

แต่เนื่องจากในการเปรียบเทียบด้านนี้จะต้องทำการควบคุมชุดข้อมูลที่จะนำไปใช้ในแต่ละซอฟต์แวร์ให้เหมือนกัน ดังนั้นทางผู้ศึกษาจะทำการใช้ซอฟต์แวร์อาร์ ดาต้ามายนิ่งเป็นเครื่องมือในการแบ่งชุดข้อมูล และนำข้อมูลที่ได้จากการแบ่งไปใช้กับซอฟต์แวร์อื่นๆต่อไป เนื่องจากในการใช้การทดสอบแบบ Split validation ของซอฟต์แวร์อาร์ ดาต้ามายนิ่ง เราสามารถเขียนคำสั่งให้มีการแสดงข้อมูลในส่วนของทั้งชุดข้อมูลเรียนรู้และชุดข้อมูลทดสอบที่ทำการแบ่งแล้วได้ พร้อมทั้งมีการแสดงบรรทัดของข้อมูลที่ทำการแบ่ง ทำให้ง่ายต่อการบันทึกข้อมูลเป็นไฟล์สกุล ARFF เพื่อนำไปใช้กับซอฟต์แวร์อื่นๆในขณะที่ซอฟต์แวร์ราปิดมายเนอร์สามารถแสดงชุดข้อมูลเรียนรู้และชุดข้อมูลทดสอบได้แต่ไม่แสดงบรรทัดของข้อมูลเดิมทำให้ผู้ใช้ไม่สามารถนำมาทำเป็นไฟล์สกุล ARFF ได้ และซอฟต์แวร์เวก้าไม่แสดงชุดข้อมูลเรียนรู้และชุดข้อมูลทดสอบที่ทำการแบ่งแล้วนั่นเอง

3.5 การออกแบบการทดลองเปรียบเทียบประสิทธิภาพของแต่ละซอฟต์แวร์

ในการทดลองเปรียบเทียบประสิทธิภาพของซอฟต์แวร์แต่ละตัวเราจะทำการทดสอบโดยการทำเหมือนข้อมูลกับชุดข้อมูลทั้ง 7 ชุดและอัลกอริทึมทั้ง 3 อัลกอริทึม โดยใช้เพียงการแบ่งชุดข้อมูลแบบแบ่งเป็นชุดข้อมูลเรียนรู้ 90% และชุดข้อมูลทดสอบ 10% โดยทำการเปรียบเทียบในแต่ละชุดข้อมูลแต่

ละอัลกอริทึมในแต่ละซอฟต์แวร์เพื่อดูประสิทธิภาพที่ได้โดยหาเป็นค่าเฉลี่ยของการ Random seed ทั้ง 10 ครั้ง

บทที่ 4

การทดลองและผลการทดลอง

4.1 การควบคุมชุดข้อมูล

ในการทดลองเพื่อเปรียบเทียบประสิทธิภาพของซอฟต์แวร์ในการทำเหมืองข้อมูลทั้ง 3 ซอฟต์แวร์ ทำให้เราต้องมีการควบคุมชุดข้อมูลให้เหมือนกันในการนำไปใช้ โดยเปรียบเทียบในแต่ละอัลกอริทึมที่เหมือนกัน รวมไปถึงการเปรียบเทียบในด้านของประสิทธิภาพในการแบ่งชุดข้อมูลที่แตกต่างกัน

4.1.1 การแบ่งชุดข้อมูล

ผู้ศึกษาจะมีการแบ่งชุดข้อมูลออกเป็น 3 รูปแบบ ดังนี้

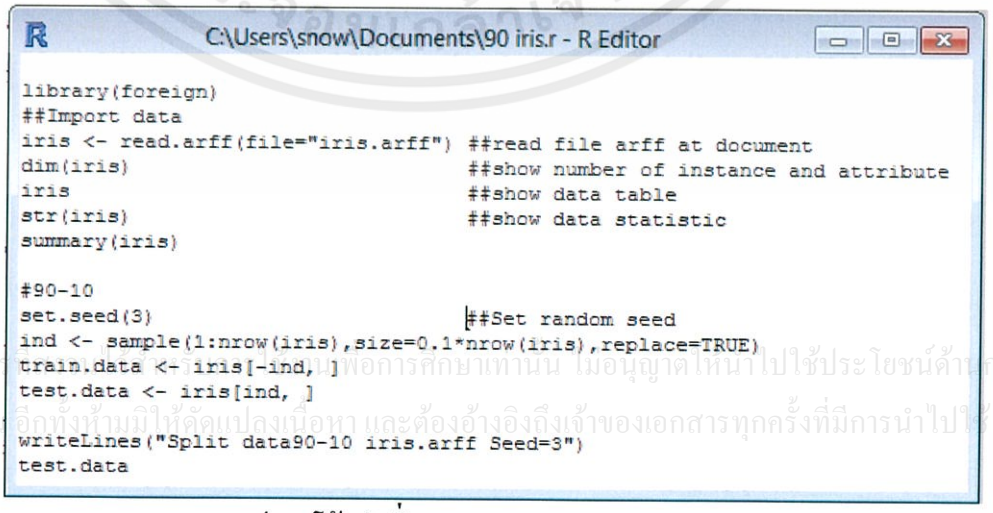
- แบ่งเป็นชุดข้อมูลเรียนรู้ 90% และชุดข้อมูลทดสอบ 10%
- แบ่งเป็นชุดข้อมูลเรียนรู้ 80% และชุดข้อมูลทดสอบ 20%
- แบ่งเป็นชุดข้อมูลเรียนรู้ 70% และชุดข้อมูลทดสอบ 30%

ซึ่งในแต่ละชุดข้อมูลจะมีจำนวนข้อมูลที่ไม่เท่ากันในการแบ่งแต่ละแบบ และเป็นการแบ่งโดยใช้ซอฟต์แวร์อาร์ ดาต้ามายนิ่ง เป็นเครื่องมือในการแบ่งชุดข้อมูล

4.1.1.1 การแบ่งชุดข้อมูลดอกไอริส (Iris.arff)

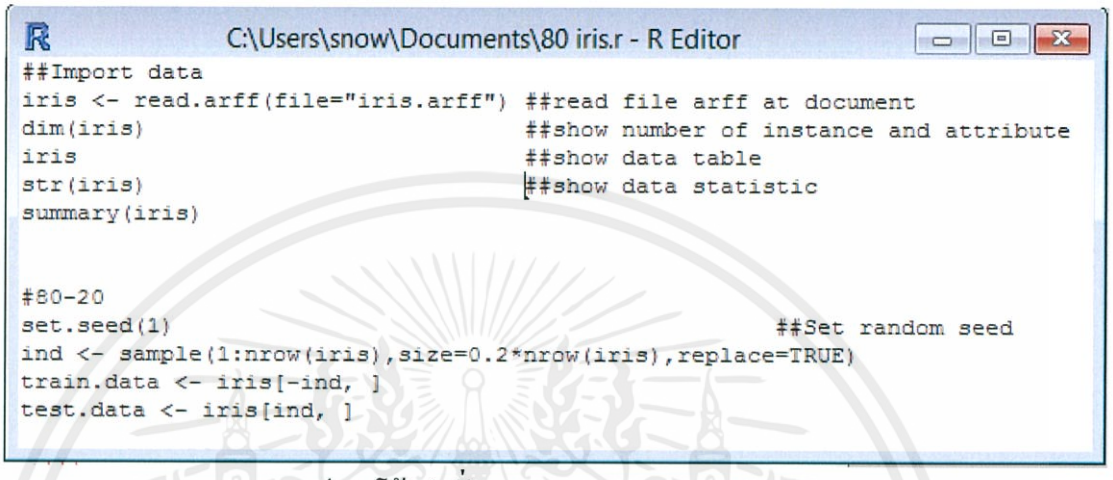
ชุดข้อมูลดอกไอริสประกอบด้วยข้อมูลทั้งหมด 150 instances

การแบ่งชุดข้อมูลดอกไอริสแบบแบ่งเป็นชุดข้อมูลเรียนรู้ 90% และชุดข้อมูลทดสอบ 10% จะได้ชุดข้อมูลทดสอบจำนวน 15 instances และชุดข้อมูลเรียนรู้ 135-138 instances โดยสามารถเขียนคำสั่งได้ดังนี้



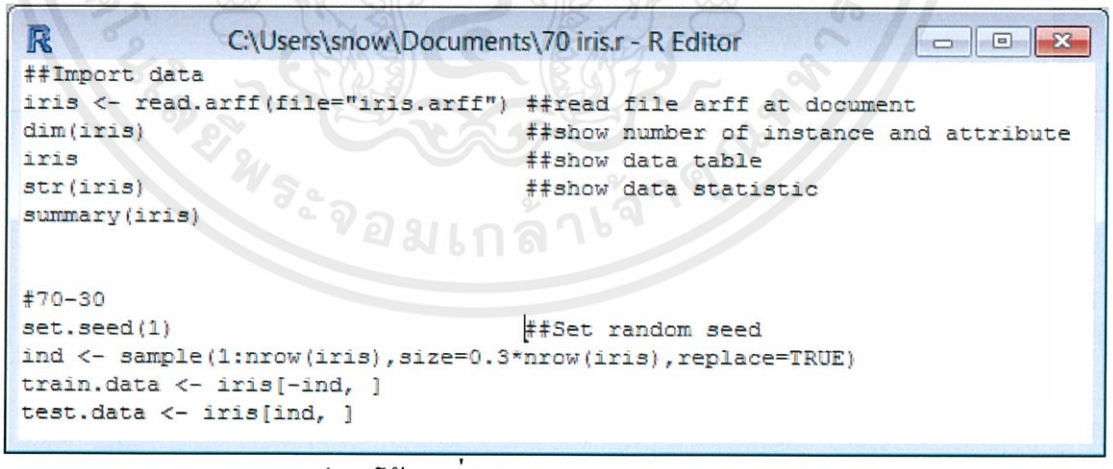
รูป 4.1 โค้ดคำสั่ง Split data Iris.arff แบบ 90-10

การแบ่งชุดข้อมูลดอกไอริสแบบแบ่งเป็นชุดข้อมูลเรียนรู้ 80% และชุดข้อมูลทดสอบ 20% จะได้ชุดข้อมูลทดสอบจำนวน 30 instances และชุดข้อมูลเรียนรู้ 121-126 instances โดยสามารถเขียนคำสั่งได้ดังนี้



รูป 4.2 โค้ดคำสั่ง Split data Iris.arff แบบ 80-20

การแบ่งชุดข้อมูลดอกไอริสแบบแบ่งเป็นชุดข้อมูลเรียนรู้ 70% และชุดข้อมูลทดสอบ 30% จะได้ชุดข้อมูลทดสอบจำนวน 45 instance และชุดข้อมูลเรียนรู้ 109-114 instance โดยสามารถเขียนคำสั่งได้ดังนี้

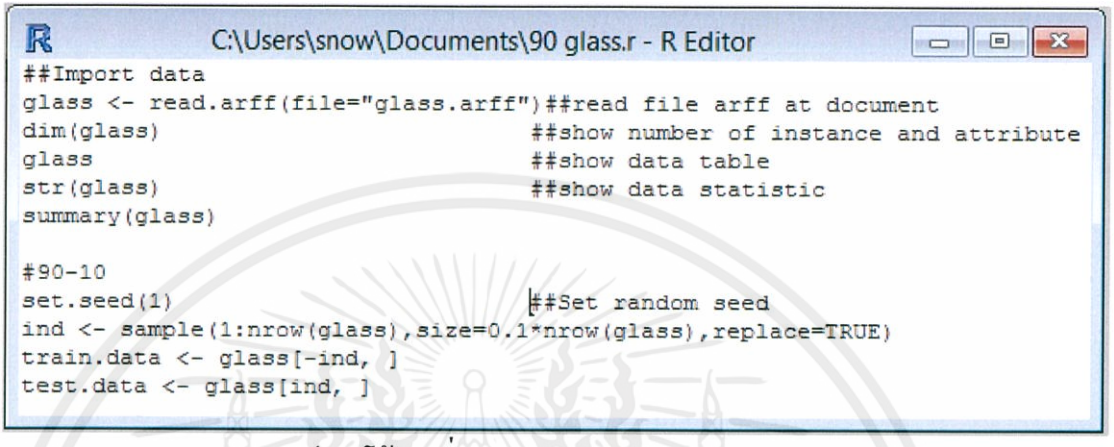


รูป 4.3 โค้ดคำสั่ง Split data Iris.arff แบบ 70-30

4.1.1.2 การแบ่งชุดข้อมูลกระจก (Glass.arff)

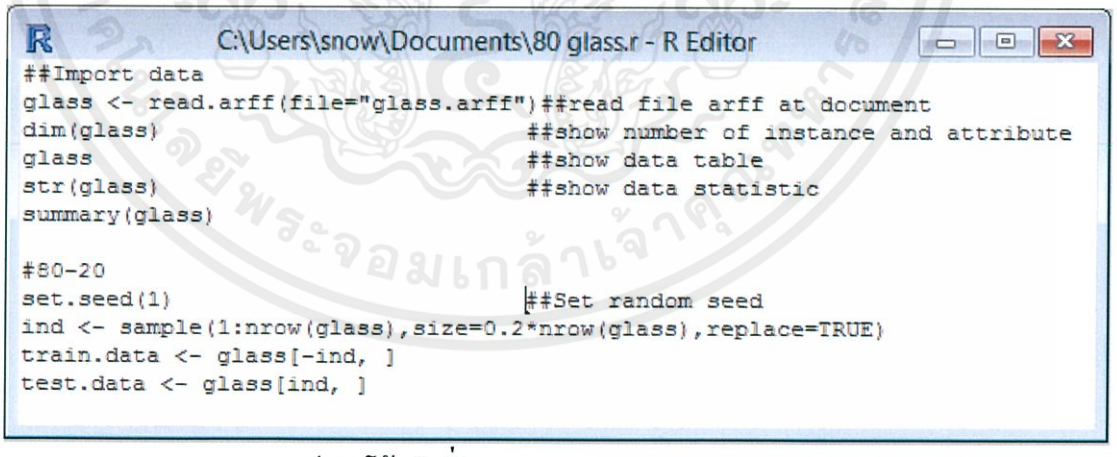
ชุดข้อมูลกระจกประกอบด้วยข้อมูลทั้งหมด 214 instances

การแบ่งชุดข้อมูลกระจกแบบแบ่งเป็นชุดข้อมูลเรียนรู้ 90% และชุดข้อมูลทดสอบ 10% จะได้ชุดข้อมูลทดสอบจำนวน 21 instances และชุดข้อมูลเรียนรู้ 193-196 instances โดยสามารถเขียนคำสั่งได้ดังนี้



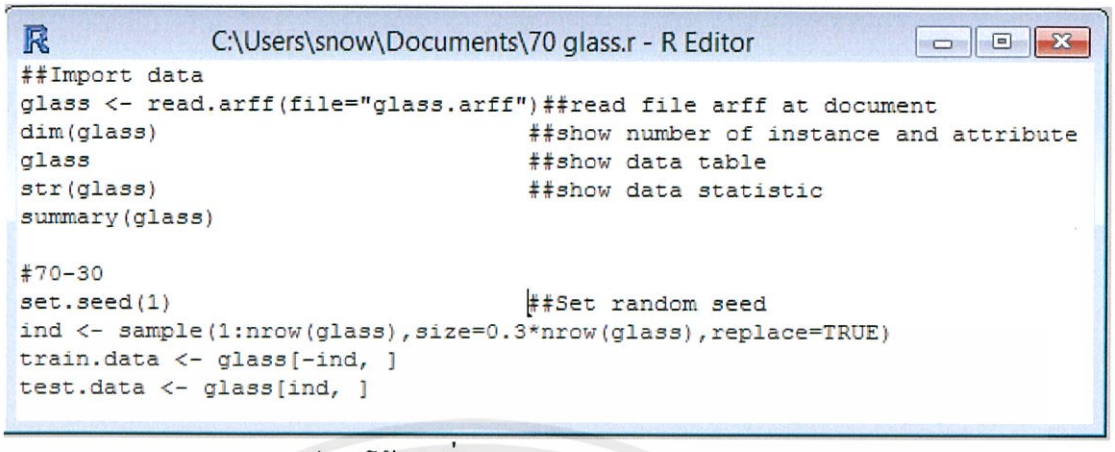
รูป 4.4 โค้ดคำสั่ง Split data Glass.arff แบบ 90-10

การแบ่งชุดข้อมูลกระจกแบบแบ่งเป็นชุดข้อมูลเรียนรู้ 80% และชุดข้อมูลทดสอบ 20% จะได้ชุดข้อมูลทดสอบจำนวน 42 instances และชุดข้อมูลเรียนรู้ 173-180 instances โดยสามารถเขียนคำสั่งได้ดังนี้



รูป 4.5 โค้ดคำสั่ง Split data Glass.arff แบบ 80-20

การแบ่งชุดข้อมูลกระจกแบบแบ่งเป็นชุดข้อมูลเรียนรู้ 70% และชุดข้อมูลทดสอบ 30% จะได้ชุดข้อมูลทดสอบจำนวน 64 instances และชุดข้อมูลเรียนรู้ 154-163 instances โดยสามารถเขียนคำสั่งได้ดังนี้

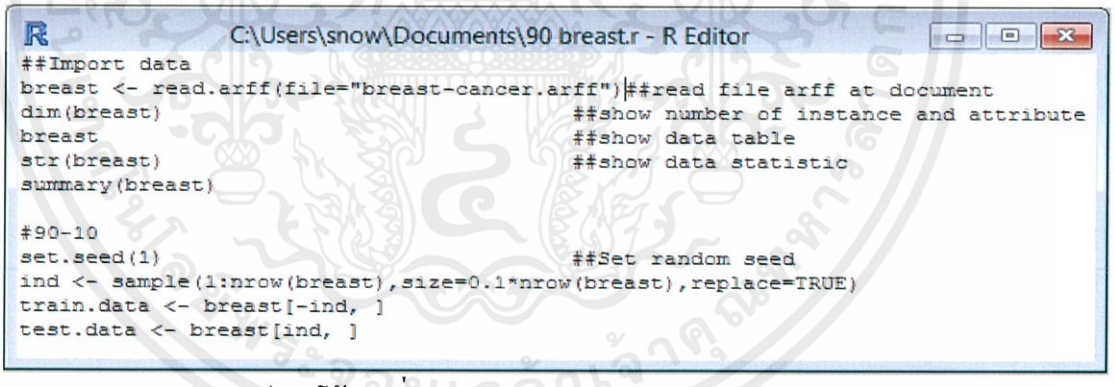


รูป 4.6 โค้ดคำสั่ง Split data Glass.arff แบบ 70-30

4.1.1.3 การแบ่งชุดข้อมูลมะเร็งเต้านม (Breast-cancer.arff)

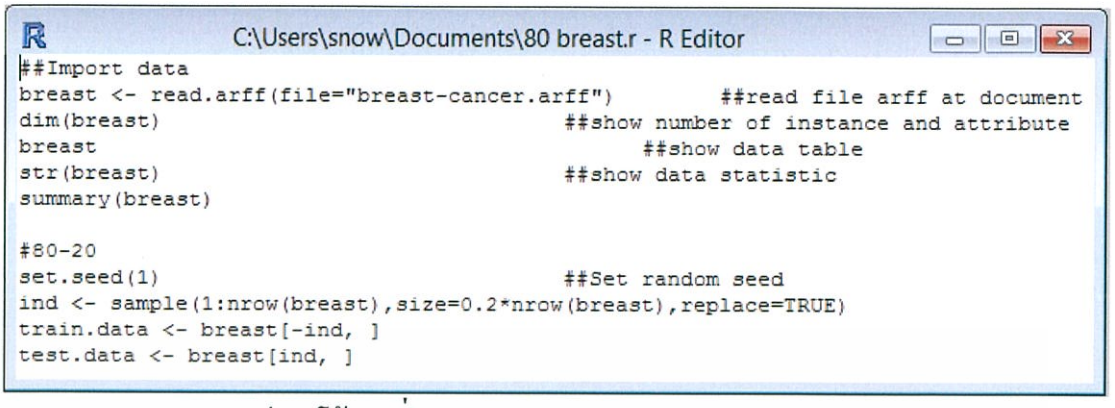
ชุดข้อมูลมะเร็งเต้านมประกอบด้วยข้อมูลทั้งหมด 286 instance

การแบ่งชุดข้อมูลมะเร็งเต้านมแบบแบ่งเป็นชุดข้อมูลเรียนรู้ 90% และชุดข้อมูลทดสอบ 10% จะได้ชุดข้อมูลทดสอบจำนวน 28 instances และชุดข้อมูลเรียนรู้ 259-261 instances โดยสามารถเขียนคำสั่งได้ดังนี้



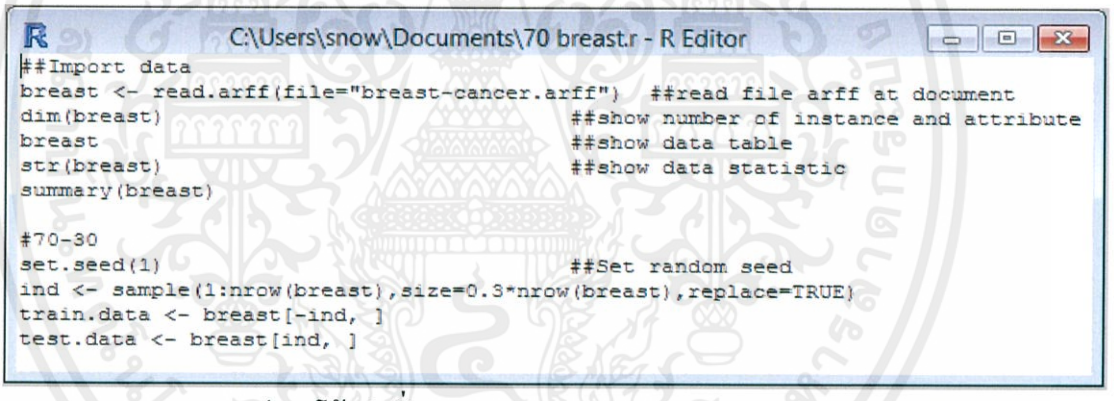
รูป 4.7 โค้ดคำสั่ง Split data Breast-cancer.arff แบบ 90-10

การแบ่งชุดข้อมูลมะเร็งเต้านมแบบแบ่งเป็นชุดข้อมูลเรียนรู้ 80% และชุดข้อมูลทดสอบ 20% จะได้ชุดข้อมูลทดสอบจำนวน 57 instances และชุดข้อมูลเรียนรู้ 231-236 instances โดยสามารถเขียนคำสั่งได้ดังนี้



รูป 4.8 โค้ดคำสั่ง Split data Breast-cancer.arff แบบ 80-20

การแบ่งชุดข้อมูลมะเร็งเต้านมแบบแบ่งเป็นชุดข้อมูลเรียนรู้ 70% และชุดข้อมูลทดสอบ 30% จะได้ชุดข้อมูลทดสอบจำนวน 85 instances และชุดข้อมูลเรียนรู้ 208-216 instances โดยสามารถเขียนคำสั่งได้ดังนี้

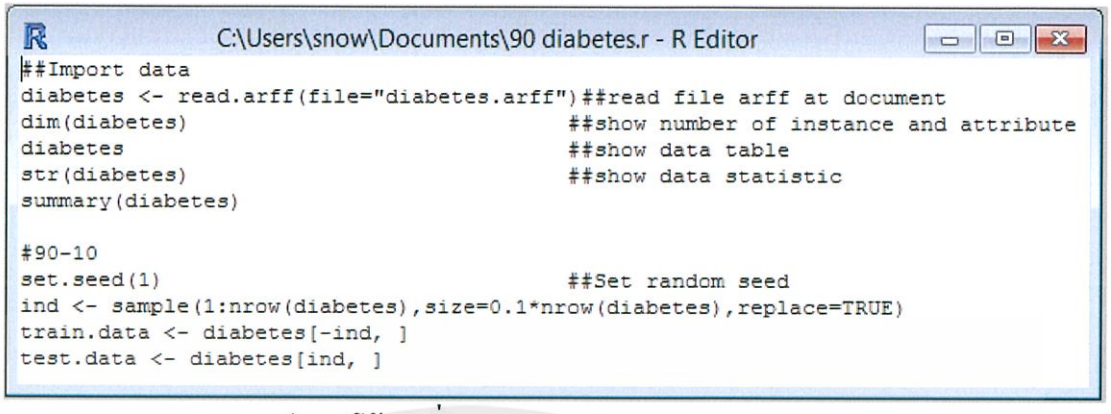


รูป 4.9 โค้ดคำสั่ง Split data Breast-cancer.arff แบบ 70-30

4.1.1.4 การแบ่งชุดข้อมูลเบาหวาน (Diabetes.arff)

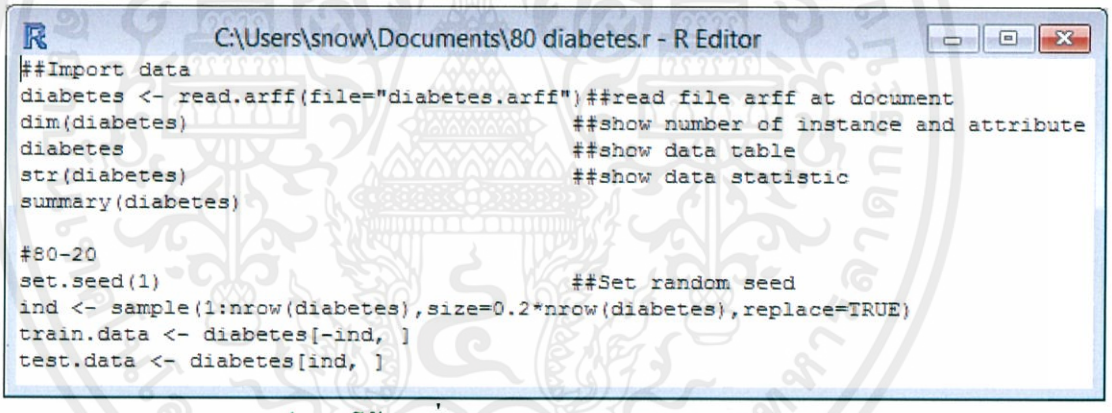
ชุดข้อมูลเบาหวานประกอบด้วยข้อมูลทั้งหมด 768 instances

การแบ่งชุดข้อมูลเบาหวานแบบแบ่งเป็นชุดข้อมูลเรียนรู้ 90% และชุดข้อมูลทดสอบ 10% จะได้ชุดข้อมูลทดสอบจำนวน 76 instances และชุดข้อมูลเรียนรู้ 693-698 instances โดยสามารถเขียนคำสั่งได้ดังนี้



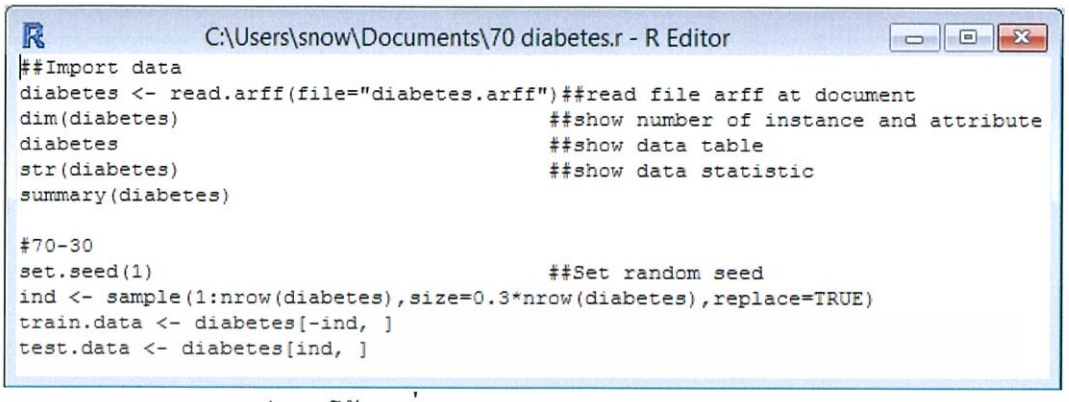
รูป 4.10 โค้ดคำสั่ง Split data Diabetes.arff แบบ 90-10

การแบ่งชุดข้อมูลเบาหวานแบบแบ่งเป็นชุดข้อมูลเรียนรู้ 80% และชุดข้อมูลทดสอบ 20% จะได้ชุดข้อมูลทดสอบจำนวน 153 instances และชุดข้อมูลเรียนรู้ 622-637 instances โดยสามารถเขียนคำสั่งได้ดังนี้



รูป 4.11 โค้ดคำสั่ง Split data Diabetes.arff แบบ 80-20

การแบ่งชุดข้อมูลเบาหวานแบบแบ่งเป็นชุดข้อมูลเรียนรู้ 70% และชุดข้อมูลทดสอบ 30% จะได้ชุดข้อมูลทดสอบจำนวน 230 instances และชุดข้อมูลเรียนรู้ 568-578 instances โดยสามารถเขียนคำสั่งได้ดังนี้

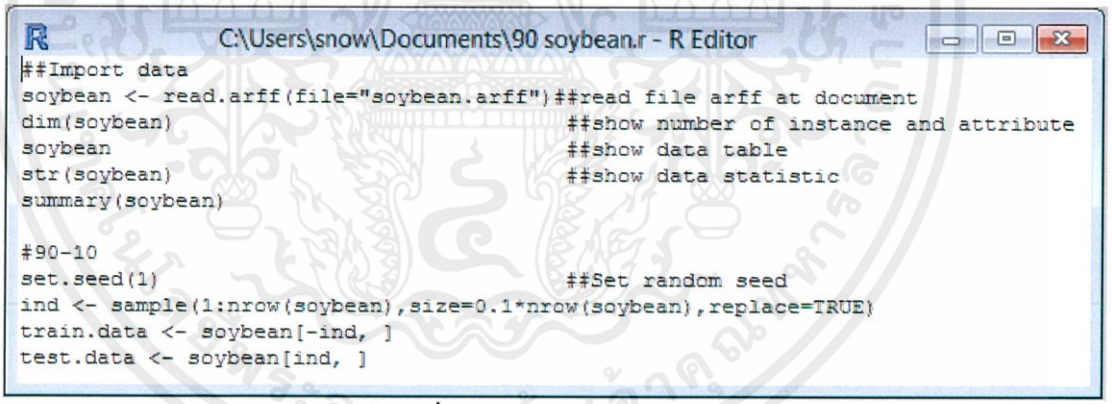


รูป 4.12 โค้ดคำสั่ง Split data Diabetes.arff แบบ 70-30

4.1.1.5 การแบ่งชุดข้อมูลถั่วเหลือง (Soybean.arff)

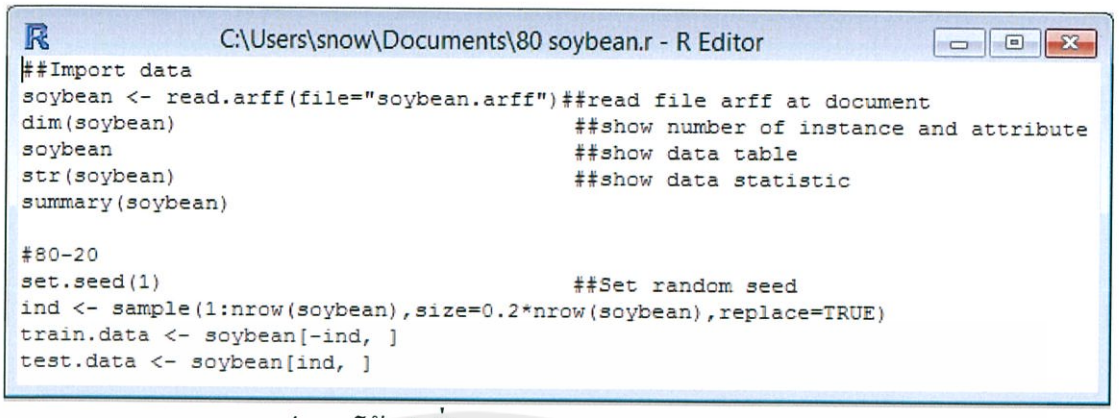
ชุดข้อมูลถั่วเหลืองประกอบด้วยข้อมูลทั้งหมด 683 instances

การแบ่งชุดข้อมูลถั่วเหลืองแบบแบ่งเป็นชุดข้อมูลเรียนรู้ 90% และชุดข้อมูลทดสอบ 10% จะได้ชุดข้อมูลทดสอบจำนวน 68 instances และชุดข้อมูลเรียนรู้ 616-620 instances โดยสามารถเขียนคำสั่งได้ดังนี้



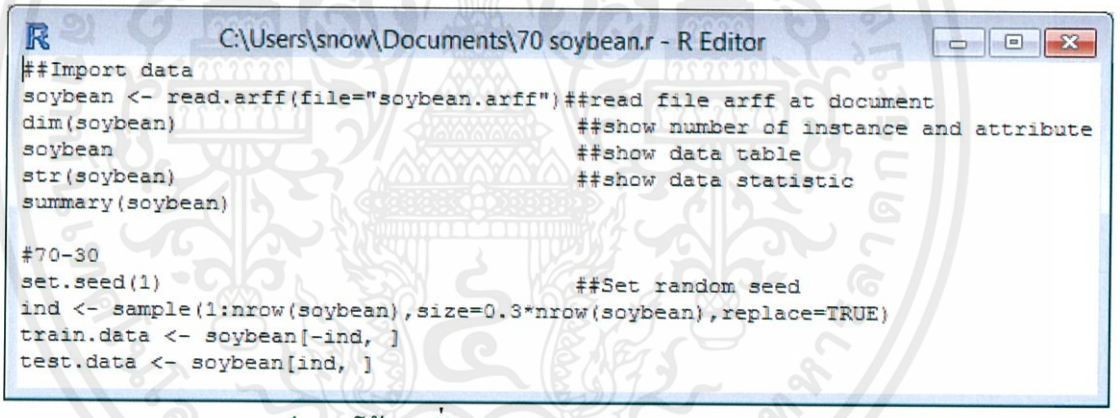
รูป 4.13 โค้ดคำสั่ง Split data Soybean.arff แบบ 90-10

การแบ่งชุดข้อมูลถั่วเหลืองแบบแบ่งเป็นชุดข้อมูลเรียนรู้ 80% และชุดข้อมูลทดสอบ 20% จะได้ชุดข้อมูลทดสอบจำนวน 136 instances และชุดข้อมูลเรียนรู้ 553-566 instances โดยสามารถเขียนคำสั่งได้ดังนี้



รูป 4.14 โค้ดคำสั่ง Split data Soybean.arff แบบ 80-20

การแบ่งชุดข้อมูลถั่วเหลืองแบบแบ่งเป็นชุดข้อมูลเรียนรู้ 70% และชุดข้อมูลทดสอบ 30% จะได้ชุดข้อมูลทดสอบจำนวน 204 instances และชุดข้อมูลเรียนรู้ 499-516 instances โดยสามารถเขียนคำสั่งได้ดังนี้

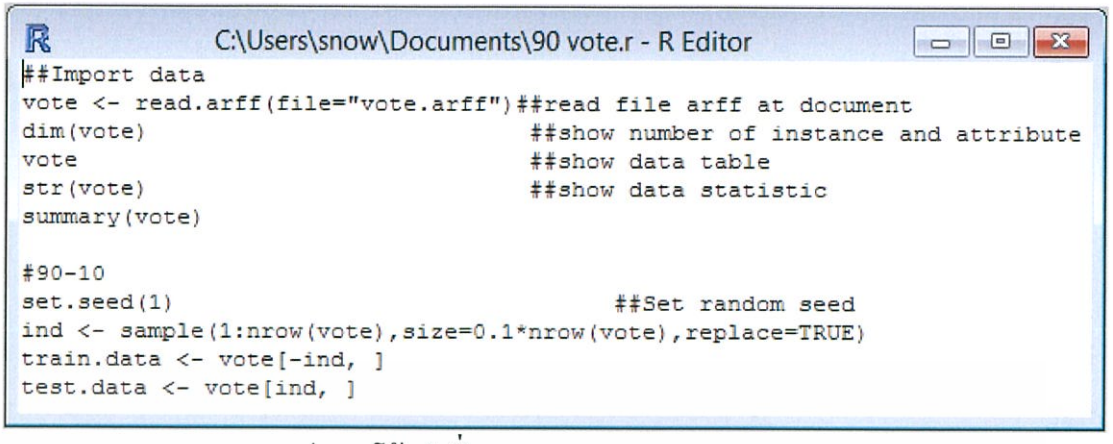


รูป 4.15 โค้ดคำสั่ง Split data Soybean.arff แบบ 70-30

4.1.1.6 การแบ่งชุดข้อมูลการลงคะแนนเสียงเลือกตั้ง (Vote.arff)

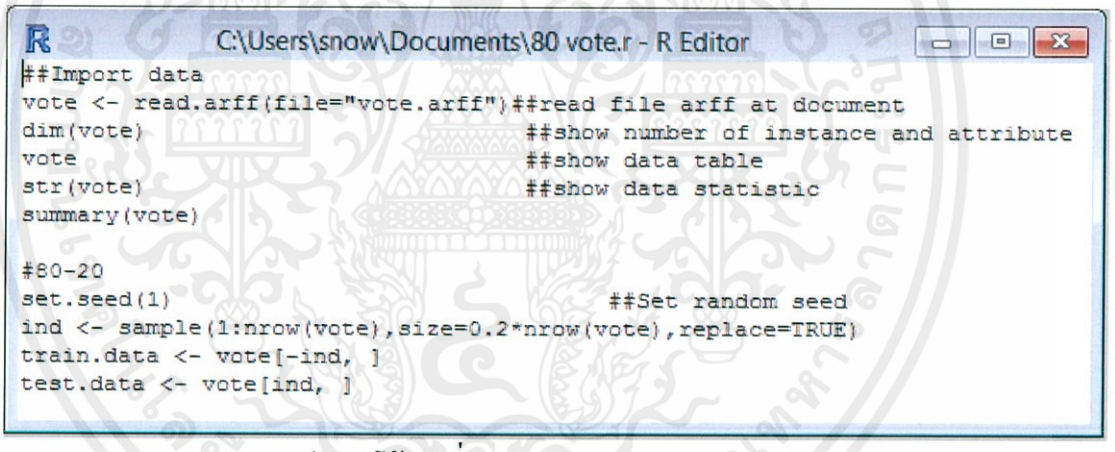
ชุดข้อมูลการลงคะแนนเสียงเลือกตั้งประกอบด้วยข้อมูลทั้งหมด 435 instances

การแบ่งชุดข้อมูลการลงคะแนนเสียงเลือกตั้งแบบแบ่งเป็นชุดข้อมูลเรียนรู้ 90% และชุดข้อมูลทดสอบ 10% จะได้ชุดข้อมูลทดสอบจำนวน 43 instances และชุดข้อมูลเรียนรู้ 378-396 instances โดยสามารถเขียนคำสั่งได้ดังนี้



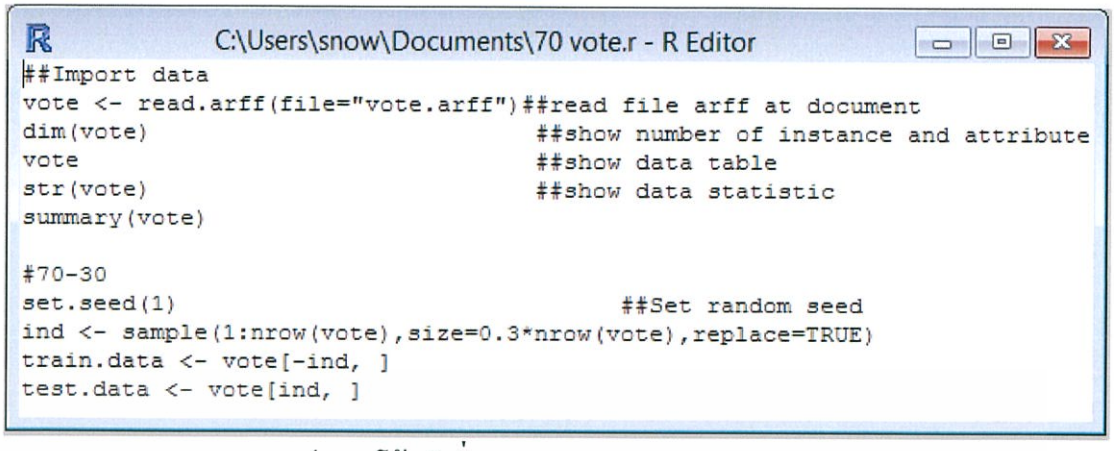
รูป 4.16 โค้ดคำสั่ง Split data Vote.arff แบบ 90-10

การแบ่งชุดข้อมูลการลงคะแนนเสียงเลือกตั้งแบบแบ่งเป็นชุดข้อมูลเรียนรู้ 80% และชุดข้อมูลทดสอบ 20% จะได้ชุดข้อมูลทดสอบจำนวน 87 instances และชุดข้อมูลเรียนรู้ 352-358 instances โดยสามารถเขียนคำสั่งได้ดังนี้



รูป 4.17 โค้ดคำสั่ง Split data Vote.arff แบบ 80-20

การแบ่งชุดข้อมูลการลงคะแนนเสียงเลือกตั้งแบบแบ่งเป็นชุดข้อมูลเรียนรู้ 70% และชุดข้อมูลทดสอบ 30% จะได้ชุดข้อมูลทดสอบจำนวน 130 instances และชุดข้อมูลเรียนรู้ 318-326 instances โดยสามารถเขียนคำสั่งได้ดังนี้

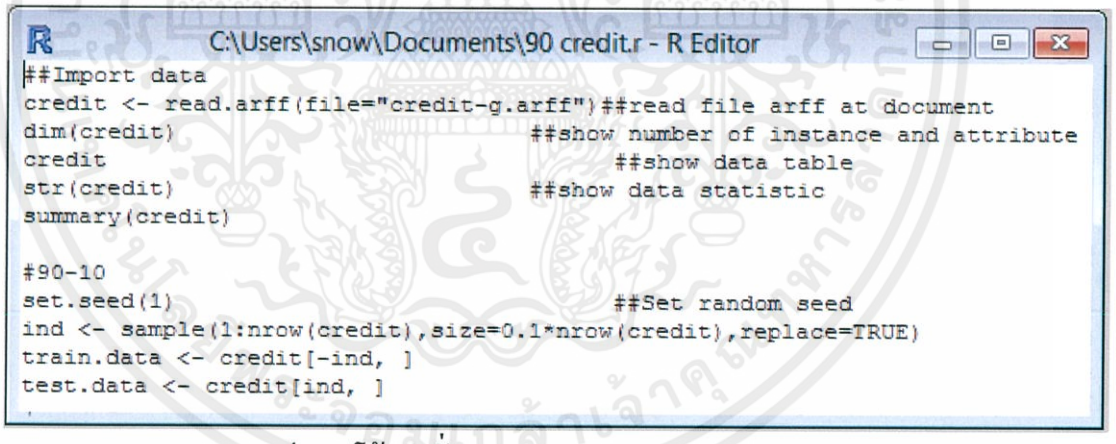


รูป 4.18 โค้ดคำสั่ง Split data Vote.arff แบบ 70-30

4.1.1.7 การแบ่งชุดข้อมูลสินเชื่อ (Credit-g.arff)

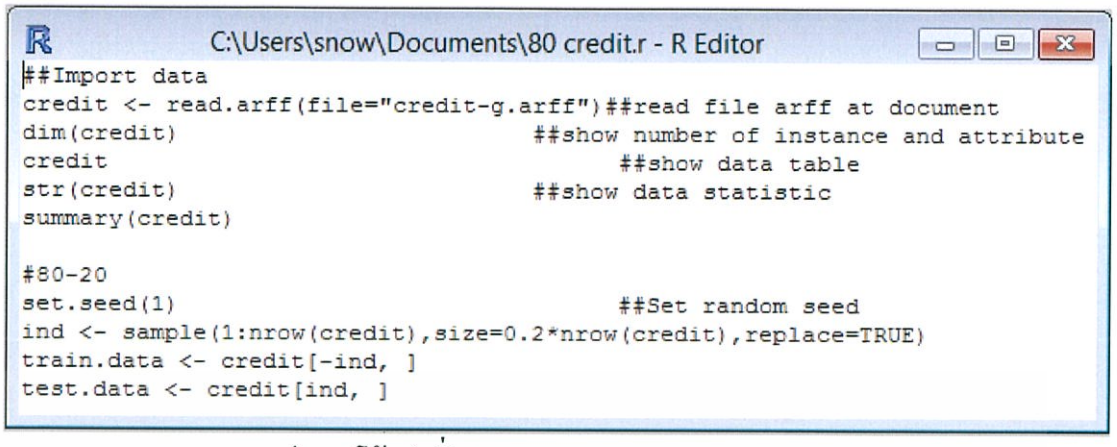
ชุดข้อมูลสินเชื่อประกอบด้วยข้อมูลทั้งหมด 1000 instances

การแบ่งชุดข้อมูลสินเชื่อแบบแบ่งเป็นชุดข้อมูลเรียนรู้ 90% และชุดข้อมูลทดสอบ 10% จะได้ชุดข้อมูลทดสอบจำนวน 100 instances และชุดข้อมูลเรียนรู้ 902-907 instances โดยสามารถเขียนคำสั่งได้ดังนี้



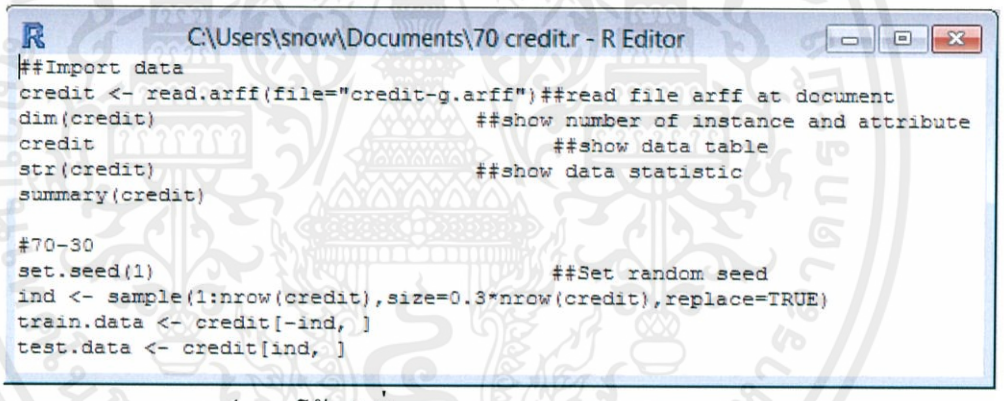
รูป 4.19 โค้ดคำสั่ง Split data Credit-g.arff แบบ 90-10

การแบ่งชุดข้อมูลสินเชื่อแบบแบ่งเป็นชุดข้อมูลเรียนรู้ 80% และชุดข้อมูลทดสอบ 20% จะได้ชุดข้อมูลทดสอบจำนวน 200 instances และชุดข้อมูลเรียนรู้ 815-825 instances โดยสามารถเขียนคำสั่งได้ดังนี้



รูป 4.20 โค้ดคำสั่ง Split data Credit-g.arff แบบ 80-20

การแบ่งชุดข้อมูลสินเชื่อแบบแบ่งเป็นชุดข้อมูลเรียนรู้ 70% และชุดข้อมูลทดสอบ 30% จะได้ชุดข้อมูลทดสอบจำนวน 300 instances และชุดข้อมูลเรียนรู้ 737-753 instances โดยสามารถเขียนคำสั่งได้ดังนี้

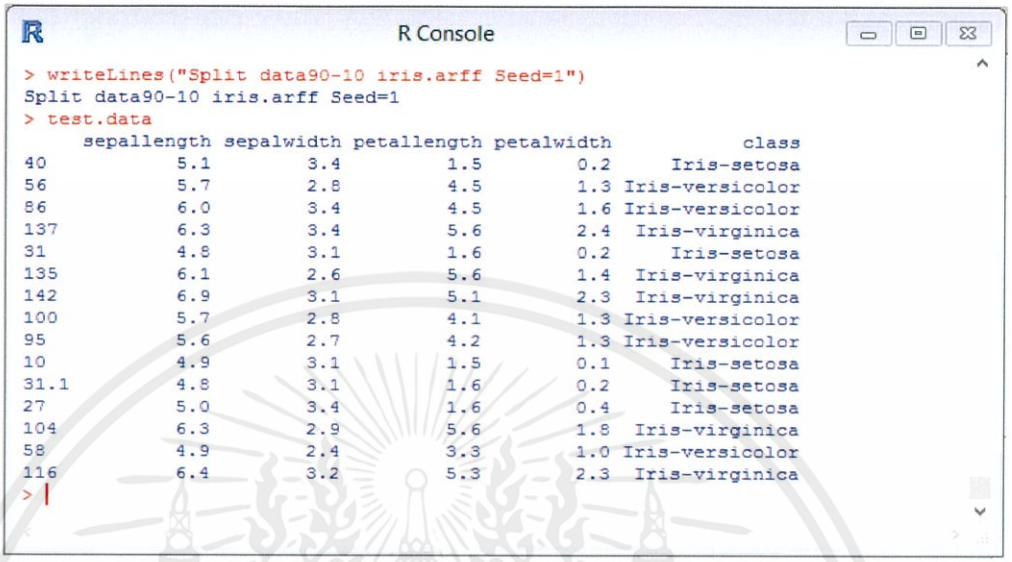


รูป 4.21 โค้ดคำสั่ง Split data Credit-g.arff แบบ 70-30

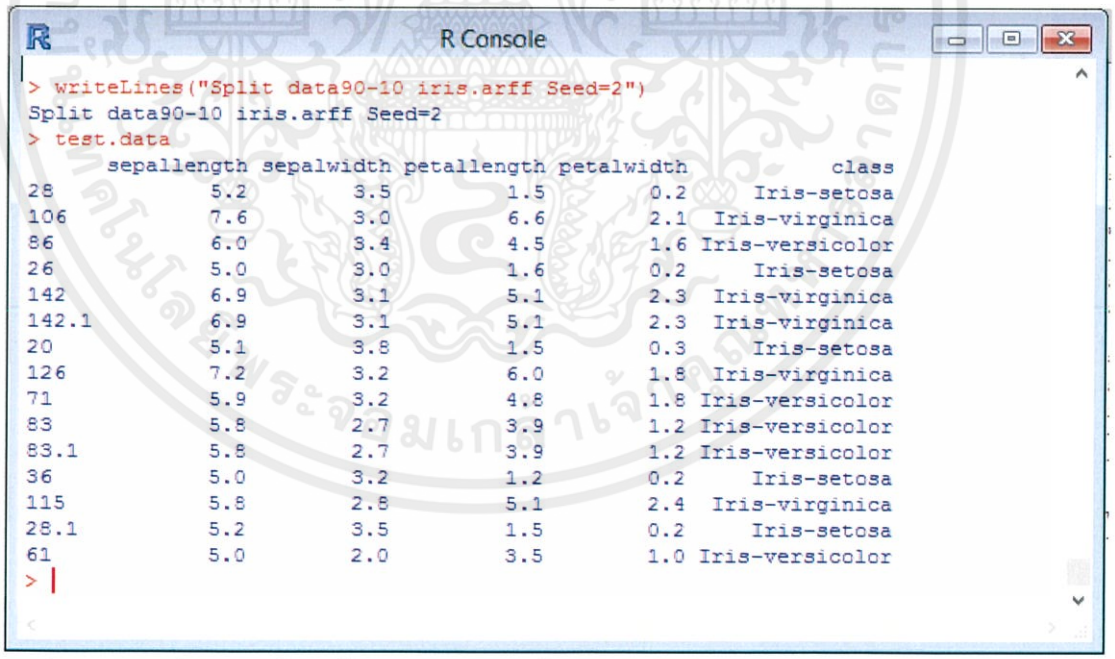
4.1.2 การกำหนดจำนวนครั้งในการสุ่ม (Random seed)

การแบ่งชุดข้อมูลโดยใช้ซอฟต์แวร์อาร์ ดาต้ามายนิ่ง เป็นเครื่องมือในการแบ่งชุดข้อมูลโดยใช้คำสั่งในการ Split data ซึ่งจะทำให้เกิดการแบ่งข้อมูลแบบสุ่ม ในการทดลองเพื่อวิเคราะห์ผลเราจึงจะทำการสุ่มด้วยกันทั้งหมด 10 ครั้ง เพื่อหาค่าเฉลี่ยของการแบ่งแต่ละแบบในแต่ละชุดข้อมูลและแต่ละอัลกอริทึม โดยการทดลองสุ่ม 10 ครั้งที่ไม่เหมือนกันนั้นเราสามารถตั้งค่าได้ในส่วนของ Random seed โดยการเปลี่ยนค่าเป็น 1, 2, 3, 4, … ไปเรื่อยๆจนครบ 10 ครั้ง จะทำให้เราได้ชุดข้อมูลที่แตกต่างกัน แต่มีจำนวนของชุดข้อมูลทดสอบที่เท่ากันในการกำหนดค่าการแบ่งที่เหมือนกัน แต่สำหรับในชุดข้อมูลเรียนรู้อาจมีจำนวนข้อมูลที่แตกต่างกันบ้างเพียงเล็กน้อยตามค่าการสุ่มของ

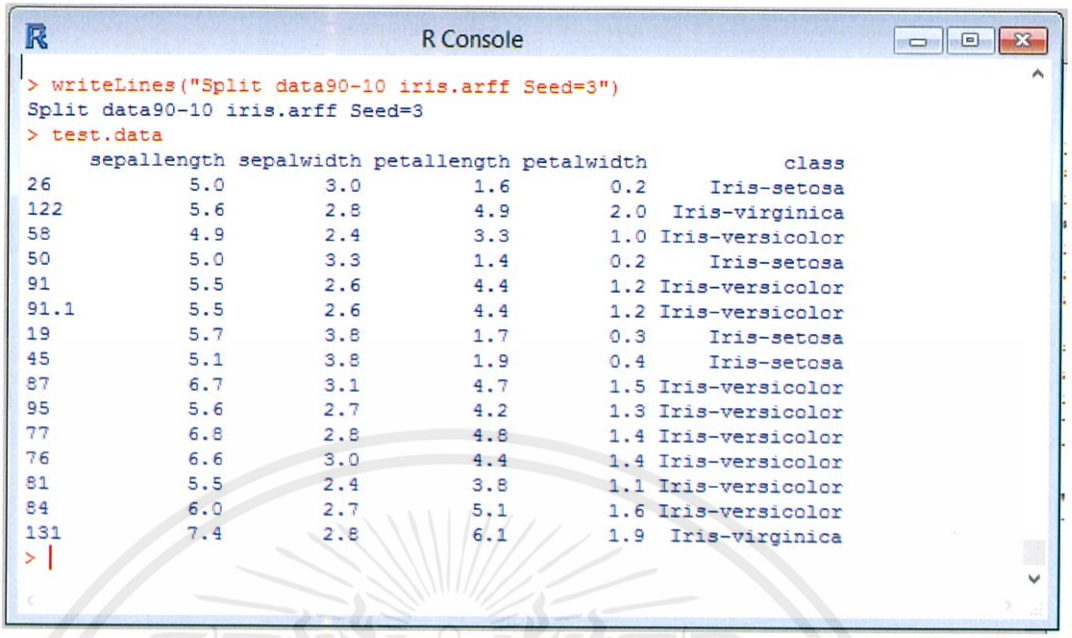
ซอฟต์แวร์ ตัวอย่างการแบ่งชุดข้อมูลดอกไอริส แบบแบ่งเป็นชุดข้อมูลเรียนรู้ 90% และเป็นชุดข้อมูลทดสอบ 10% โดยมีค่า Random seed ที่แตกต่างกัน จะเห็นได้ดังรูป



รูป 4.22 ตัวอย่างการแบ่งชุดข้อมูลทดสอบดอกไอริส แบบแบ่งเป็นชุดข้อมูลเรียนรู้ 90% และเป็นชุดข้อมูลทดสอบ 10% ค่า Random seed เท่ากับ 1



รูป 4.23 ตัวอย่างการแบ่งชุดข้อมูลทดสอบดอกไอริส แบบแบ่งเป็นชุดข้อมูลเรียนรู้ 90% และเป็นชุดข้อมูลทดสอบ 10% ค่า Random seed เท่ากับ 2

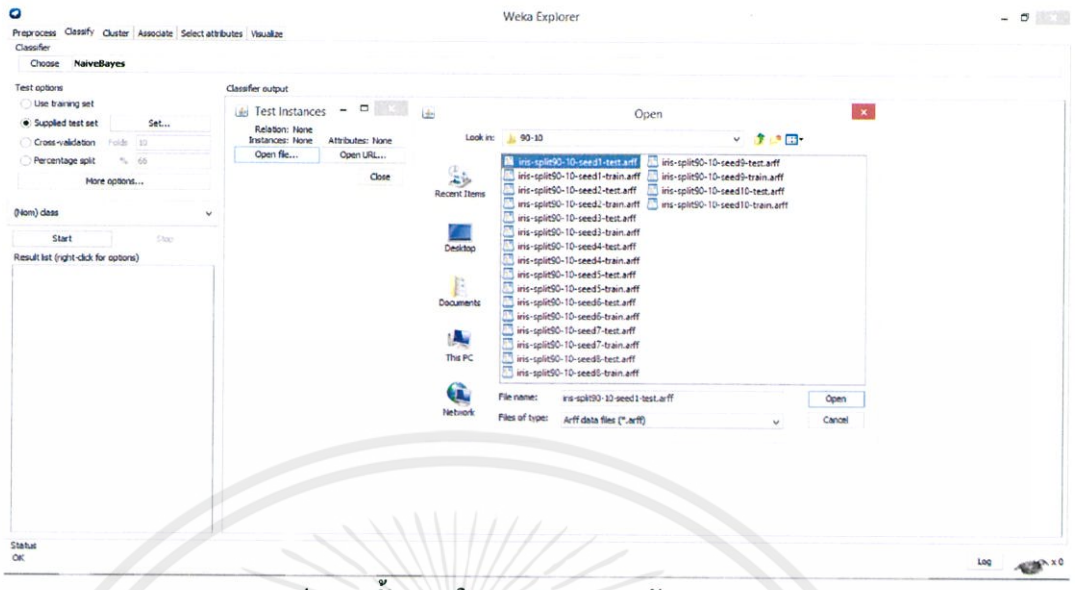


รูป 4.24 ตัวอย่างการแบ่งชุดข้อมูลทดสอบดอกไอริส แบบแบ่งเป็นชุดข้อมูลเรียนรู้ 90% และเป็นชุดข้อมูลทดสอบ10% ค่า Random seed เท่ากับ 3

4.2 การทดลองทำเหมืองข้อมูลด้วยซอฟต์แวร์เวก้า

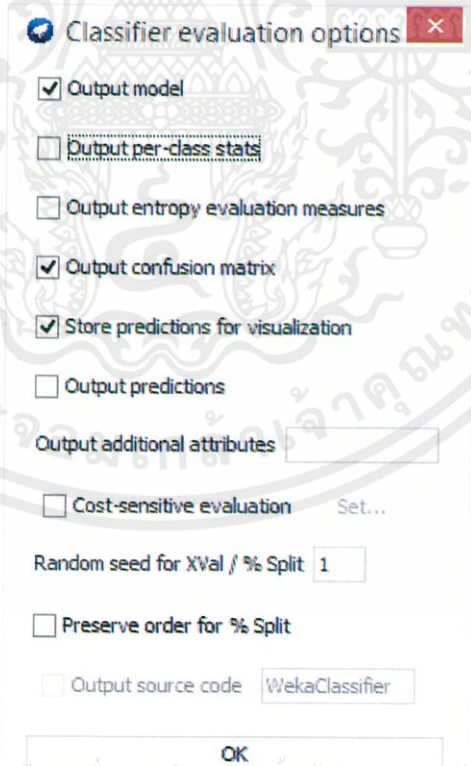
ในการทดลองศึกษาผลลัพธ์ในด้านการเปรียบเทียบประสิทธิภาพของซอฟต์แวร์ จะมีการทดลองทำเหมืองข้อมูลในซอฟต์แวร์เวก้า โดยการควบคุมชุดข้อมูล หลังจากการนำเข้าข้อมูลเรียบร้อยแล้วจะเข้าสู่กระบวนการสร้างโมเดลทำนายข้อมูล ซึ่งในหน้าต่างสำหรับการทำนายข้อมูลจะมีส่วนของการกำหนดชุดข้อมูลทดสอบ และการกำหนดค่าในการแสดงผล ตามขั้นตอนดังต่อนี้

ทำการตั้งค่า Test options โดยเลือกเป็นแบบ Supplied test set และคลิก Set…>Open file… และทำการเลือกชุดข้อมูลที่จะนำมาทำเป็นชุดข้อมูลทดสอบ



รูป 4.25 ขั้นตอนในการกำหนดชุดข้อมูลทดสอบ

การกำหนดค่าในการแสดงผล ผู้ใช้งานสามารถกำหนดรายละเอียดการแสดงผลของกระบวนการในการสร้างโมเดลการทำนายค่าได้ โดยคลิก More options ซึ่งจะแสดงหน้าต่างดังรูป



รูป 4.26 หน้าต่างการตั้งค่าการแสดงผลใน Classify ของ Weka

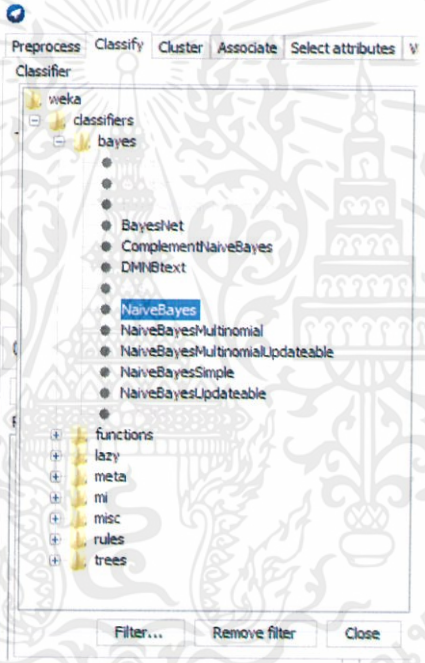
ซึ่งในที่นี้จะทำการตั้งค่าให้แสดงผลข้อมูล 3 อย่าง ได้แก่

- Output model
- Output confuse matrix
- Store predictions for visualization

หลังจากการตั้งค่ารูปแบบการทดสอบ และการแสดงผลแล้ว ขั้นตอนต่อไปจะเป็นการเลือกอัลกอริทึมในการทำนายค่า ซึ่งแต่ละอัลกอริทึมจะมีการตั้งค่าพารามิเตอร์แตกต่างกัน ดังต่อไปนี้

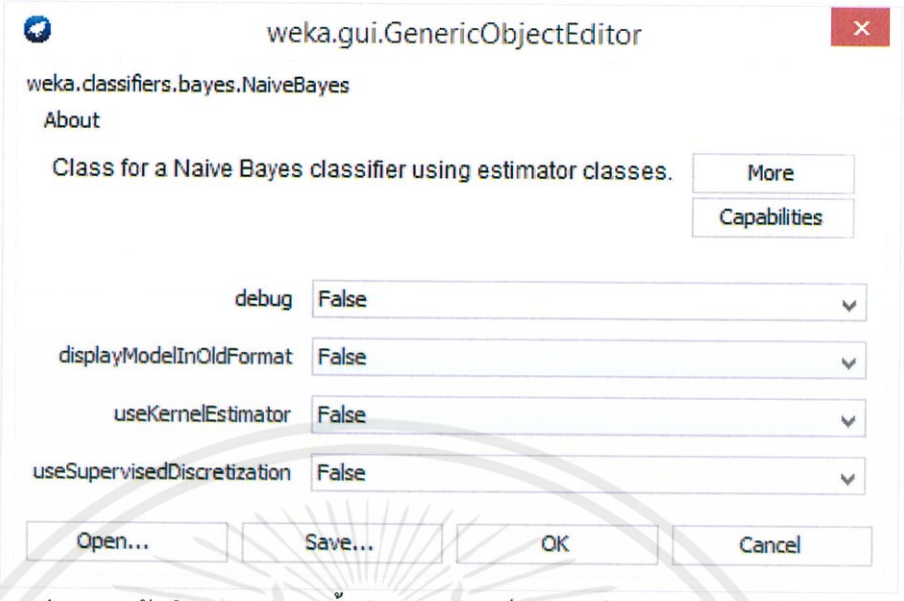
4.2.1 Naïve Bayes และการตั้งค่า

เข้าสู่หน้าต่าง Classify ทำการเลือกอัลกอริทึม NaïveBayes โดยคลิก Choose>classifiers>bayes>NaiveBayes



รูป 4.27 ขั้นตอนในการเลือกใช้อัลกอริทึม NaiveBayes ใน Weka

หลังจากนั้นทำการตั้งค่าพารามิเตอร์ต่างๆของอัลกอริทึม NaiveBayes โดยการคลิกที่ชื่ออัลกอริทึม NaiveBayes จะแสดงหน้าต่างดังรูป



รูป 4.28 หน้าต่างสำหรับการตั้งค่าพารามิเตอร์อัลกอริทึม NaiveBayes ใน Weka

ทำการตั้งค่าพารามิเตอร์ต่างๆ ของอัลกอริทึม NaiveBayes ดังนี้

- debug ตั้งค่าเป็น false กรณีที่ต้องการให้มีการแสดงผลรายละเอียดเพิ่มเติมหน้าคอนโซลในกรณีเกิดปัญหาต่าง ผู้ใช้สามารถทำการตั้งค่าพารามิเตอร์นี้เป็น True ได้
- displayModelInOldFormat ตั้งค่าเป็น False พารามิเตอร์นี้เป็นการตั้งค่าให้การทำนายค่ามีการแสดงผลข้อมูลในรูปแบบเดิม ซึ่งจะใช้ได้ดีกว่าในกรณีที่ข้อมูลมีค่าคลาสหลายค่า แต่จะใช้รูปแบบใหม่ได้ดีกว่าในกรณีที่ข้อมูลมีค่าคลาสน้อยและมีหลายแอตทริบิวต์ ซึ่งในการทดลองข้อมูลส่วนใหญ่มีค่าคลาสเพียง 2-3 class เท่านั้น
- useKernelEstimator ตั้งค่าเป็น False พารามิเตอร์นี้จะตั้งค่าเป็น True ในกรณีที่ต้องการใช้เคอร์เนลในการประมาณค่าสำหรับแอตทริบิวที่เป็นตัวเลขที่มีการกระจายค่ามากกว่าปกติ
- useSupervisedDiscretization ตั้งค่าเป็น False พารามิเตอร์นี้จะใช้ Supervised discretization ในการแปลงแอตทริบิวต์ที่เป็น numeric ไปเป็น nominal

หลังจากทำการตั้งค่าพารามิเตอร์ต่างๆเรียบร้อย ให้คลิก Start จะแสดงผลลัพธ์ดังรูป

```
=== Run information ===

Scheme:weka.classifiers.bayes.NaiveBayes
Relation:     iris
Instances:    136
Attributes:   5
              sepallength
              sepalwidth
              petallength
              petalwidth
              class
Test mode:user supplied test set: size unknown (reading incrementally)

=== Classifier model (full training set) ===

Naive Bayes Classifier

                         Class
Attribute          Iris-setosa Iris-versicolor  Iris-virginica
                        (0.34)         (0.33)          (0.33)
============================================================
sepallength
  mean                 4.9972         5.9765          6.6024
  std. dev.            0.3679         0.5009            0.66
  weight sum               46             45              45
  precision            0.1059         0.1059          0.1059

sepalwidth
  mean                 3.4174         2.7612          2.9552
  std. dev.            0.4025         0.3015          0.3135
  weight sum               46             45              45
  precision            0.1091         0.1091          0.1091
```

รูป 4.29 ตัวอย่างผลลัพธ์การทำนายค่าอัลกอริทึม NaiveBayes ใน Weka

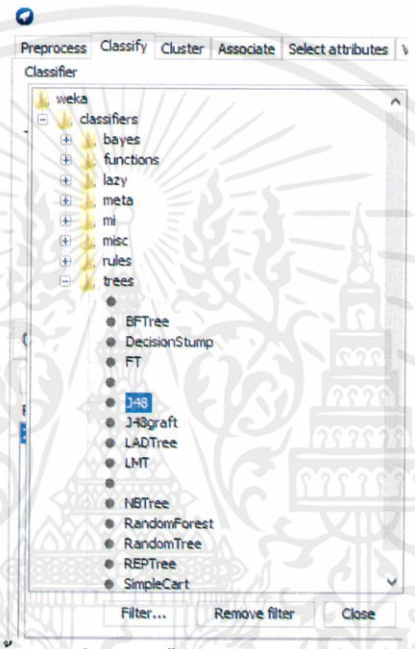
```
  petallength
    mean                1.4628         4.2611          5.5629
    std. dev.           0.1843         0.4692          0.5768
    weight sum              46             45              45
    precision           0.1405         0.1405          0.1405

  petalwidth
    mean                0.2758          1.313          2.0343
    std. dev.           0.1081         0.1922          0.2476
    weight sum              46             45              45
    precision           0.1143         0.1143          0.1143


  Time taken to build model: 0.01 seconds

  === Evaluation on test set ===
  === Summary ===

  Correctly Classified Instances          14               93.3333 %
  Incorrectly Classified Instances         1                6.6667 %
  Kappa statistic                          0.9
  Mean absolute error                      0.0357
  Root mean squared error                  0.1451
  Relative absolute error                  8.0256 %
  Root relative squared error             30.7852 %
  Total Number of Instances               15

=== Confusion Matrix ===

 a b c   <-- classified as
 5 0 0 | a = Iris-setosa
 0 5 0 | b = Iris-versicolor
 0 1 4 | c = Iris-virginica
```


รูป 4.30 ตัวอย่างผลลัพธ์การทำนายค่าอัลกอริทึม NaiveBayes ใน Weka (ต่อ)

ทำเช่นนี้กับทุกๆข้อมูลที่ต้องการทดลอง เก็บค่าผลลัพธ์เปอร์เซ็นต์ความถูกต้อง และนำไปวิเคราะห์

4.2.2 Decision Tree และการตั้งค่า

ในหน้าต่าง Classify ทำการเลือกอัลกอริทึม J48 โดยคลิกเลือก Choose>classifiers>trees>J48



รูป 4.31 ขั้นตอนในการเลือกใช้อัลกอริทึม J48 ใน Weka

หลังจากนั้นทำการตั้งค่าพารามิเตอร์ต่างๆของอัลกอริทึม J48 โดยการคลิกที่ชื่ออัลกอริทึม J48 จะแสดงหน้าต่างดังรูป

รูป 4.32 หน้าต่างสำหรับการตั้งค่าพารามิเตอร์อัลกอริทึม J48 ใน Weka

ทำการตั้งค่าพารามิเตอร์ต่างๆของอัลกอริทึม J48 ดังนี้

- binarySplit ตั้งค่าเป็น False พารามิเตอร์นี้เป็นการตั้งค่าว่าจะใช้ binary Split กับแอตทริบิวต์ nominal เมื่อมีการสร้าง tree หรือไม่ หากใช้ตั้งค่าเป็น True
- confidenceFactor กำหนดค่าเป็น 0.25 พารามิเตอร์นี้เป็นการกำหนดค่าปัจจัยความเชื่อมั่นซึ่งจะใช้เมื่อกำหนดให้มีการตัดแต่ง (Prune) หากกำหนดค่าไว้น้อยจะทำให้มีการตัดแต่งมาก
- debug ตั้งค่าเป็น false กรณีที่ต้องการให้มีการแสดงผลรายละเอียดเพิ่มเติมหน้าคอนโซลในกรณีเกิดปัญหาต่าง ผู้ใช้สามารถทำการตั้งค่าพารามิเตอร์นี้เป็น True ได้
- minNumObj กำหนดไว้เท่ากับ 2 พารามิเตอร์นี้เป็นการกำหนดจำนวน instance ขั้นต่ำในแต่ละ Leaf
- numFolds กำหนดไว้เท่ากับ 3 พารามิเตอร์นี้เป็นการกำหนดจำนวนข้อมูลซึ่งใช้สำหรับลดข้อผิดพลาดในการตัดแต่ง 1 fold จะใช้สำหรับการตัดแต่งและส่วนที่เหลือสำหรับการขยายของต้นไม้
- reducedErrorPruning ตั้งค่าเป็น False พารามิเตอร์นี้จะใช้ในการลดข้อผิดพลาดในการตัดแต่งซึ่งถูกใช้แทนการตัดแต่งใน C4.5

- saveInstanceData ตั้งค่าเป็น False พารามิเตอร์นี้ใช้ในการตั้งค่าให้มีการบันทึกชุดข้อมูลเรียนรู้สำหรับการแสดงผลลัพธ์
- seed ค่าเริ่มต้นมีค่าเป็น 1 และมีการเปลี่ยนค่าเป็น 2, 3, 4, ... ไปเรื่อยๆตามลำดับจนถึง 10 ตามการทดลอง พารามิเตอร์นี้ใช้ในการกำหนดค่าการสุ่มข้อมูลเมื่อการลดข้อผิดพลาดการตัดแต่งข้อมูลถูกใช้
- subtreeRaising กำหนดค่าเป็น True พารามิเตอร์นี้จะทำการพิจารณา subtree โดยหากมีการ Prune และ subtree นั้นสามารถเปลี่ยนระดับให้สูงขึ้นได้ จะทำการเปลี่ยนระดับ subtree นั้น
- unpruned กำหนดค่าเป็น False พารามิเตอร์นี้เป็นการกำหนดค่าว่าจะใช้การ prune หรือไม่ ถ้าให้ทำการ prune ตั้งค่าเป็น False ไม่ให้ Prune ตั้งค่าเป็น True
- useLaplace กำหนดค่าเป็น False พารามิเตอร์นี้เป็นการกำหนดให้ทำการนับจำนวนของใบที่ถูกทำให้ Smooth ด้วยวิธีการ Laplace

หลังจากทำการตั้งค่าพารามิเตอร์ต่างๆเรียบร้อย ให้คลิก Start จะแสดงผลลัพธ์ดังรูป

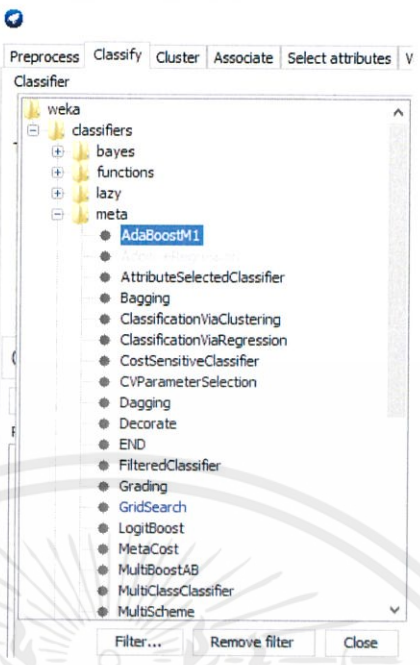
```
=== Run information ===

Scheme:weka.classifiers.trees.J48 -C 0.25 -M 2
Relation:     iris
Instances:    136
Attributes:   5
              sepallength
              sepalwidth
              petallength
              petalwidth
              class
Test mode:user supplied test set: size unknown (reading incrementally)

=== Classifier model (full training set) ===

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

petalwidth <= 0.6: Iris-setosa (46.0)
petalwidth > 0.6
|   petalwidth <= 1.7
|   |   petallength <= 4.9: Iris-versicolor (43.0/1.0)
|   |   petallength > 4.9
|   |   |   petalwidth <= 1.5: Iris-virginica (2.0)
|   |   |   petalwidth > 1.5: Iris-versicolor (3.0/1.0)
|   petalwidth > 1.7: Iris-virginica (42.0/1.0)

Number of Leaves  :     5

Size of the tree :      9

Time taken to build model: 0.02 seconds

=== Evaluation on test set ===
=== Summary ===

Correctly Classified Instances          15               100      %
Incorrectly Classified Instances         0                 0      %
Kappa statistic                          1
Mean absolute error                      0.0094
Root mean squared error                  0.0149
Relative absolute error                  2.1152 %
Root relative squared error              3.1532 %
Total Number of Instances               15

=== Confusion Matrix ===

 a b c   <-- classified as
 5 0 0 | a = Iris-setosa
 0 5 0 | b = Iris-versicolor
 0 0 5 | c = Iris-virginica
```

รูป 4.33 ตัวอย่างผลลัพธ์การทำนายค่าอัลกอริทึม J48 ใน Weka

เนื่องจากอัลกอริทึม J48 เป็นอัลกอริทึมในหมวด Trees จะสังเกตเห็นว่าจะมีการแสดงโมเดลในการทำนายค่าของข้อมูลให้ผู้ใช้ทราบ ทำเช่นนี้กับทุกๆ ข้อมูลที่ต้องการทดลอง เก็บค่าผลลัพธ์เปอร์เซ็นต์ความถูกต้อง และนำไปวิเคราะห์

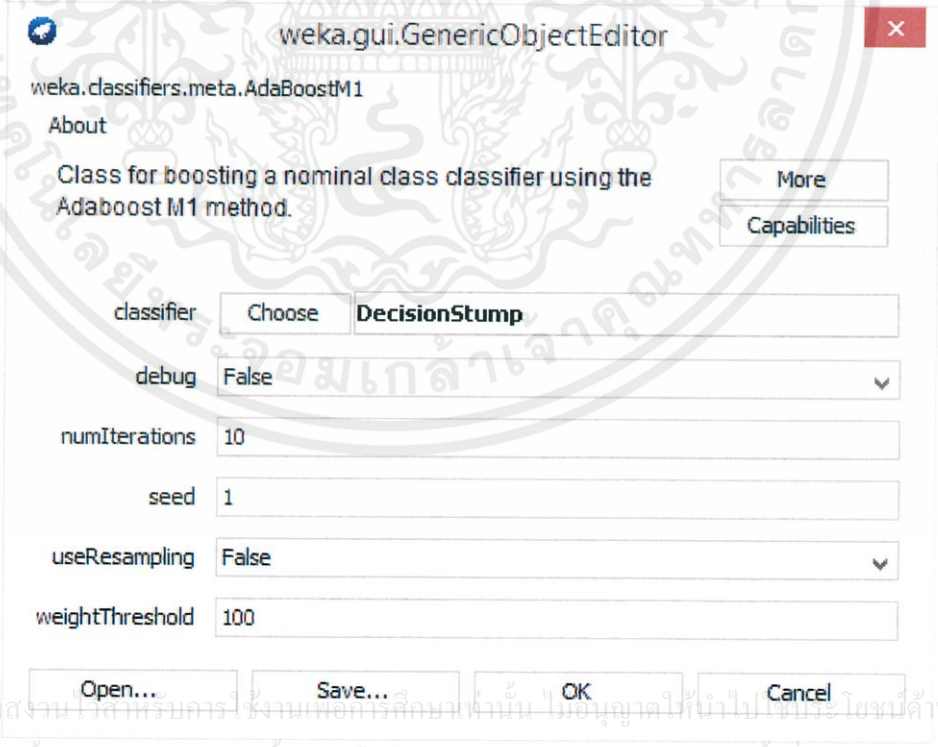
4.2.3 AdaBoost และการตั้งค่า

ในหน้าต่าง Classify ผู้ใช้สามารถทำการเลือกอัลกอริทึม AdaBoost ได้โดยคลิก Choose>classifiers>meta>AdaBoostM1



รูป 4.34 ขั้นตอนในการเลือกใช้อัลกอริทึม AdaBoostM1 ใน Weka

หลังจากนั้นทำการตั้งค่าพารามิเตอร์ต่างๆ ของอัลกอริทึม AdaBoostM1 โดยการคลิกที่ชื่ออัลกอริทึม AdaBoostM1 จะแสดงหน้าต่างดังรูป



รูป 4.35 หน้าต่างสำหรับการตั้งค่าพารามิเตอร์อัลกอริทึม AdaBoost ใน Weka

ทำการตั้งค่าพารามิเตอร์ต่างๆของอัลกอริทึม AdaBoostM1 ดังนี้

- classifier ทำการเลือกอัลกอริทึมพื้นฐานที่เราจะใช้ ซึ่งในการทดลองนี้จะเลือกใช้อัลกอริทึม J48 และทำการตั้งค่าพารามิเตอร์ภายใน J48 เหมือนกันกับในข้อ 4.2.2
- debug ตั้งค่าเป็น false กรณีที่ต้องการให้มีการแสดงผลรายละเอียดเพิ่มเติมหน้าคอนโซลในกรณีเกิดปัญหาต่าง ผู้ใช้สามารถทำการตั้งค่าพารามิเตอร์นี้เป็น True ได้
- numIterations ทำการกำหนดค่าให้เป็น 10 พารามิเตอร์นี้เป็นการกำหนดค่าจำนวนครั้งที่ต้องการให้ดำเนินการซ้ำกี่รอบ
- seed ทำการกำหนดค่าให้เป็น 1 พารามิเตอร์นี้เป็นการกำหนดค่าจำนวนครั้งการสุ่ม
- useResampling กำหนดค่าเป็น False พารามิเตอร์นี้เป็นการกำหนดให้ใช้ Resampling แทนการใช้ Reweighting
- weightThreshold กำหนดค่าเท่ากับ 100 พารามิเตอร์นี้ใช้สำหรับเป็นเกณฑ์การให้น้ำหนักสำหรับการทำ weight prune

หลังจากทำการตั้งค่าพารามิเตอร์ต่างๆ เรียบร้อย ให้คลิก Start จะแสดงผลลัพธ์ดังรูป

```
=== Run information ===

Scheme:weka.classifiers.meta.AdaBoostM1 -P 100 -S 1 -I 10 -W weka.classifiers.trees.J48 -- -C 0.25 -M 2
Relation:     iris
Instances:    136
Attributes:   5
              sepallength
              sepalwidth
              petallength
              petalwidth
              class
Test mode:user supplied test set: size unknown (reading incrementally)

=== Classifier model (full training set) ===

AdaBoostM1: Base classifiers and their weights:

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

petalwidth <= 0.6: Iris-setosa (46.0)
petalwidth > 0.6
|   petalwidth <= 1.7
|   |   petallength <= 4.9: Iris-versicolor (43.0/1.0)
|   |   petallength > 4.9
|   |   |   petalwidth <= 1.5: Iris-virginica (2.0)
|   |   |   petalwidth > 1.5: Iris-versicolor (3.0/1.0)
|   petalwidth > 1.7: Iris-virginica (42.0/1.0)

Number of Leaves  :     5

Size of the tree :      9
```

รูป 4.36 ตัวอย่างผลลัพธ์การทำนายค่าอัลกอริทึม AdaBoostM1 ใน Weka

```
Weight: 3.79

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

petalwidth <= 0.6: Iris-setosa (23.52)
petalwidth > 0.6
|   petallength <= 5
|   |   sepallength <= 4.9: Iris-virginica (22.67)
|   |   sepallength > 4.9
|   |   |   petallength <= 4.8: Iris-versicolor (44.14/1.02)
|   |   |   petallength > 4.8
|   |   |   |   petalwidth <= 1.7: Iris-versicolor (2.05/0.51)
|   |   |   |   petalwidth > 1.7: Iris-virginica (2.56)
|   petallength > 5: Iris-virginica (41.07/0.51)

Number of Leaves  :     6

Size of the tree :      11


Weight: 4.18

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

petalwidth <= 0.6: Iris-setosa (11.94)
petalwidth > 0.6
|   petalwidth <= 1.4: Iris-versicolor (8.05)
|   petalwidth > 1.4
|   |   sepalwidth <= 3.1
|   |   |   petalwidth <= 1.6
|   |   |   |   sepalwidth <= 2.3: Iris-virginica (17.26/0.26)
|   |   |   |   sepalwidth > 2.3
|   |   |   |   |   petallength <= 5.4: Iris-versicolor (19.34/0.26)
|   |   |   |   |   petallength > 5.4: Iris-virginica (11.51)
|   |   |   petalwidth > 1.6: Iris-virginica (53.03/0.26)
|   |   sepalwidth > 3.1
|   |   |   petallength <= 4.9: Iris-versicolor (12.03)
|   |   |   petallength > 4.9: Iris-virginica (2.85)

Number of Leaves  :     8

Size of the tree :      15

Weight: 5.16

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

petalwidth <= 0.6: Iris-setosa (6.0)
petalwidth > 0.6
|   petallength <= 5
|   |   sepallength <= 6
|   |   |   sepalwidth <= 3.1
|   |   |   |   petalwidth <= 1.4: Iris-versicolor (2.48)
|   |   |   |   petalwidth > 1.4: Iris-virginica (23.67/0.52)
|   |   |   sepalwidth > 3.1: Iris-versicolor (5.79)
|   |   sepallength > 6
|   |   |   petalwidth <= 1.7: Iris-versicolor (47.68)
|   |   |   petalwidth > 1.7: Iris-virginica (8.94)
|   petallength > 5
|   |   sepalwidth <= 2.7: Iris-versicolor (9.2/0.65)
|   |   sepalwidth > 2.7: Iris-virginica (32.24)

Number of Leaves  :     8

Size of the tree :      15
```

รูป 4.37 ตัวอย่างผลลัพธ์การทำนายค่าอัลกอริทึม AdaBoostM1 ใน Weka (ต่อ)

```
Weight: 4.74

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

petallength <= 5
|   petalwidth <= 0.6: Iris-setosa (3.03)
|   petalwidth > 0.6
|   |   petalwidth <= 1.7
|   |   |   sepallength <= 4.9: Iris-virginica (2.92)
|   |   |   sepallength > 4.9: Iris-versicolor (59.83/4.31)
|   |   petalwidth > 1.7
|   |   |   sepalwidth <= 3.1: Iris-virginica (8.95)
|   |   |   sepalwidth > 3.1: Iris-versicolor (2.92)
petallength > 5: Iris-virginica (58.35/4.31)

Number of Leaves  :     6

Size of the tree :      11

 Weight: 2.69

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

 sepalwidth <= 2.6
 |   petallength <= 4.9
 |   |   sepallength <= 6.1: Iris-virginica (2.01/0.46)
 |   |   sepallength > 6.1: Iris-versicolor (6.17)
 |   petallength > 4.9: Iris-virginica (42.14)
 sepalwidth > 2.6
 |   petalwidth <= 1.7
 |   |   sepallength <= 6.2
 |   |   |   petalwidth <= 1.4: Iris-setosa (2.0/0.42)
 |   |   |   petalwidth > 1.4: Iris-versicolor (50.13)
 |   |   sepallength > 6.2
 |   |   |   petallength <= 5: Iris-versicolor (6.49)
 |   |   |   petallength > 5: Iris-virginica (7.66)
 |   petalwidth > 1.7
 |   |   sepalwidth <= 3: Iris-virginica (17.34)
 |   |   sepalwidth > 3: Iris-versicolor (2.05/0.49)

 Number of Leaves  :     9

 Size of the tree :      17

Weight: 4.59

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

petallength <= 4.7
|   petalwidth <= 0.6: Iris-setosa (2.54)
|   petalwidth > 0.6: Iris-versicolor (54.04/0.79)
petallength > 4.7
|   petalwidth <= 1.7
|   |   petalwidth <= 1.5: Iris-virginica (20.31/0.05)
|   |   petalwidth > 1.5: Iris-versicolor (21.04/0.79)
|   petalwidth > 1.7: Iris-virginica (38.06/0.79)

Number of Leaves  :     5

Size of the tree :      9
```

รูป 4.38 ตัวอย่างผลลัพธ์การทำนายค่าอัลกอริทึม AdaBoostM1 ใน Weka (ต่อ)

```
Weight: 4.01

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

petallength <= 5.1
|   sepallength <= 4.9: Iris-virginica (23.21/1.04)
|   sepallength > 4.9
|   |   petallength <= 4.9: Iris-versicolor (52.24/1.46)
|   |   petallength > 4.9
|   |   |   petalwidth <= 1.5: Iris-virginica (10.31)
|   |   |   petalwidth > 1.5
|   |   |   |   petalwidth <= 1.7: Iris-versicolor (10.31)
|   |   |   |   petalwidth > 1.7: Iris-virginica (3.0)
petallength > 5.1: Iris-virginica (36.93)

Number of Leaves  :     6

Size of the tree :      11

Weight: 3.98

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

petalwidth <= 0.6: Iris-setosa (35.13)
petalwidth > 0.6
|   sepalwidth <= 3.1
|   |   petalwidth <= 1.6
|   |   |   sepallength <= 7
|   |   |   |   petalwidth <= 1.5
|   |   |   |   |   petallength <= 4.9: Iris-versicolor (14.56)
|   |   |   |   |   petallength > 4.9: Iris-virginica (5.25)
|   |   |   |   petalwidth > 1.5: Iris-versicolor (4.45)
|   |   |   sepallength > 7: Iris-virginica (11.29)
|   |   petalwidth > 1.6: Iris-virginica (49.03/0.8)
|   sepalwidth > 3.1
|   |   petallength <= 4.9: Iris-versicolor (11.31)
|   |   petallength > 4.9: Iris-virginica (4.97)

Number of Leaves  :     8

Size of the tree :      15

Weight: 5.13

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

petalwidth <= 0.6: Iris-setosa (17.67)
petalwidth > 0.6
|   petalwidth <= 1.7
|   |   petallength <= 5.4
|   |   |   sepallength <= 4.9: Iris-virginica (5.68)
|   |   |   sepallength > 4.9
|   |   |   |   sepalwidth <= 2.2: Iris-virginica (3.1/0.86)
|   |   |   |   sepalwidth > 2.2: Iris-versicolor (77.11/0.4)
|   |   petallength > 5.4: Iris-virginica (5.68)
|   petalwidth > 1.7
|   |   sepallength <= 5.9: Iris-versicolor (6.34/0.66)
|   |   sepallength > 5.9: Iris-virginica (20.42)

Number of Leaves  :     7

Size of the tree :      13
```

รูป 4.39 ตัวอย่างผลลัพธ์การทำนายค่าอัลกอริทึม AdaBoostM1 ใน Weka (ต่อ)

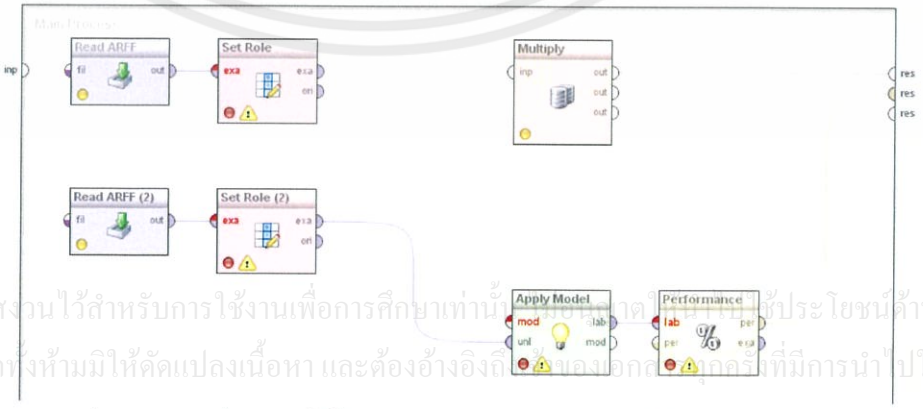
```
Weight: 4.25

Number of performed Iterations: 10


Time taken to build model: 0.11 seconds

=== Evaluation on test set ===
=== Summary ===

Correctly Classified Instances          15               100      %
Incorrectly Classified Instances         0                 0      %
Kappa statistic                          1
Mean absolute error                      0.0018
Root mean squared error                  0.0086
Relative absolute error                  0.41   %
Root relative squared error              1.8334 %
Total Number of Instances               15

=== Confusion Matrix ===

 a b c   <-- classified as
 5 0 0 | a = Iris-setosa
 0 5 0 | b = Iris-versicolor
 0 0 5 | c = Iris-virginica
```

รูป 4.40 ตัวอย่างผลลัพธ์การทำนายค่าอัลกอริทึม AdaBoostM1 ใน Weka (ต่อ)

เนื่องจากเรามีการกำหนดค่าในการทำซ้ำไว้เท่ากับ 10 จึงจะเห็นได้ว่ามีการคำนวณโมเดลออกมาทั้งหมด 10 แบบ และมีการให้ค่าน้ำหนักของแต่ละโมเดลที่แตกต่างกัน การทำงานของอัลกอริทึม AdaBoostM1 จะช่วยพัฒนาประสิทธิภาพของอัลกอริทึมพื้นฐานที่เราเลือกใช้ให้ดียิ่งขึ้น ทำเช่นนี้กับทุกๆ ข้อมูลที่ต้องการทดลอง เก็บค่าผลลัพธ์เปอร์เซ็นต์ความถูกต้อง และนำไปวิเคราะห์

4.3 การทดลองทำเหมืองข้อมูลด้วยซอฟต์แวร์ราปิดมายเนอร์

ในการทดลองศึกษาผลลัพธ์ในด้านการเปรียบเทียบประสิทธิภาพของซอฟต์แวร์ จะมีการทดลองทำเหมืองข้อมูลในซอฟต์แวร์ราปิดมายเนอร์ โดยการควบคุมชุดข้อมูลเดียวกันกับชุดข้อมูลที่ทดลองในซอฟต์แวร์อื่นๆ ซึ่งในการทดลองนี้จะทำการทดสอบแบบ Supplied test set ซึ่งจะมีการต่อโมดูล ดังรูป

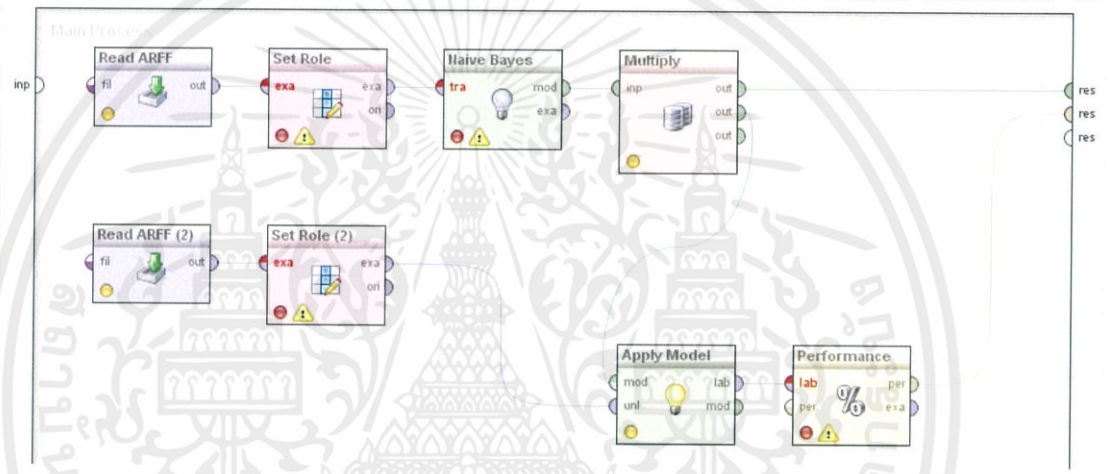


รูป 4.41 ตัวอย่างการต่อโมดูลแบบ Supplied test set ใน RapidMiner

หลังจากการเลือกใช้โมดูลสำหรับนำเข้าข้อมูลและทำการ Set label เรียบร้อยแล้วจะเข้าสู่ขั้นตอนในการเลือกโมดูลอัลกอริทึมสำหรับการทำเหมืองข้อมูลแบบการทำนายข้อมูลมาใช้งาน ในกรณีที่ทำการทดสอบแบบ Supplied test set นี้ และมีการตั้งค่าพารามิเตอร์ต่างๆ ของแต่ละโมดูลดังต่อไปนี้

4.3.1 Naïve Bayes และการตั้งค่า

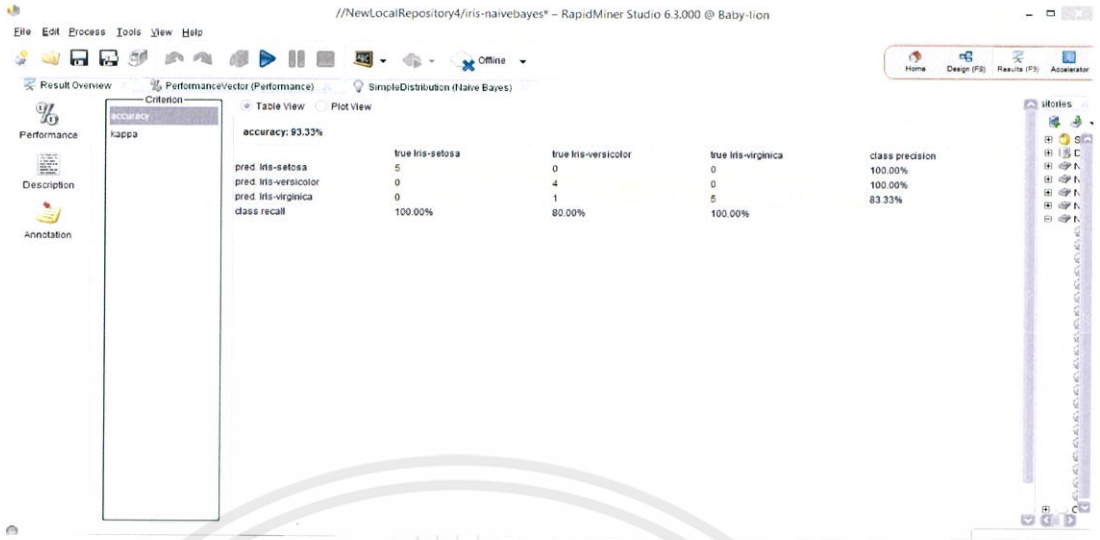
ทำการเลือกโมดูลอัลกอริทึม Naïve Bayes ซึ่งอยู่ในหมวดหมู่ Modeling>Classification and Regression>Bayesian Modeling ลากวางไว้ในตำแหน่งระหว่างโมดูล Set role ตัวที่ 1 และโมดูล Multiply ดังรูป



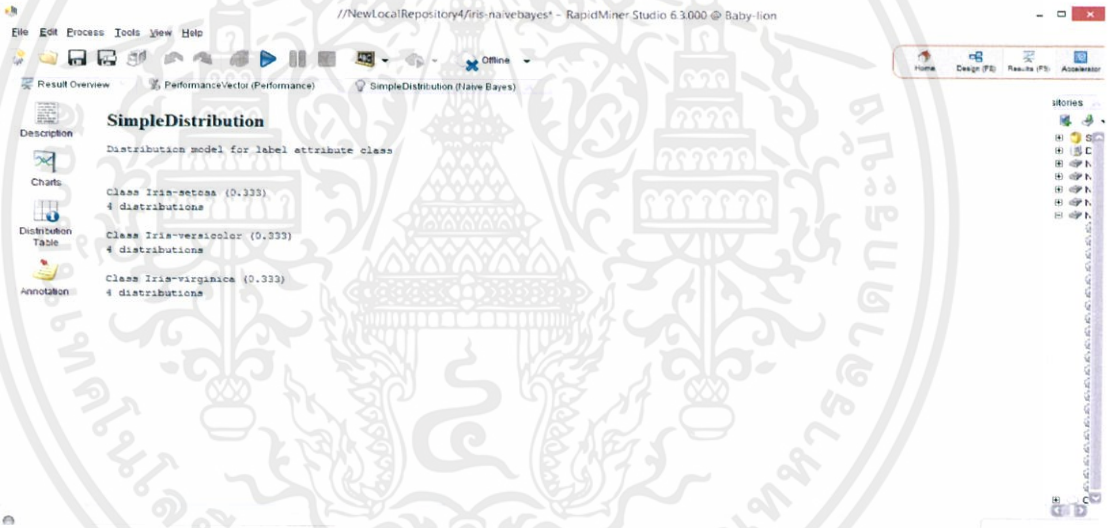
รูป 4.42 ตัวอย่างการต่อโมดูลอัลกอริทึม Naïve Bayes ใน RapidMiner

หลังจากนั้นให้ทำการตั้งค่าพารามิเตอร์ของอัลกอริทึม Naïve Bayes ซึ่งสำหรับในซอฟต์แวร์ RapidMiner จะเห็นได้ว่ามีเพียงแค่พารามิเตอร์เดียวเมื่อเทียบกับซอฟต์แวร์ Weka นั่นคือพารามิเตอร์ Laplace correction โดยในการทดลองจะทำการคลิกเครื่องหมายถูกไว้หน้าพารามิเตอร์นี้ โดยพารามิเตอร์นี้เป็นการกำหนดว่า Laplace correction จะถูกใช้เพื่อป้องกันเมื่อมีการคำนวณความน่าจะเป็น โดยมีผลลัพธ์เป็นศูนย์

หลังจากทำการต่อโมดูลเสร็จเรียบร้อยแล้ว ให้ทำการคลิกรันเพื่อดูผลลัพธ์ จะได้ผลลัพธ์ในกรณีที่ต่อตามรูปด้านบน ดังรูป



รูป 4.43 หน้าต่างผลลัพธ์ Performance Vector การทำนายค่าอัลกอริทึม Naïve Bayes ใน RapidMiner

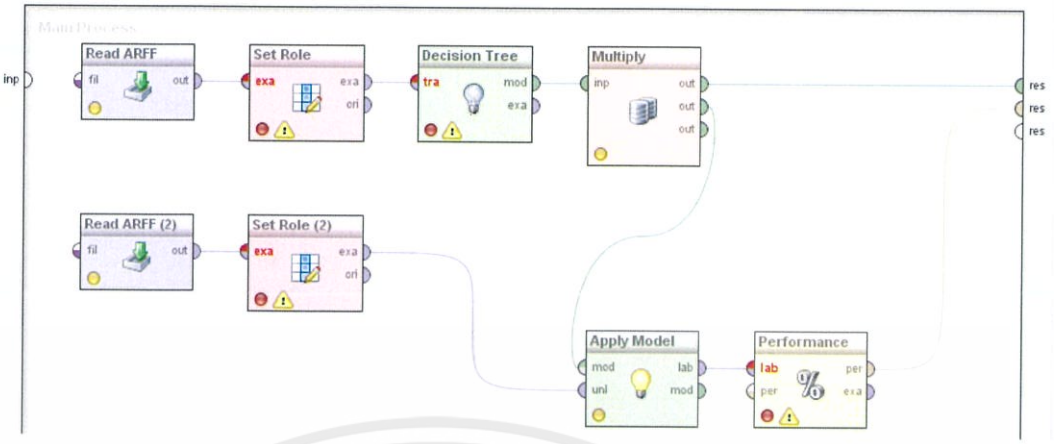


รูป 4.44 หน้าต่างผลลัพธ์ SimpleDistribution การทำนายค่าอัลกอริทึม Naïve Bayes ใน RapidMiner

ทำเช่นนี้กับทุกๆ ข้อมูลที่ต้องการทดลอง เก็บค่าผลลัพธ์เปอร์เซ็นต์ความถูกต้อง และนำไปวิเคราะห์

4.3.2 Decision Tree และการตั้งค่า

ทำการเลือกโมดูลอัลกอริทึม Decision Tree ซึ่งอยู่ในหมวดหมู่ Modeling>Classification and Regression>Tree Induction ลากวางไว้ในตำแหน่งระหว่างโมดูล Set role ตัวที่ 1 และโมดูล Multiply ดังรูป



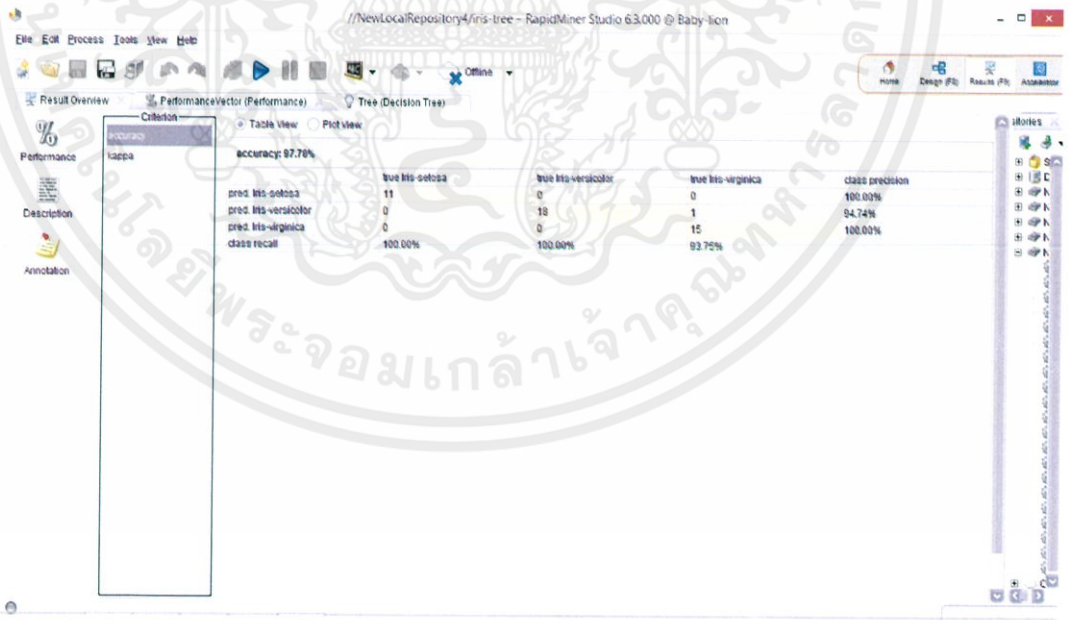
รูป 4.45 ตัวอย่างการต่อโมดูลอัลกอริทึม Decision Tree ใน RapidMiner

หลังจากนั้นทำการตั้งค่าพารามิเตอร์ของอัลกอริทึม Decision Tree ต่างๆ ดังนี้

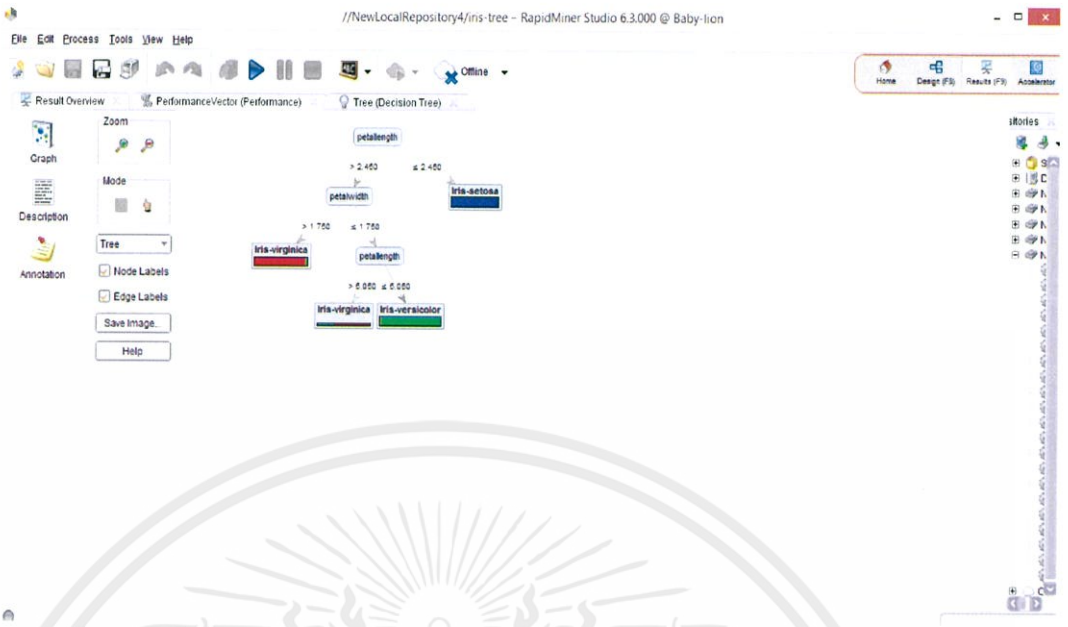
- criterion กำหนดให้เป็น information_gain ซึ่งพารามิเตอร์นี้เป็นการกำหนดเกณฑ์ที่ใช้ในการเลือกแอตทริบิวต์ที่จะถูก Split ซึ่งพารามิเตอร์นี้จะมีค่าให้เลือก 4 ค่า ดังต่อไปนี้
- information_gain Entropy ของทุกๆ แอตทริบิวต์จะถูกนำมาคำนวณ และทำการเลือกแอตทริบิวต์ที่มีค่า Entropy น้อยที่สุดมาทำการ Split
- gain_ratio จะทำการปรับค่า Information gain ของทุกๆ แอตทริบิวให้ค่าของแอตทริบิวต์มีความกว้างสม่ำเสมอกัน
- gini_index พารามิเตอร์นี้จะทำการชี้วัดการปนเปื้อนของชุดข้อมูลตัวอย่าง จะทำการ Split แอตทริบิวต์ที่ถูกเลือกโดยการลดค่าเฉลี่ยของผลลัพธ์ย่อย
- accuracy แอตทริบิวต์ที่มีความถูกต้องมากที่สุดของทั้ง Tree จะถูกเลือกสำหรับการทำการ Split
- maximal depth กำหนดให้มีค่าเท่ากับ 20 เป็นการกำหนดค่าความลึกสูงสุดของ tree ซึ่งจะมีความแตกต่างกันไปตามขนาดและลักษณะของชุดข้อมูลตัวอย่างที่ใช้ พารามิเตอร์นี้จะใช้ในการจำกัดขนาดของ decision tree ซึ่ง tree ที่สร้างขึ้นจากกระบวนการอาจจะไม่มีความต่อเนื่องหากความลึกของ tree เท่ากับค่าความลึกสูงสุด
- apply pruning ให้ทำการคลิกเครื่องหมายถูกหน้าพารามิเตอร์นี้ เป็นการกำหนดให้ tree ที่สร้างขึ้นนั้นเป็น tree ที่มีการตัดแต่ง (Prune)
- confidence กำหนดให้มีค่าเท่ากับ 0.25 เป็นการกำหนดค่าระดับความเชื่อมั่น ซึ่งจะถูกใช้สำหรับการคำนวณข้อผิดพลาดที่เกินขึ้นในการ prune

- apply prepruning ทำการคลิกเครื่องหมายถูกหน้าพารามิเตอร์นี้ ซึ่งพารามิเตอร์นี้ค่าเริ่มต้นจะกำหนดให้ decision tree ถูกสร้างขึ้นด้วยการ prepruning การตั้งค่าพารามิเตอร์นี้เป็น false จะเป็นการปิดการทำ prepruning และจะทำให้มีการสร้าง decision tree แบบไม่มีการ prepruning
- minimal gain ทำการกำหนดค่าเป็น 0.1 พารามิเตอร์นี้ gain ของแต่ละ node จะถูกคำนวณก่อนทำการ Split ซึ่ง node นั้นจะถูก Split เมื่อค่า gain ที่ได้มีค่ามากกว่าค่า minimal gain ที่กำหนด หากทำการตั้งค่า minimal gain ไว้สูง การ Split จะเกิดขึ้นน้อยทำให้ต้นไม้มีขนาดเล็ก
- minimal leaf size กำหนดค่าให้เท่ากับ 2 พารามิเตอร์นี้เป็นการกำหนดค่าให้มี instance ขั้นต่ำในแต่ละ leaf
- minimal size for Split กำหนดค่าให้เท่ากับ 4 พารามิเตอร์นี้เป็นการกำหนดค่าให้มีการ Split node ให้มีขนาดมากกว่าหรือเท่ากับค่า minimal size of Split ที่กำหนดไว้
- number of prepruning alternatives ทำการกำหนดค่าเท่ากับ 3 พารามิเตอร์นี้เป็นการกำหนดจำนวนnode ที่แยกออกไป

หลังจากทำการตั้งค่าพารามิเตอร์ต่างๆของอัลกอริทึม Decision tree เสร็จเรียบร้อยแล้ว ให้คลิก Run เพื่อดูผลลัพธ์ที่เกิดขึ้น จะได้ผลลัพธ์ดังรูป



รูป 4.46 หน้าต่างผลลัพธ์ Performance Vector การทำนายค่าอัลกอริทึม Decision Tree ใน RapidMiner

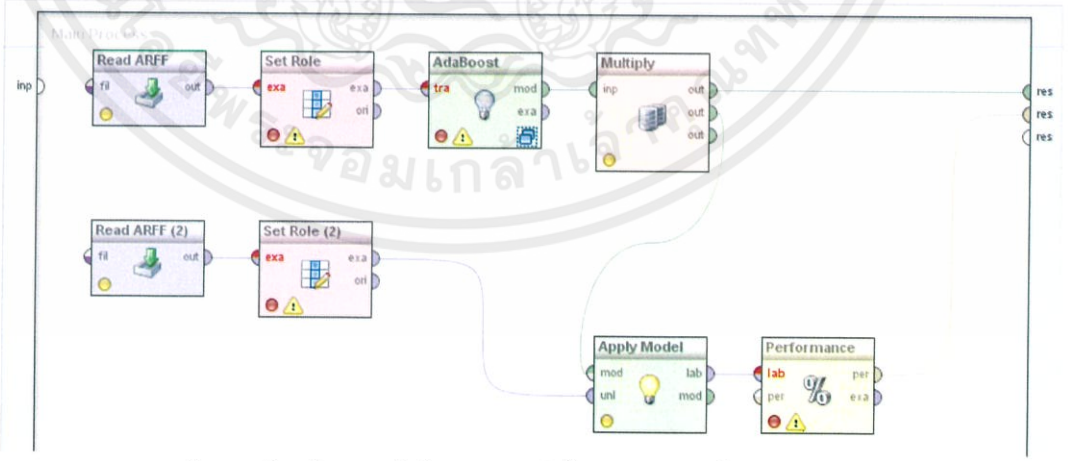


รูป 4.47 หน้าต่างผลลัพธ์ Tree การทำนายค่าอัลกอริทึม Decision Tree ใน RapidMiner

ทำเช่นนี้กับทุกๆ ข้อมูลที่ต้องการทดลอง เก็บค่าผลลัพธ์เปอร์เซ็นต์ความถูกต้อง และนำไปวิเคราะห์

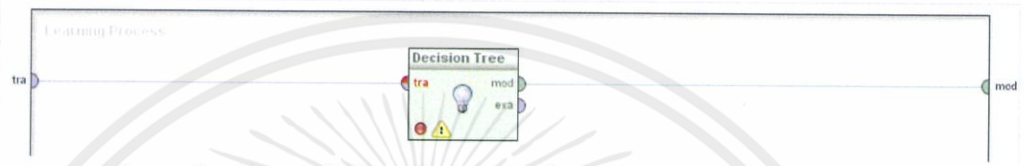
4.3.3 AdaBoost และการตั้งค่า

ทำการเลือกโมดูลอัลกอริทึม AdaBoost ซึ่งอยู่ในหมวดหมู่ Modeling>Classification and Regression>Meta Modeling ลากวางไว้ในตำแหน่งระหว่างโมดูล Set role ตัวที่ 1 และโมดูล Multiply ดังรูป



รูป 4.48 ตัวอย่างการต่อโมดูลอัลกอริทึม AdaBoost ใน RapidMiner

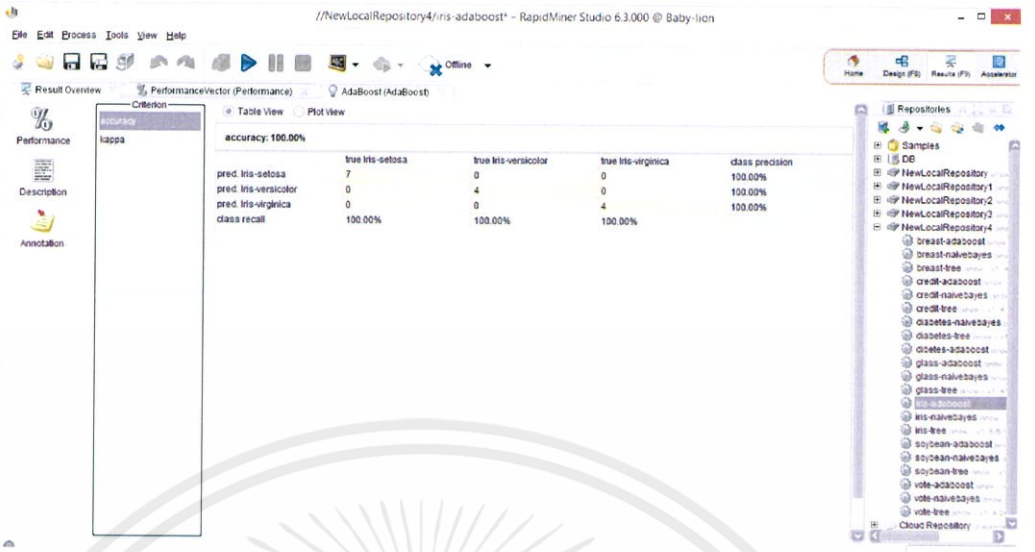
เนื่องจากอัลกอริทึม AdaBoost เป็นอัลกอริทึมแบบ nested operator ซึ่งหมายถึงเป็นอัลกอริทึมที่มีอีกอัลกอริทึมภายในได้ ดังนั้นการต่อโมดูลกระบวนการ เราจึงจะต้องทำการต่อโมดูลย่อยภายในโมดูลอัลกอริทึม AdaBoost ให้เรียบร้อยด้วย ไม่เช่นนั้นอัลกอริทึม AdaBoost จะไม่สามารถทำงานได้ ขั้นตอนในการต่อโมดูลย่อยภายใน AdaBoost สามารถทำได้โดยการดับเบิ้ลคลิกที่โมดูล AdaBoost หลังจากนั้นจะเข้าสู่หน้าต่างกระบวนการย่อย ให้ผู้ใช้ทำการเลือกโมดูลอัลกอริทึมภายใน โดยในการศึกษาทดลองนี้จะเลือกใช้โมดูลอัลกอริทึม Decision tree ดังรูป



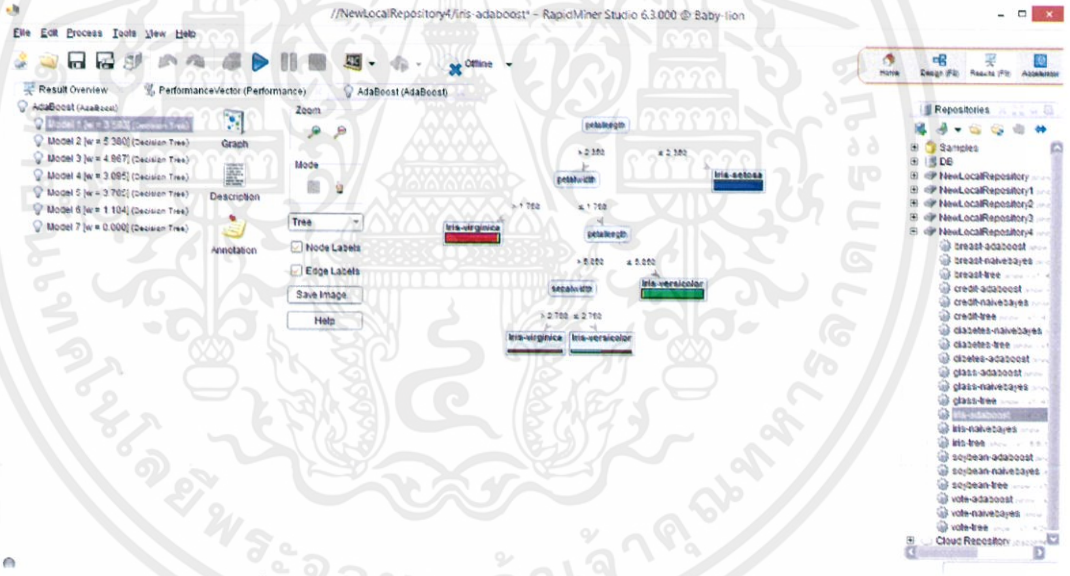
รูป 4.49 ตัวอย่างการต่อโมดูลย่อยอัลกอริทึม Decision Tree ใน RapidMiner

โดยทำการตั้งค่าพารามิเตอร์ต่างๆของอัลกอริทึม Decision Tree ตามข้อ 4.3.2 หลังจากนั้นทำการคลิกที่หน้าต่าง Process เพื่อกลับมาสู่หน้าต่าง main process หลังจากนั้นทำการตั้งค่าพารามิเตอร์ต่างๆของอัลกอริทึม AdaBoost โดยการคลิกที่โมดูลอัลกอริทึม AdaBoost ดังนี้ ซึ่งเมื่อทำการคลิกที่โมดูลอัลกอริทึม AdaBoost แล้วจะปรากฏว่ามีเพียงพารามิเตอร์ iterations เพียงพารามิเตอร์เดียวให้ผู้ใช้ทำการตั้งค่า ในการทดลองเราจะทำการกำหนดค่าให้มีค่าเท่ากับ 10 ซึ่งพารามิเตอร์นี้เป็นการกำหนด จำนวนครั้งในการทำซ้ำสูงสุดทั้งหมดกี่รอบ

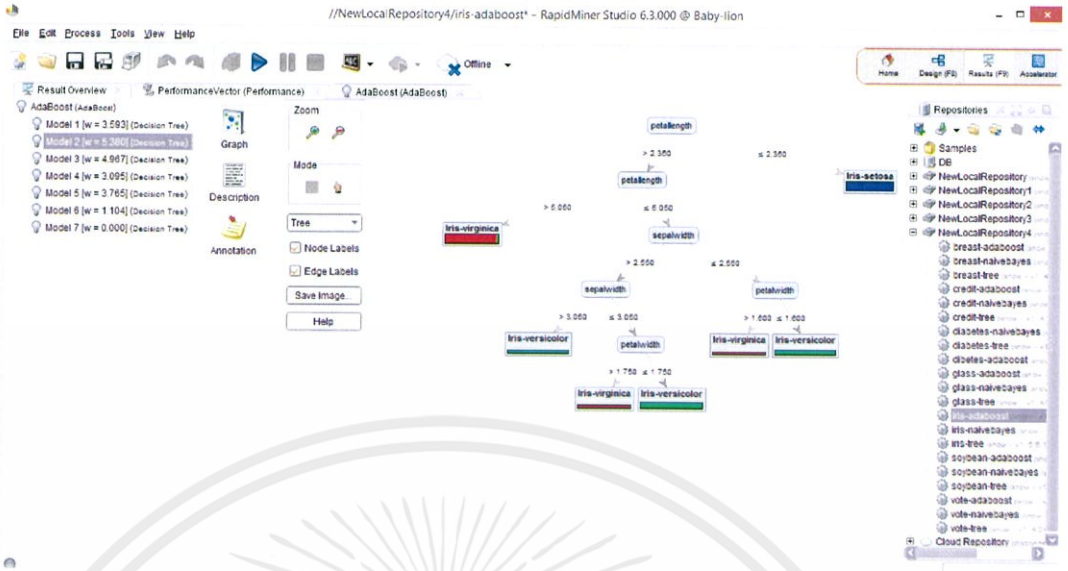
หลังจากทำการตั้งค่าพารามิเตอร์ของทั้งอัลกอริทึม AdaBoost และอัลกอริทึมย่อย Decision tree เรียบร้อยแล้วให้ทำการคลิก Run เพื่อดูผลลัพธ์ที่เกิดขึ้น จะได้ผลลัพธ์ดังรูป



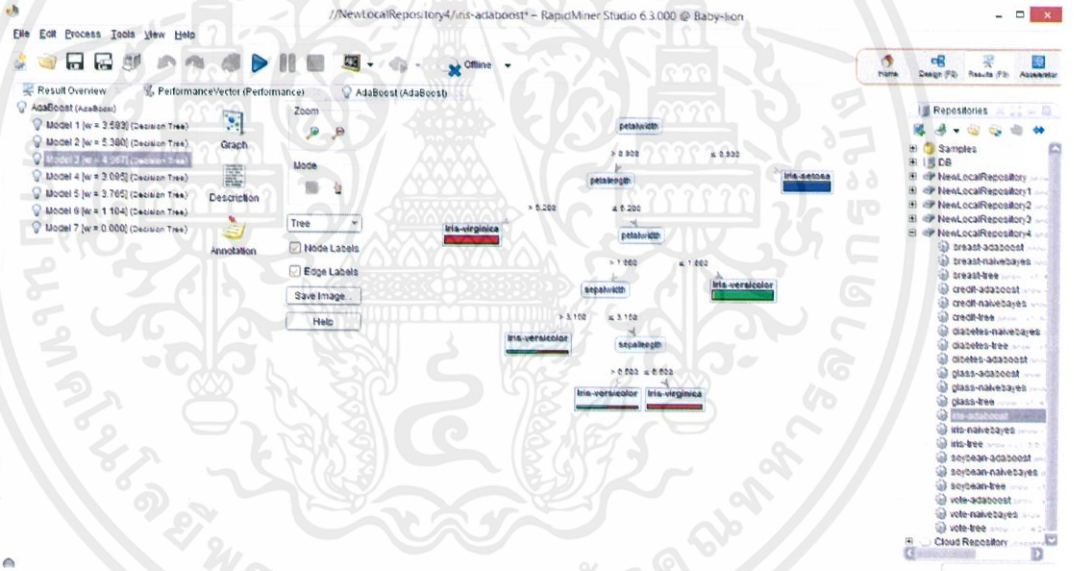
รูป 4.50 หน้าต่างผลลัพธ์ Performance Vector การทำนายค่าอัลกอริทึม AdaBoost ใน RapidMiner



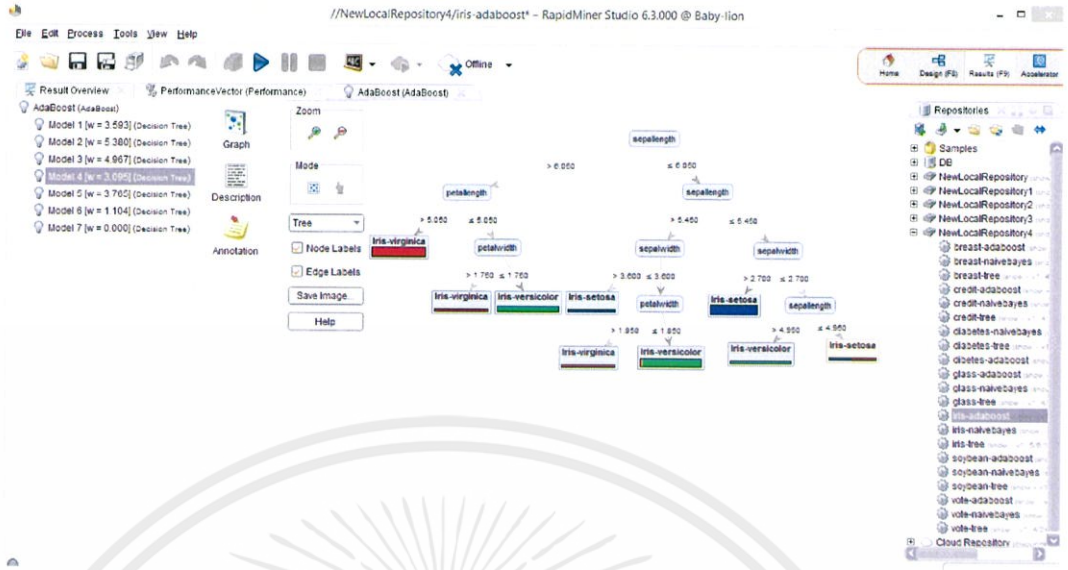
รูป 4.51 หน้าต่างผลลัพธ์ AdaBoost model 1 การทำนายค่าอัลกอริทึม AdaBoost ใน RapidMiner



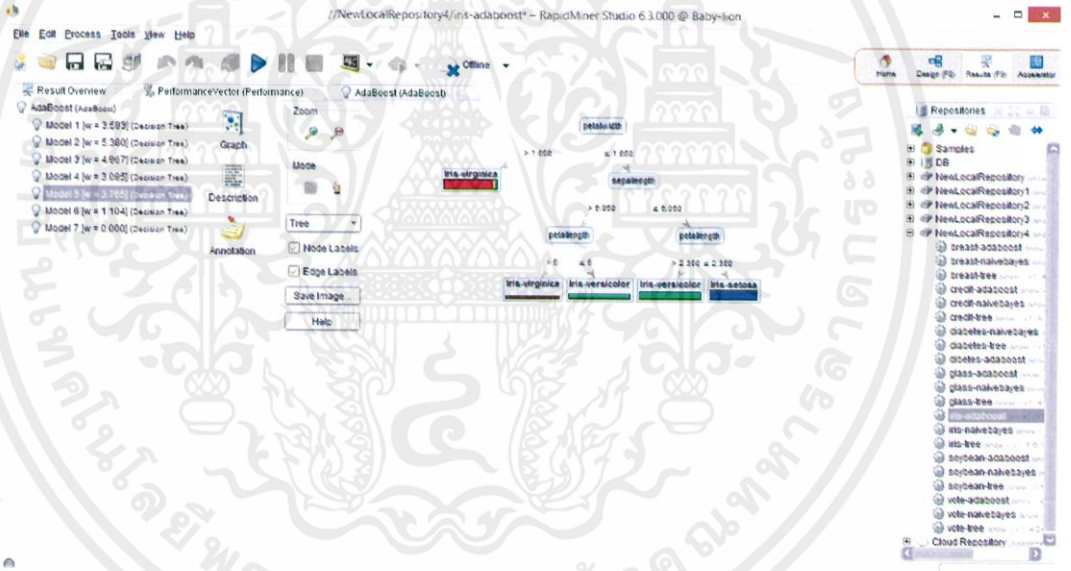
รูป 4.52 หน้าต่างผลลัพธ์ AdaBoost model 2 การทำนายค่าอัลกอริทึม AdaBoost ใน RapidMiner



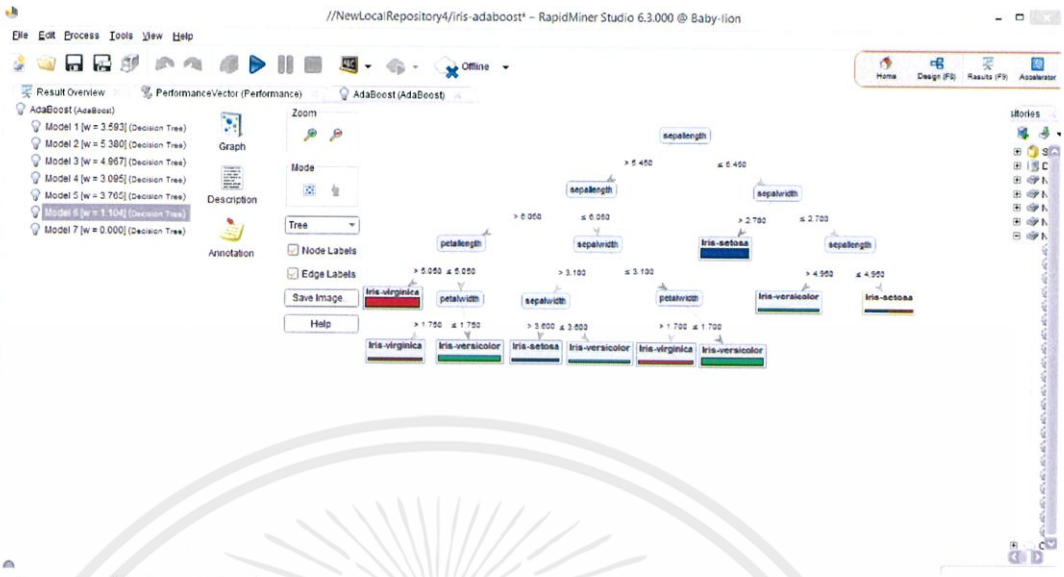
รูป 4.53 หน้าต่างผลลัพธ์ AdaBoost model 3 การทำนายค่าอัลกอริทึม AdaBoost ใน RapidMiner



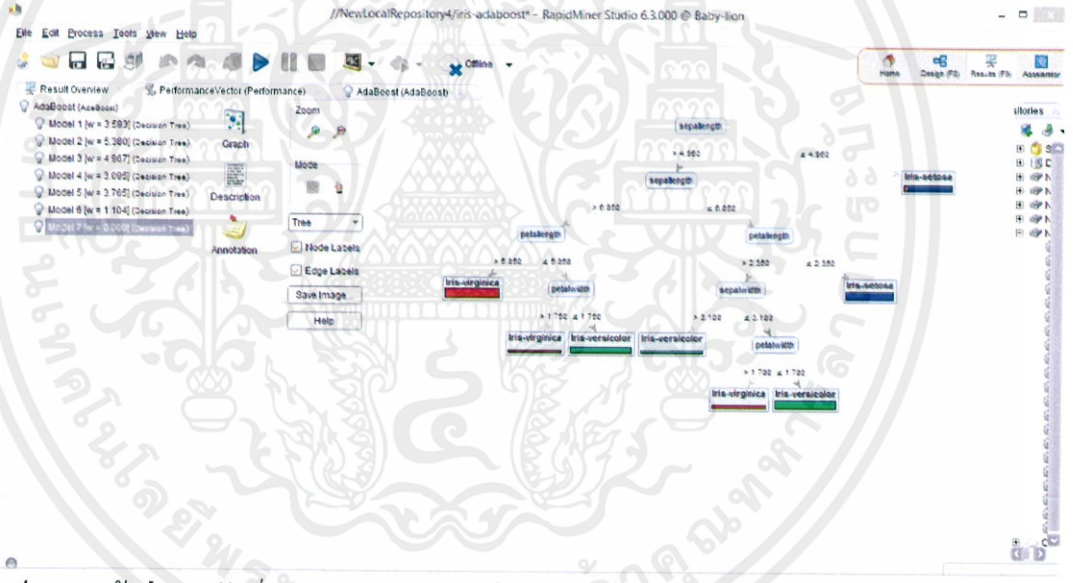
รูป 4.54 หน้าต่างผลลัพธ์ AdaBoost model 4 การทำนายค่าอัลกอริทึม AdaBoost ใน RapidMiner



รูป 4.55 หน้าต่างผลลัพธ์ AdaBoost model 5 การทำนายค่าอัลกอริทึม AdaBoost ใน RapidMiner



รูป 4.56 หน้าต่างผลลัพธ์ AdaBoost model 6 การทำนายค่าอัลกอริทึม AdaBoost ใน RapidMiner



รูป 4.57 หน้าต่างผลลัพธ์ AdaBoost model 7 การทำนายค่าอัลกอริทึม AdaBoost ใน RapidMiner

จากผลลัพธ์ที่ได้จะเห็นว่าอัลกอริทึม AdaBoost มีการสร้าง Model decision tree ทั้งหมด 7 model ที่แตกต่างกัน ซึ่งในแต่ละชุดข้อมูลอาจมีการสร้างจำนวนโมเดลที่แตกต่างกันขึ้นอยู่กับขนาดและลักษณะของชุดข้อมูล

ทำเช่นนี้กับทุกๆข้อมูลที่ต้องการทดลอง เก็บค่าผลลัพธ์เปอร์เซ็นต์ความถูกต้อง และนำไปวิเคราะห์

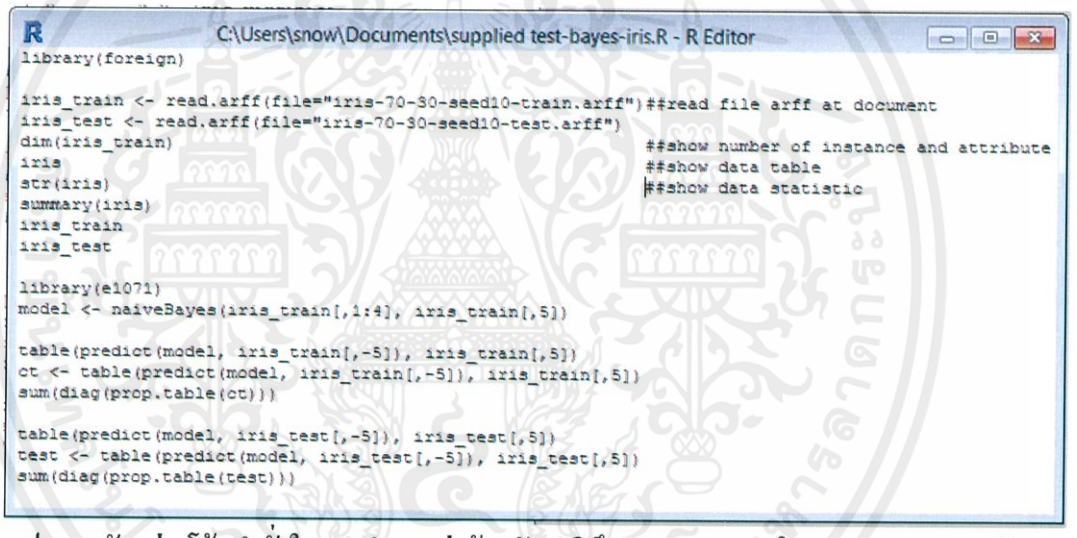
4.4 การทดลองทำเหมืองข้อมูลด้วยซอฟต์แวร์อาร์ ดาต้ามายนิ่ง

ในการทดลองศึกษาผลลัพธ์ในด้านการเปรียบเทียบประสิทธิภาพของซอฟต์แวร์ จะมีการทดลองทำเหมืองข้อมูลในซอฟต์แวร์ราปิดมายเนอร์ โดยการควบคุมชุดข้อมูลเดียวกันกับชุดข้อมูลที่ทดลองในซอฟต์แวร์อื่นๆ โดยการทดลองในซอฟต์แวร์อาร์ ดาต้ามายนิ่งผู้ใช้จะต้องทำการเขียนโค้ดเพื่อสั่งรันในหน้า R console ซึ่งในการเรียกใช้แต่ละอัลกอริทึมในอาร์ จะมีการเรียกใช้หรือใช้โค้ดคำสั่งที่แตกต่างกันไป ดังต่อไปนี้

4.4.1 การเขียนโค้ดอัลกอริทึม Naïve Bayes

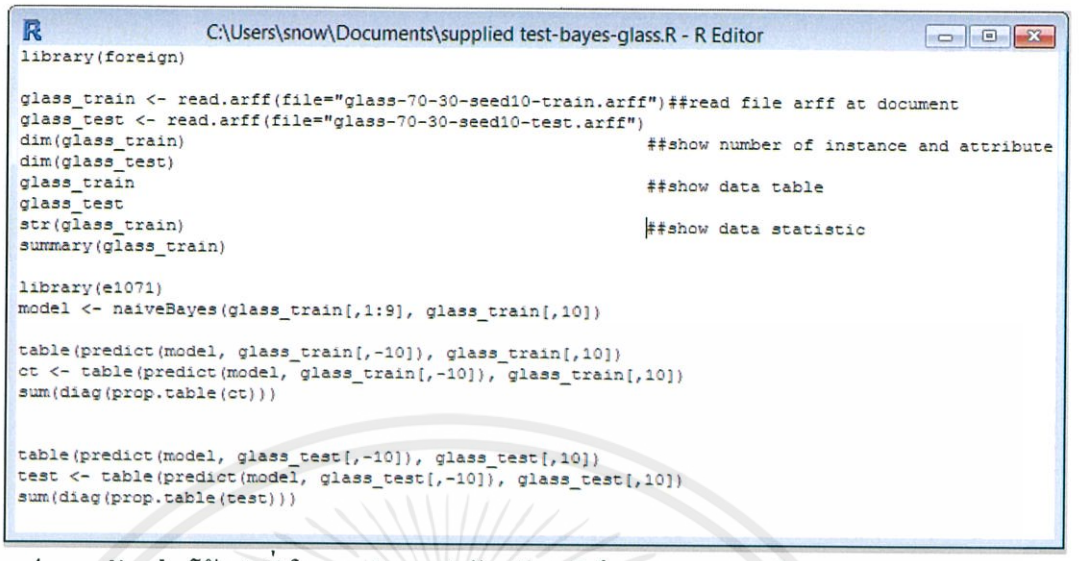
สำหรับการทำเหมืองข้อมูล ด้วยวิธีการทำนายค่าข้อมูล โดยใช้อัลกอริทึม Naïve Bayes ในซอฟต์แวร์ R Data Mining สามารถเขียนโค้ดคำสั่งของชุดข้อมูลต่างๆ ได้ดังนี้

- ชุดข้อมูลดอกไอริส



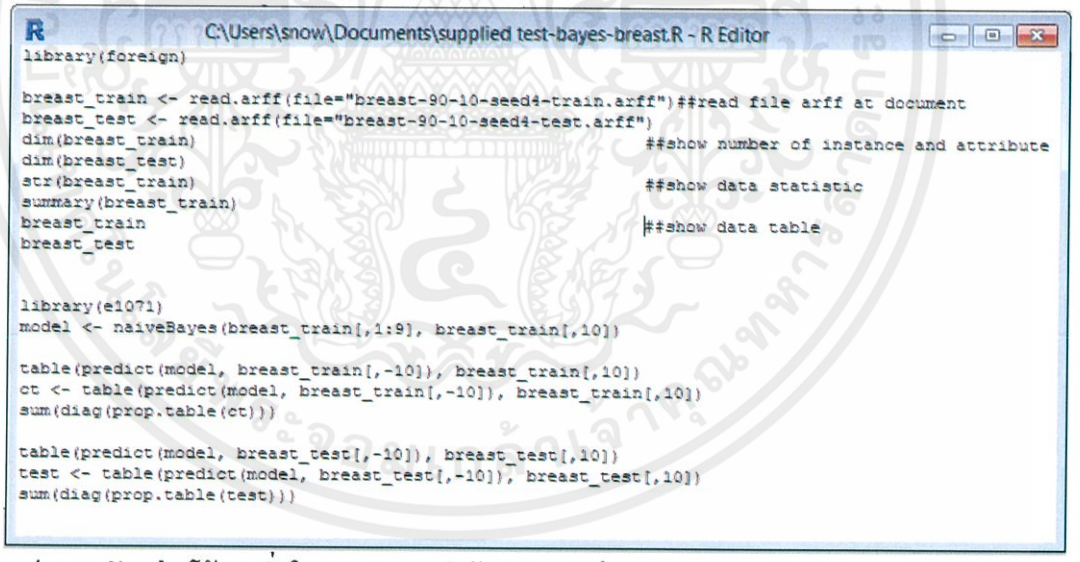
รูป 4.58 ตัวอย่างโค้ดคำสั่งในการทำนายค่าด้วยอัลกอริทึม Naïve Bayes ใน R Data Mining กับชุดข้อมูลดอกไอริส (Iris.arff)

- ชุดข้อมูลกระจก



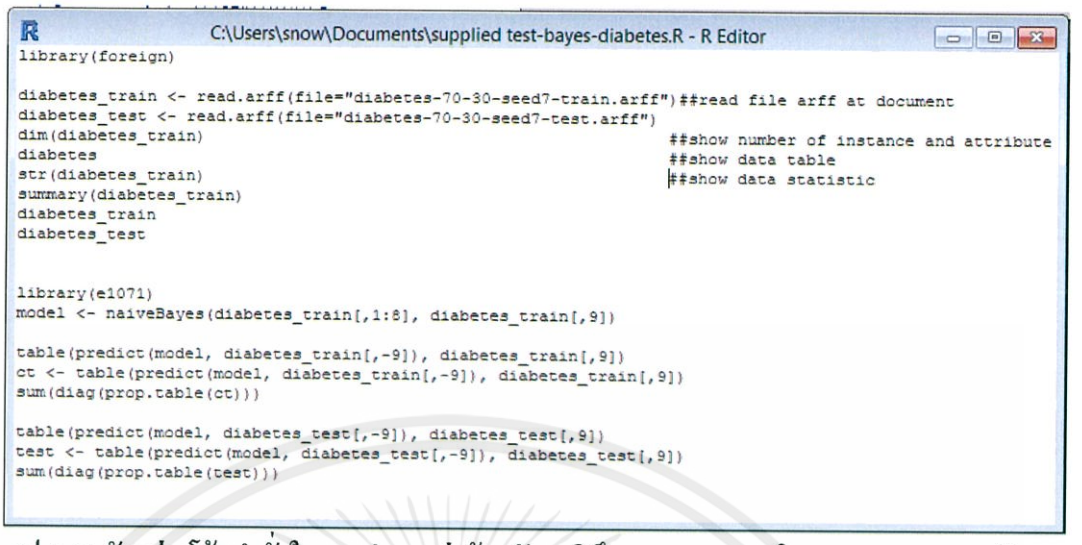
รูป 4.59 ตัวอย่างโค้ดคำสั่งในการทำนายค่าด้วยอัลกอริทึม Naïve Bayes ใน R Data Mining กับชุดข้อมูลกระจก (Glass.arff)

- ชุดข้อมูลมะเร็งเต้านม



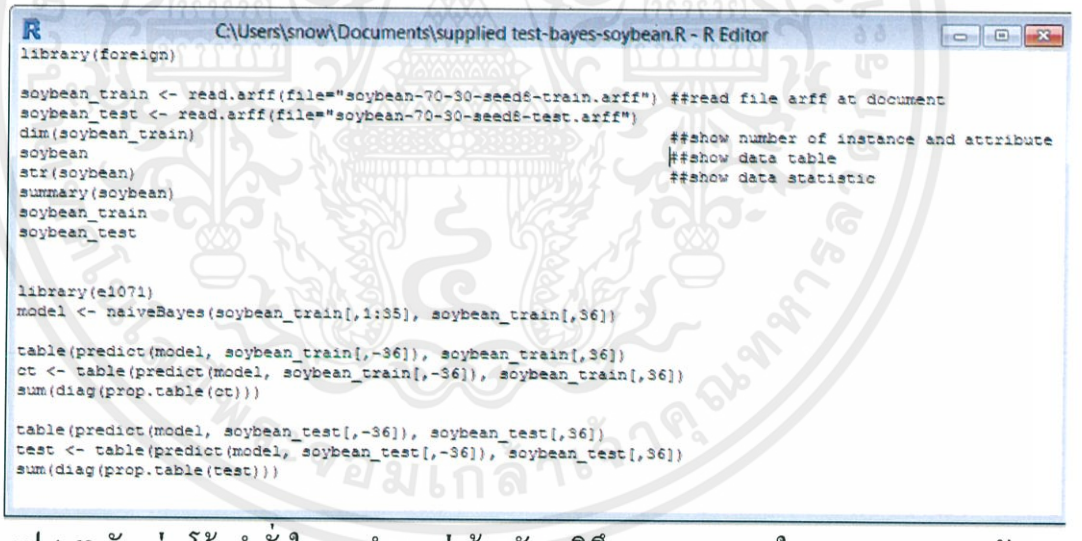
รูป 4.60 ตัวอย่างโค้ดคำสั่งในการทำนายค่าด้วยอัลกอริทึม Naïve Bayes ใน R Data Mining กับชุดข้อมูลมะเร็งเต้านม (Breast-cancer.arff)

- ชุดข้อมูลเบาหวาน



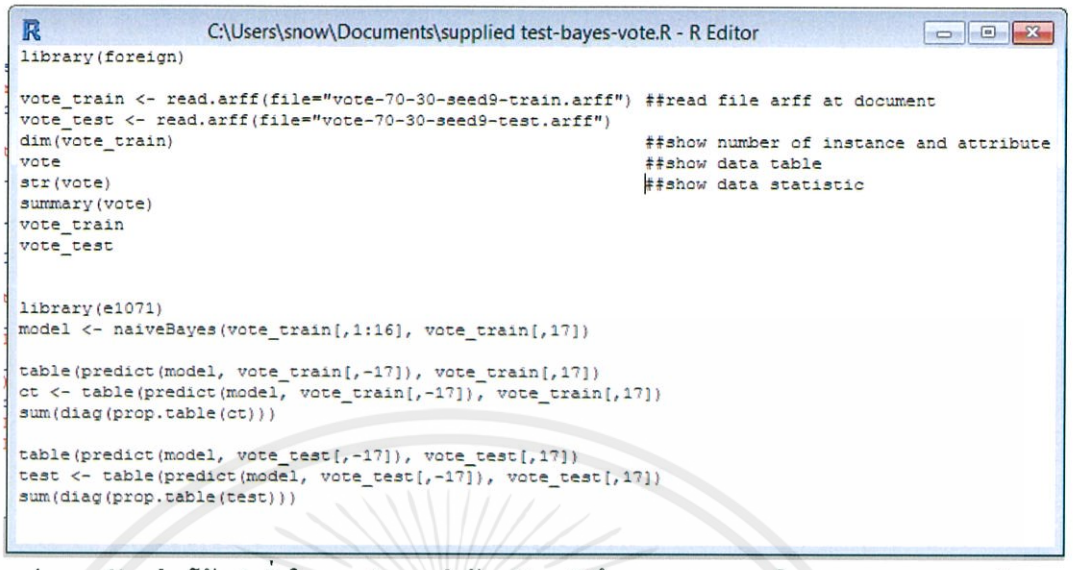
รูป 4.61 ตัวอย่างโค้ดคำสั่งในการทำนายค่าด้วยอัลกอริทึม Naïve Bayes ใน R Data Mining กับชุดข้อมูลเบาหวาน (Diabetes.arff)

- ชุดข้อมูลถั่วเหลือง



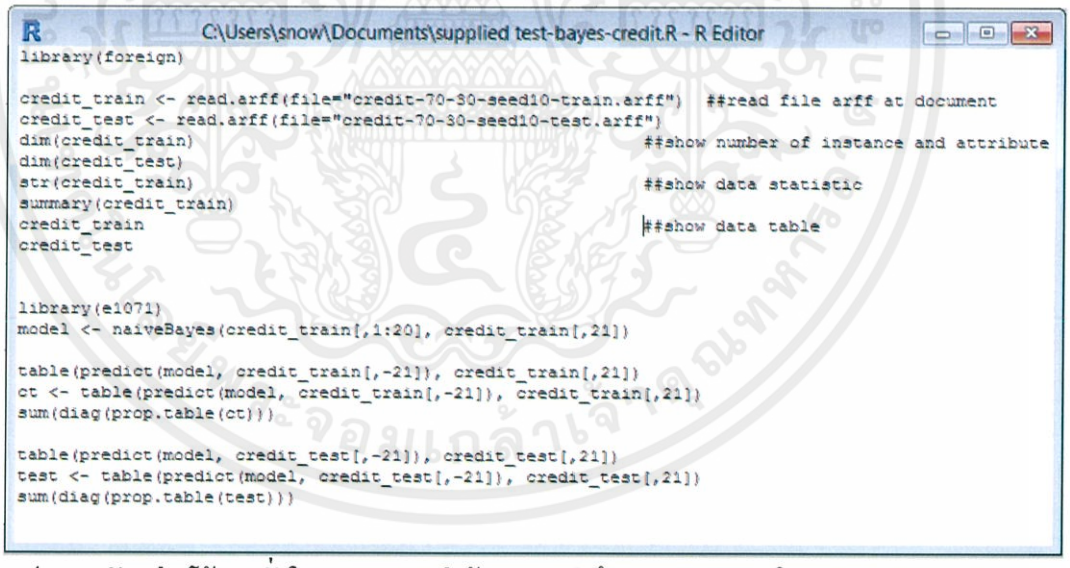
รูป 4.62 ตัวอย่างโค้ดคำสั่งในการทำนายค่าด้วยอัลกอริทึม Naïve Bayes ใน R Data Mining กับชุดข้อมูลถั่วเหลือง (Soybean.arff)

- ชุดข้อมูลการลงคะแนนเสียงเลือกตั้ง



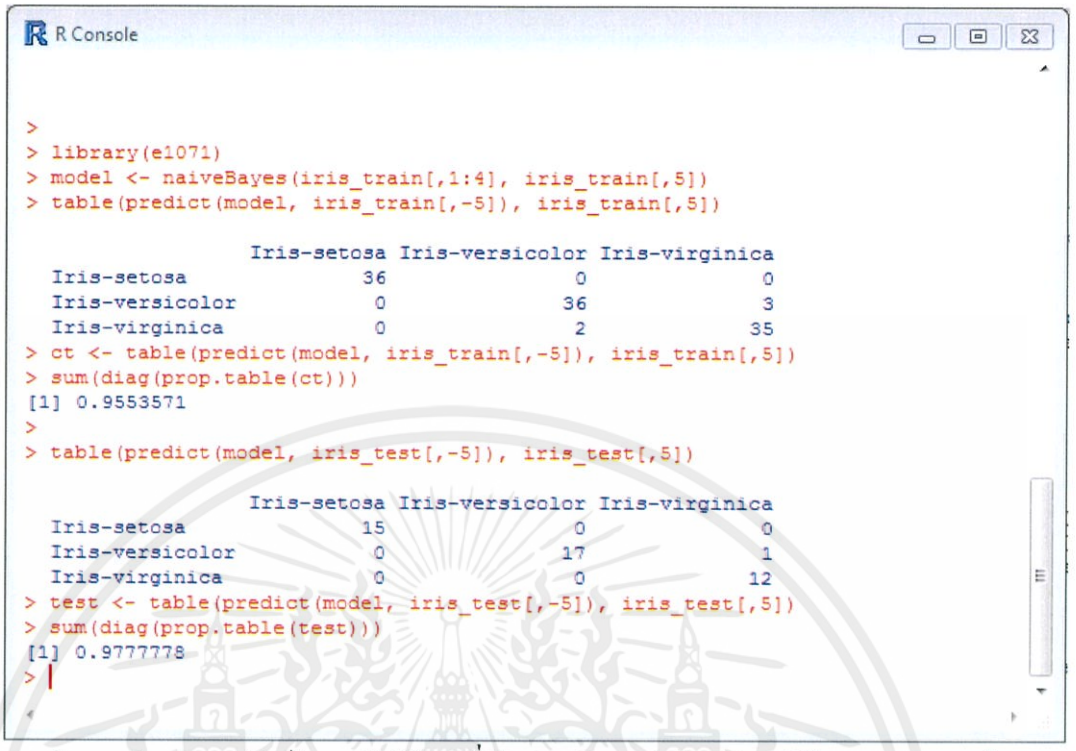
รูป 4.63 ตัวอย่างโค้ดคำสั่งในการทำนายค่าด้วยอัลกอริทึม Naïve Bayes ใน R Data Mining กับชุดข้อมูลการลงคะแนนเสียงเลือกตั้ง (Vote.arff)

- ชุดข้อมูลสินเชื่อ



รูป 4.64 ตัวอย่างโค้ดคำสั่งในการทำนายค่าด้วยอัลกอริทึม Naïve Bayes ใน R Data Mining กับชุดข้อมูลการสินเชื่อ (Credit-g.arff)

หลังจากเขียนโค้ดคำสั่งตามตัวอย่างรูปข้างต้นแล้ว ทำการรันโค้ดคำสั่งทั้งหมด โดยจะแสดงผลลัพธ์ ยกตัวอย่างดังรูป



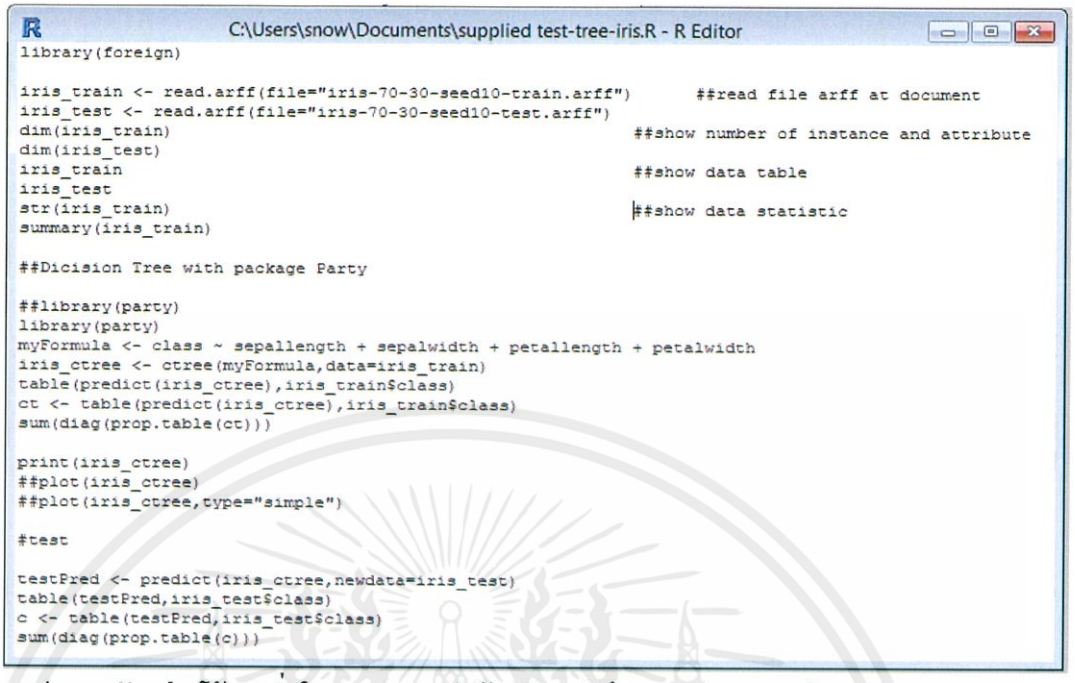
รูป 4.65 ตัวอย่างผลลัพธ์ในการรันโค้ดคำสั่งในการทำนายค่าด้วยอัลกอริทึม Naïve Bayes ใน R Data Mining กับชุดข้อมูลดอกไอริส (Iris.arff)

ซึ่งในการรันโค้ดคำสั่งอัลกอริทึม Naïve Bayes กับชุดข้อมูลอื่นๆ อาจมีการเปลี่ยนแปลงคำสั่งในบางส่วนตามความเหมาะสม ทำเช่นนี้กับทุกๆข้อมูลที่ต้องการทดลอง หาผลลัพธ์และนำค่าที่ได้ไปวิเคราะห์

4.4.2 การเขียนโค้ดอัลกอริทึม Decision Tree

สำหรับการทำเหมืองข้อมูล ด้วยวิธีการทำนายค่าข้อมูล โดยใช้อัลกอริทึม Decision Tree ในซอฟต์แวร์ R Data Mining สามารถเขียนโค้ดคำสั่งของชุดข้อมูลต่างๆ ได้ดังนี้

- ชุดข้อมูลดอกไอริส

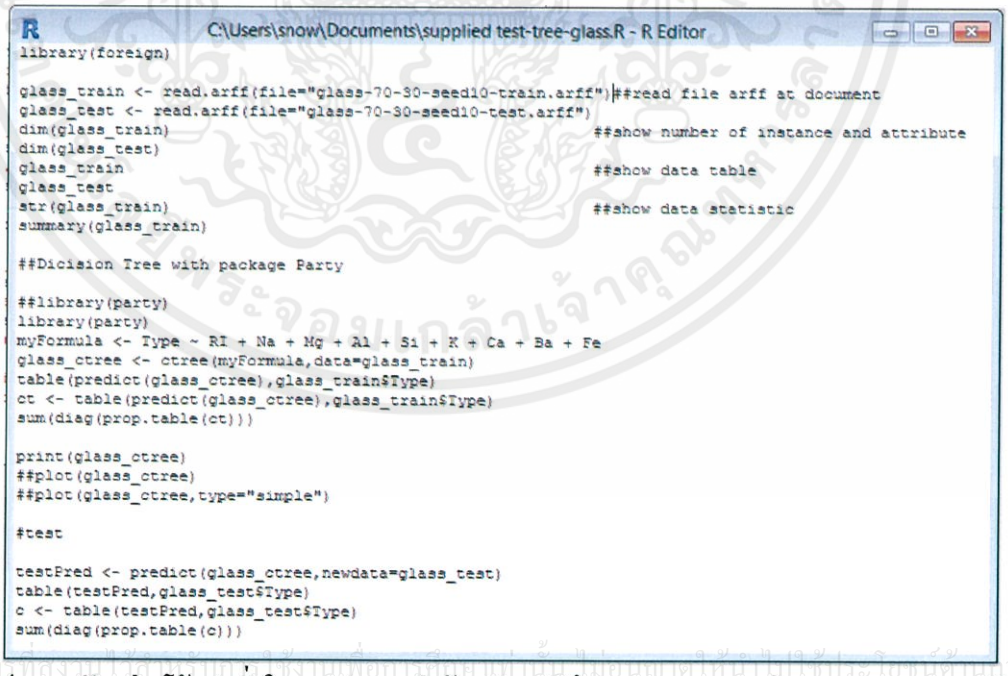


C:\Users\snow\Documents\supplied test-tree-iris.R - R Editor

```
library(foreign)

iris_train <- read.arff(file="iris-70-30-seed10-train.arff")      ##read file arff at document
iris_test <- read.arff(file="iris-70-30-seed10-test.arff")
dim(iris_train)                                                   ##show number of instance and attribute
dim(iris_test)
iris_train                                                        ##show data table
iris_test
str(iris_train)                                                   ##show data statistic
summary(iris_train)

##Dicision Tree with package Party

##library(party)
library(party)
myFormula <- class ~ sepallength + sepalwidth + petallength + petalwidth
iris_ctree <- ctree(myFormula,data=iris_train)
table(predict(iris_ctree),iris_train$class)
ct <- table(predict(iris_ctree),iris_train$class)
sum(diag(prop.table(ct)))

print(iris_ctree)
##plot(iris_ctree)
##plot(iris_ctree,type="simple")

#test

testPred <- predict(iris_ctree,newdata=iris_test)
table(testPred,iris_test$class)
c <- table(testPred,iris_test$class)
sum(diag(prop.table(c)))
```

รูป 4.66 ตัวอย่างโค้ดคำสั่งในการทำนายค่าด้วยอัลกอริทึม Decision Tree ใน R Data Mining กับชุดข้อมูลดอกไอริส (Iris.arff)

- ชุดข้อมูลกระจก



C:\Users\snow\Documents\supplied test-tree-glass.R - R Editor

```
library(foreign)

glass_train <- read.arff(file="glass-70-30-seed10-train.arff")##read file arff at document
glass_test <- read.arff(file="glass-70-30-seed10-test.arff")
dim(glass_train)                                              ##show number of instance and attribute
dim(glass_test)
glass_train                                                   ##show data table
glass_test
str(glass_train)                                              ##show data statistic
summary(glass_train)

##Dicision Tree with package Party

##library(party)
library(party)
myFormula <- Type ~ RI + Na + Mg + Al + Si + K + Ca + Ba + Fe
glass_ctree <- ctree(myFormula,data=glass_train)
table(predict(glass_ctree),glass_train$Type)
ct <- table(predict(glass_ctree),glass_train$Type)
sum(diag(prop.table(ct)))

print(glass_ctree)
##plot(glass_ctree)
##plot(glass_ctree,type="simple")

#test

testPred <- predict(glass_ctree,newdata=glass_test)
table(testPred,glass_test$Type)
c <- table(testPred,glass_test$Type)
sum(diag(prop.table(c)))
```

รูป 4.67 ตัวอย่างโค้ดคำสั่งในการทำนายค่าด้วยอัลกอริทึม Decision Tree ใน R Data Mining กับชุดข้อมูลกระจก (Glass.arff)

- ชุดข้อมูลมะเร็งเต้านม

C:\Users\snow\Documents\supplied test-tree-breast.R - R Editor

```
library(foreign)

breast_train <- read.arff(file="breast-80-20-seed10-train.arff")  ##read file arff at document
breast_test <- read.arff(file="breast-80-20-seed10-test.arff")
dim(breast_train)                                                 ##show number of instance and attribute
dim(breast_test)
breast_train                                                      ##show data table
breast_test
str(breast_train)                                                 ##show data statistic
summary(breast_train)

library(party)
myFormula <- Class ~ age + menopause + tumorSize + invNodes + nodeCaps + degMalig + breast
+ breastQuad + irradiat
breast_ctree <- ctree(myFormula,data=breast_train)
table(predict(breast_ctree),breast_train$Class)
ct <- table(predict(breast_ctree),breast_train$Class)
sum(diag(prop.table(ct)))

print(breast_ctree)
##plot(breast_ctree)
##plot(breast_ctree,type="simple")

testPred <- predict(breast_ctree,newdata=breast_test)
table(testPred,breast_test$Class)
c <- table(testPred,breast_test$Class)
sum(diag(prop.table(c)))
```

รูป 4.68 ตัวอย่างโค้ดคำสั่งในการทำนายค่าด้วยอัลกอริทึม Decision Tree ใน R Data Mining กับชุดข้อมูลมะเร็งเต้านม (Breast-cancer.arff)

- ชุดข้อมูลเบาหวาน

C:\Users\snow\Documents\supplied test-tree-diabetes.R - R Editor

```
library(foreign)

diabetes_train <- read.arff(file="diabetes-70-30-seed10-train.arff")##read file arff at document
diabetes_test <- read.arff(file="diabetes-70-30-seed10-test.arff")
dim(diabetes_train)                                                ##show number of instance and attribute
dim(diabetes_test)
diabetes_train                                                     ##show data table
diabetes_test
str(diabetes_train)                                                ##show data statistic
summary(diabetes_train)

##Dicision Tree with package Party

##library(party)
library(party)
myFormula <- class ~ preg + plas + pres + skin + insu + mass + pedi + age
diabetes_ctree <- ctree(myFormula,data=diabetes_train)
table(predict(diabetes_ctree),diabetes_train$class)
ct <- table(predict(diabetes_ctree),diabetes_train$class)
sum(diag(prop.table(ct)))

print(diabetes_ctree)
##plot(diabetes_ctree)
##plot(diabetes_ctree,type="simple")

#test

testPred <- predict(diabetes_ctree,newdata=diabetes_test)
table(testPred,diabetes_test$class)
c <- table(testPred,diabetes_test$class)
sum(diag(prop.table(c)))
```


รูป 4.69 ตัวอย่างโค้ดคำสั่งในการทำนายค่าด้วยอัลกอริทึม Decision Tree ใน R Data Mining กับชุดข้อมูลเบาหวาน (Diabetes.arff)

- ชุดข้อมูลถั่วเหลือง

```
C:\Users\snow\Documents\supplied test-tree-soybean.R - R Editor
library(foreign)

soybean_train <- read.arff(file="soybean-90-10-seed10-train.arff")##read file arff at document
soybean_test <- read.arff(file="soybean-90-10-seed10-test.arff")
dim(soybean_train)                                              ##show number of instance and attribute
dim(soybean_test)
##soybean_train                                                 ##show data table
##soybean_test
str(soybean_train)                                              ##show data statistic
summary(soybean_train)

library(party)
levels(soybean_train$class)
levels(soybean_test$class)
myFormula <- class ~ date + plantStand + precip + temp + hail + cropHist + areaDamaged + severity + seedTmt
+ germination + plantGrowth + leaves + leafspotsHalo + leafspotsMarg + leafspotSize + leafShread + leafMalf
+ leafMild + stem + lodging + stemCankers + cankerLesion + fruitingBodies + externalDecay + mycelium
+ intDiscolor + sclerotia + fruitPods + fruitSpots + seed + moldGrowth + seedDiscolor + seedSize + shriveling
+ roots
soybean_ctree <- ctree(myFormula,data=soybean_train)
table(predict(soybean_ctree),soybean_train$class)
ct <- table(predict(soybean_ctree),soybean_train$class)
sum(diag(prop.table(ct)))


testPred <- predict(soybean_ctree,newdata=soybean_test)
table(testPred,soybean_test$class)
c <- table(testPred,soybean_test$class)
sum(diag(prop.table(c)))
```

รูป 4.70 ตัวอย่างโค้ดคำสั่งในการทำนายค่าด้วยอัลกอริทึม Decision Tree ใน R Data Mining กับชุดข้อมูลถั่วเหลือง (Soybean.arff)

- ชุดข้อมูลการลงคะแนนเสียงเลือกตั้ง

```
C:\Users\snow\Documents\supplied test-tree-vote.R - R Editor
library(foreign)

vote_train <- read.arff(file="vote-70-30-seed10-train.arff")##read file arff at document
vote_test <- read.arff(file="vote-70-30-seed10-test.arff")
dim(vote_train)                                             ##show number of instance and attribute
dim(vote_test)
vote_train                                                  ##show data table
vote_test
str(vote_train)                                             ##show data statistic
summary(vote_train)

##Dicision Tree with package Party

##library(party)
library(party)
myFormula <- Class ~ handicappedInfants + waterProjectCostSharing + adoptionOfTheBudgetResolution
+ physicianFeeFreeze + elSalvadorAid + religiousGroupsInSchools + antiSatelliteTestBan
+ aidToNicaraguanContras + mxMissile + immigration + synfuelsCorporationCutback + educationSpending
+ superfundRightToSue + crime + dutyFreeExports + exportAdministrationActSouthAfrica
vote_ctree <- ctree(myFormula,data=vote_train)
table(predict(vote_ctree),vote_train$Class)
ct <- table(predict(vote_ctree),vote_train$Class)
sum(diag(prop.table(ct)))

print(vote_ctree)
##plot(vote_ctree)
##plot(vote_ctree,type="simple")

#test

testPred <- predict(vote_ctree,newdata=vote_test)
table(testPred,vote_test$Class)
c <- table(testPred,vote_test$Class)
sum(diag(prop.table(c)))
```


รูป 4.71 ตัวอย่างโค้ดคำสั่งในการทำนายค่าด้วยอัลกอริทึม Decision Tree ใน R Data Mining กับชุดข้อมูลการลงคะแนนเสียงเลือกตั้ง (Vote.arff)

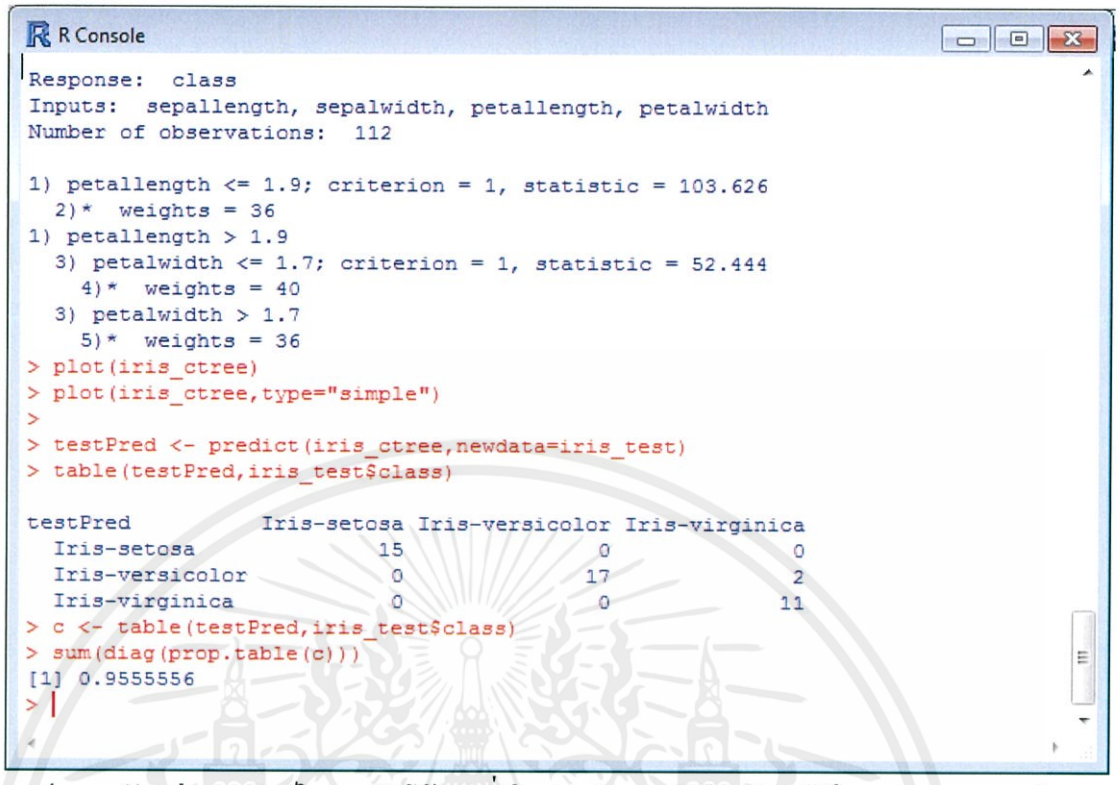
- ชุดข้อมูลสินเชื่อ

C:\Users\snow\Documents\supplied test-tree-credit.R - R Editor

```
library(foreign)

credit_train <-read.arff(file="credit-70-30-seed10-train.arff")
credit_test <-read.arff(file="credit-70-30-seed10-test.arff")

##Dicision Tree with package Party

##library(party)
library(party)
myFormula <- class ~ checking_status + duration + credit_history + purpose
+ credit_amount + savings_status + employment + installment_commitment
+ personal_status + other_parties + residence_since + property_magnitude
+ age + other_payment_plans + housing + existing_credits + job + num_dependents
+ own_telephone + foreign_worker
credit_ctree <- ctree(myFormula,data=credit_train)
table(predict(credit_ctree),credit_train$class)
ct <- table(predict(credit_ctree),credit_train$class)
sum(diag(prop.table(ct)))

print(credit_ctree)
##plot(credit_ctree)
##plot(credit_ctree,type="simple")

#test

testPred <- predict(credit_ctree,newdata=credit_test)
table(testPred,credit_test$class)
c <- table(testPred,credit_test$class)
sum(diag(prop.table(c)))
```

รูป 4.72 ตัวอย่างโค้ดคำสั่งในการทำนายค่าด้วยอัลกอริทึม Decision Tree ใน R Data Mining กับชุดข้อมูลสินเชื่อ (Credit-g.arff)

หลังจากเขียนโค้ดคำสั่งตามตัวอย่างรูปข้างต้นแล้ว ทำการรันโค้ดคำสั่งทั้งหมด โดยจะแสดงผลลัพธ์ ยกตัวอย่างดังรูป



รูป 4.73 ตัวอย่างผลลัพธ์ในการรันโค้ดคำสั่งในการทำนายค่าด้วยอัลกอริทึม Decision Tree ใน R Data Mining กับชุดข้อมูลดอกไอริส (Iris.arff)

ซึ่งในการรันโค้ดคำสั่งอัลกอริทึม Decision Tree กับชุดข้อมูลอื่นๆ อาจมีการเปลี่ยนแปลงคำสั่งในบางส่วนตามความเหมาะสม ทำเช่นนี้กับทุกๆข้อมูลที่ต้องการทดลอง หาผลลัพธ์และนำค่าที่ได้ไปวิเคราะห์

4.4.3 การเขียนโค้ดอัลกอริทึม AdaBoost

สำหรับการทำเหมืองข้อมูล ด้วยวิธีการทำนายค่าข้อมูล โดยใช้อัลกอริทึม AdaBoost ในซอฟต์แวร์ R Data Mining สามารถเขียนโค้ดคำสั่งของชุดข้อมูลต่างๆ ได้ดังนี้

- ชุดข้อมูลดอกไอริส

C:\Users\snow\Documents\supplied test-adaboost-iris.R - R Editor

```
library(adabag);
library(foreign)

iris_train<-read.arff(file="iris-70-30-seed10-train.arff")
iris_test<-read.arff(file="iris-70-30-seed10-test.arff")
dim(iris_train)
dim(iris_test)
myFormula <- class ~ sepallength + sepalwidth + petallength + petalwidth
adaboost<-boosting(myFormula, data=iris_train, boos=TRUE, mfinal=20,coeflearn='Breiman')
summary(adaboost)
##adaboost$trees
adaboost$weights
adaboost$importance
##adaboost$class
##errorevol(adaboost,adadata)
predict(adaboost,iris_train)
table_train <-predict(adaboost,iris_train)
table_train$confusion

result <- predict(adaboost,iris_test)
result$confusion
sum(diag(prop.table(result$confusion)))
```

รูป 4.74 ตัวอย่างโค้ดคำสั่งในการทำนายค่าด้วยอัลกอริทึม AdaBoost ใน R Data Mining กับชุดข้อมูลดอกไอริส (Iris.arff)

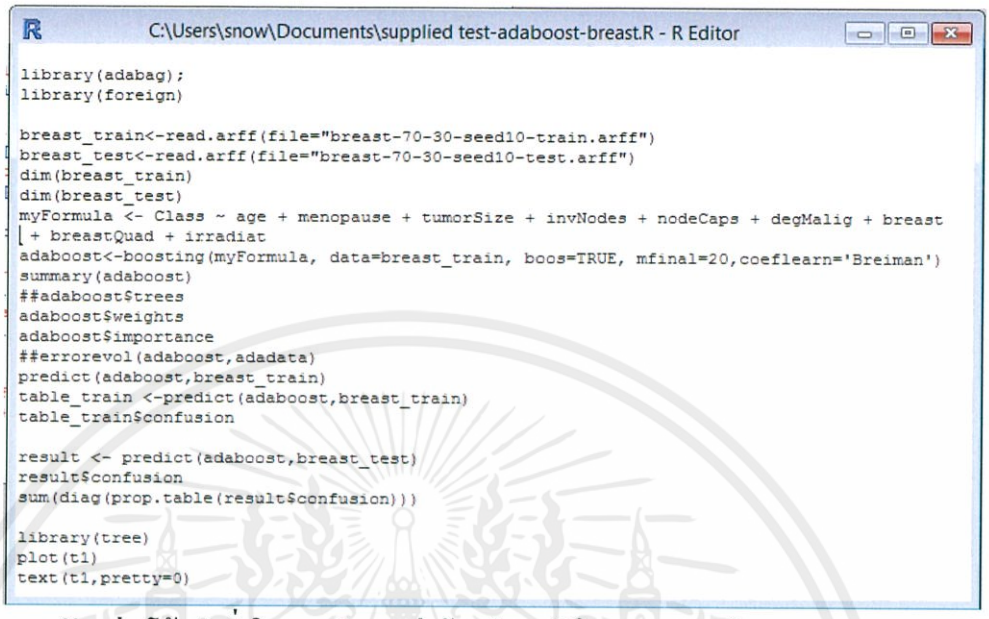
- ชุดข้อมูลกระจก

C:\Users\snow\Documents\supplied test-adaboost-glass.R - R Editor

```
library(adabag);
library(foreign)

glass_train<-read.arff(file="glass-80-20-seed10-train.arff")
glass_test<-read.arff(file="glass-80-20-seed10-test.arff")
dim(glass_train)
dim(glass_test)
myFormula <- Type ~ RI + Na + Mg + Al + Si + K + Ca + Ba + Fe
adaboost<-boosting(myFormula, data=glass_train, boos=TRUE, mfinal=20,coeflearn='Breiman')
summary(adaboost)
##adaboost$trees
adaboost$weights
adaboost$importance
##errorevol(adaboost,adadata)
predict(adaboost,glass_train)
table_train <-predict(adaboost,glass_train)
table_train$confusion

result <- predict(adaboost,glass_test)
result$confusion
sum(diag(prop.table(result$confusion)))

library(tree)
plot(t1)
text(t1,pretty=0)
```

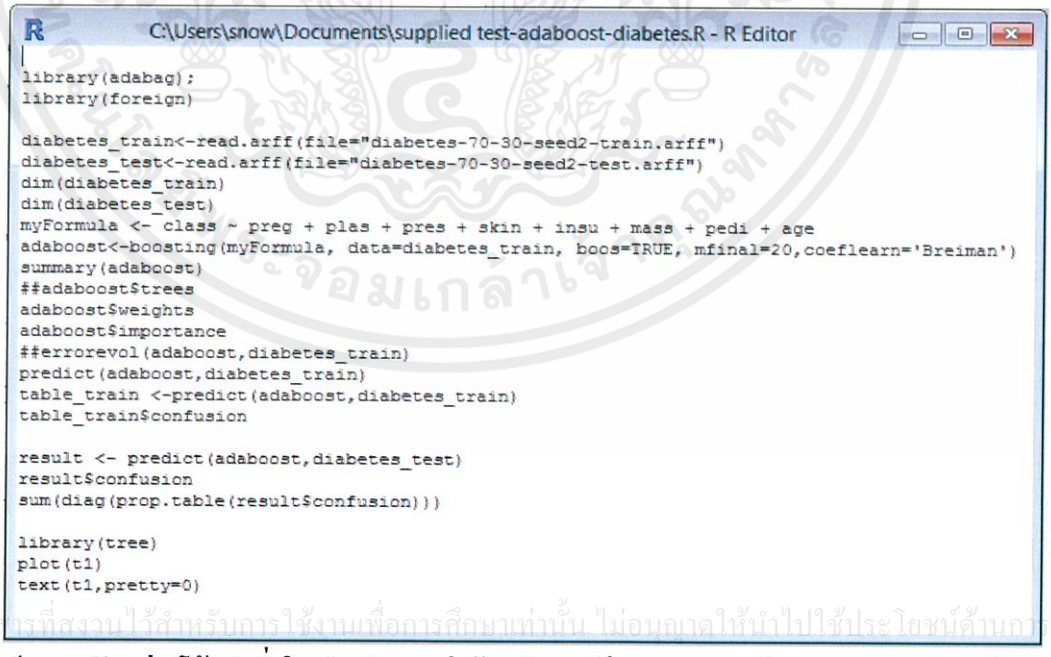
รูป 4.75 ตัวอย่างโค้ดคำสั่งในการทำนายค่าด้วยอัลกอริทึม AdaBoost ใน R Data Mining กับชุดข้อมูลกระจก (Glass.arff)

- ชุดข้อมูลมะเร็งเต้านม



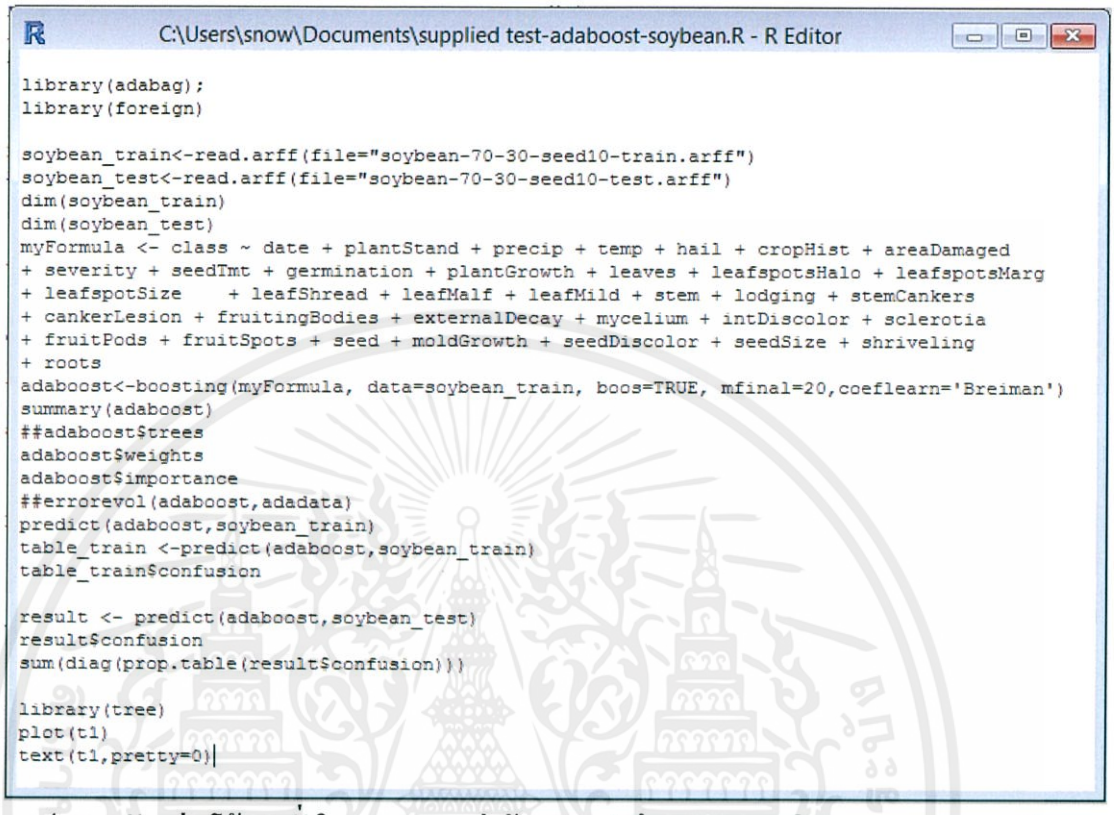
รูป 4.76 ตัวอย่างโค้ดคำสั่งในการทำนายค่าด้วยอัลกอริทึม AdaBoost ใน R Data Mining กับชุดข้อมูลมะเร็งเต้านม (Breast-cancer.arff)

- ชุดข้อมูลเบาหวาน



รูป 4.77 ตัวอย่างโค้ดคำสั่งในการทำนายค่าด้วยอัลกอริทึม AdaBoost ใน R Data Mining กับชุดข้อมูลเบาหวาน (Diabetes.arff)

- ชุดข้อมูลถั่วเหลือง



รูป 4.78 ตัวอย่างโค้ดคำสั่งในการทำนายค่าด้วยอัลกอริทึม AdaBoost ใน R Data Mining กับชุดข้อมูลถั่วเหลือง (Soybean.arff)

- ชุดข้อมูลการลงคะแนนเสียงเลือกตั้ง

C:\Users\snow\Documents\supplied test-adaboost-vote.R - R Editor

```
library(adabag);
library(foreign)

vote_train<-read.arff(file="vote-70-30-seed10-train.arff")
vote_test<-read.arff(file="vote-70-30-seed10-test.arff")
dim(vote_train)
dim(vote_test)
myFormula <- class ~ preg + plas + pres + skin + insu + mass + pedi + age
adaboost<-boosting(myFormula, data=vote_train, boos=TRUE, mfinal=20,coeflearn='Breiman')
summary(adaboost)
##adaboost$trees
adaboost$weights
adaboost$importance
##errorevol(adaboost,vote_train)
predict(adaboost,vote_train)
table_train <-predict(adaboost,vote_train)
table_train$confusion

result <- predict(adaboost,vote_test)
result$confusion
sum(diag(prop.table(result$confusion)))

library(tree)
plot(t1)
text(t1,pretty=0)
```

รูป 4.79 ตัวอย่างโค้ดคำสั่งในการทำนายค่าด้วยอัลกอริทึม AdaBoost ใน R Data Mining กับชุดข้อมูลการลงคะแนนเสียงเลือกตั้ง (Vote.arff)

- ชุดข้อมูลสินเชื่อ

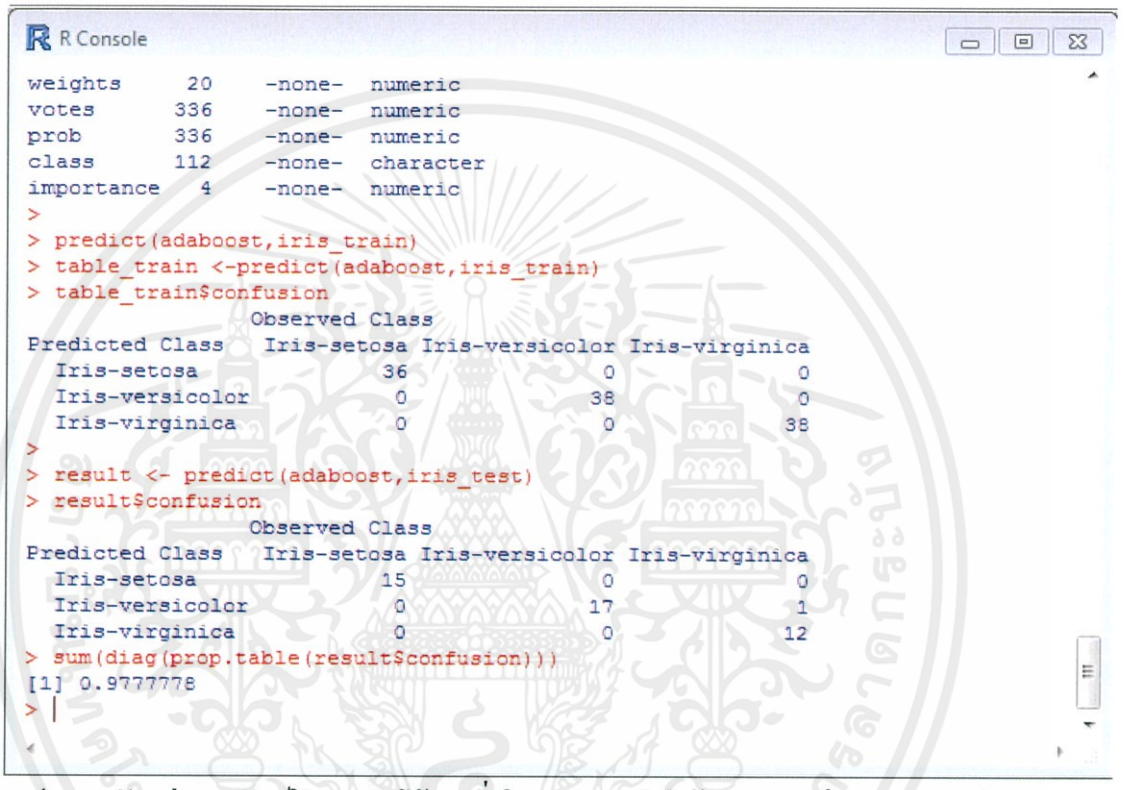
C:\Users\snow\Documents\supplied test-adaboost-credit.R - R Editor

```
library(adabag);
library(foreign)

credit_train <-read.arff(file="credit-90-10-seed7-train.arff")
credit_test <-read.arff(file="credit-90-10-seed7-test.arff")

myFormula <- class ~ checking_status + duration + credit_history + purpose + credit_amount
 + savings_status + employment + installment_commitment + personal_status + other_parties
+ residence_since + property_magnitude + age + other_payment_plans + housing
+ existing_credits + job + num_dependents + own_telephone + foreign_worker
adaboost<-boosting(myFormula, data=credit_train, boos=TRUE, mfinal=20,coeflearn='Breiman')
summary(adaboost)
adaboost$weights
adaboost$importance
predict(adaboost,credit_train)
table_train <-predict(adaboost,credit_train)
table_train$confusion

result <- predict(adaboost,credit_test)
result$confusion
sum(diag(prop.table(result$confusion)))

t1<-adaboost$trees[[1]]

library(tree)
plot(t1)
text(t1,pretty=0)
```


รูป 4.80 ตัวอย่างโค้ดคำสั่งในการทำนายค่าด้วยอัลกอริทึม AdaBoost ใน R Data Mining กับชุดข้อมูลสินเชื่อ (Credit-g.arff)

โดยการใช้อัลกอริทึม AdaBoost ในซอฟต์แวร์ R Data Mining จะคล้ายกับ AdaBoost ในซอฟต์แวร์อื่นๆ คือ มีอัลกอริทึมย่อยภายใน ซึ่งใน R จะใช้อัลกอริทึมย่อยภายในเป็น Decision tree เช่นเดียวกัน

หลังจากเขียนโค้ดคำสั่งตามตัวอย่างรูปข้างต้นแล้ว ทำการรันโค้ดคำสั่งทั้งหมด โดยจะแสดงผลลัพธ์ ยกตัวอย่างดังรูป



รูป 4.81 ตัวอย่างผลลัพธ์ในการรันโค้ดคำสั่งในการทำนายค่าด้วยอัลกอริทึม AdaBoost ใน R Data Mining กับชุดข้อมูลดอกไอริส (Iris.arff)

ซึ่งในการรันโค้ดคำสั่งอัลกอริทึม AdaBoost กับชุดข้อมูลอื่นๆ อาจมีการเปลี่ยนแปลงคำสั่งในบางส่วนตามความเหมาะสม ทำเช่นนี้กับทุกๆข้อมูลที่ต้องการทดลอง หาผลลัพธ์และนำค่าที่ได้ไปวิเคราะห์

4.5 เปรียบเทียบผลการทดลองการแบ่งชุดข้อมูลที่แตกต่างกัน

ในการเปรียบเทียบผลของการแบ่งชุดข้อมูลที่แตกต่างกัน โดยแบ่งเป็น

- แบ่งเป็นชุดข้อมูลเรียนรู้ 90% และชุดข้อมูลทดสอบ 10%
- แบ่งเป็นชุดข้อมูลเรียนรู้ 80% และชุดข้อมูลทดสอบ 20%

- แบ่งเป็นชุดข้อมูลเรียนรู้ 70% และชุดข้อมูลทดสอบ 30%

จะมีการนำชุดข้อมูล 2 ชุดข้อมูลได้แก่ ชุดข้อมูลดอกไอริส (Iris.arf) และชุดข้อมูลเบาหวาน (Diabetes.arff) ที่ได้จากการแบ่งแบบต่างๆมาทำการทดสอบด้วยอัลกอริทึมทั้งหมด 2 อัลกอริทึม ได้แก่ Naïve Bayes และDecision Tree และทำการเปรียบเทียบ ในแต่ละซอฟต์แวร์ แต่ละอัลกอริทึม

4.5.1 เปรียบเทียบการแบ่งชุดข้อมูลที่แตกต่างกันในซอฟต์แวร์เวก้า

ทำการทดลองโดยการเปลี่ยนชุดข้อมูลที่มีการนำเข้าเป็นชุดข้อมูลเรียนรู้ และชุดข้อมูลทดสอบที่สอดคล้องกัน

4.5.1.1 ดอกไอริส (Iris.arff)

แบ่งชุดข้อมูลดอกไอริสแบบแบ่งเป็นชุดข้อมูลเรียนรู้ 90% และชุดข้อมูลทดสอบ 10% ทดลองโดยใช้อัลกอริทึม NaiveBayes ทำการเปลี่ยนค่า Random Seed ทั้งหมด 10 ค่า ได้ผลลัพธ์ดังรูป

Correctly Classified Instances	14	93.3333	%
Incorrectly Classified Instances	1	6.6667	%
Correctly Classified Instances	14	93.3333	%
Incorrectly Classified Instances	1	6.6667	%
Correctly Classified Instances	15	100	%
Incorrectly Classified Instances	0	0	%
Correctly Classified Instances	15	100	%
Incorrectly Classified Instances	0	0	%
Correctly Classified Instances	14	93.3333	%
Incorrectly Classified Instances	1	6.6667	%
Correctly Classified Instances	15	100	%
Incorrectly Classified Instances	0	0	%
Correctly Classified Instances	15	100	%
Incorrectly Classified Instances	0	0	%
Correctly Classified Instances	14	93.3333	%
Incorrectly Classified Instances	1	6.6667	%
Correctly Classified Instances	15	100	%
Incorrectly Classified Instances	0	0	%
Correctly Classified Instances	15	100	%
Incorrectly Classified Instances	0	0	%

รูป 4.82 ผลลัพธ์การทำเหมืองข้อมูลด้วยชุดข้อมูลดอกไอริส มีการแบ่งแบบ Split 90% โดยใช้อัลกอริทึม NaïveBayes และทำการสุ่มทั้ง 10 ครั้ง ใน Weka ตามลำดับ

แบ่งชุดข้อมูลดอกไอริสแบบแบ่งเป็นชุดข้อมูลเรียนรู้ 80% และชุดข้อมูลทดสอบ 20% ทดลองโดยใช้อัลกอริทึม NaiveBayes ทำการเปลี่ยนค่า Random Seed ทั้งหมด 10 ค่า ได้ผลลัพธ์ดังรูป

```
Correctly Classified Instances          29               96.6667 %
Incorrectly Classified Instances         1                3.3333 %
Correctly Classified Instances          28               93.3333 %
Incorrectly Classified Instances         2                6.6667 %
Correctly Classified Instances          29               96.6667 %
Incorrectly Classified Instances         1                3.3333 %
Correctly Classified Instances          29               96.6667 %
Incorrectly Classified Instances         1                3.3333 %
Correctly Classified Instances          28               93.3333 %
Incorrectly Classified Instances         2                6.6667 %
Correctly Classified Instances          29               96.6667 %
Incorrectly Classified Instances         1                3.3333 %
Correctly Classified Instances          30              100      %
Incorrectly Classified Instances         0                0      %
Correctly Classified Instances          29               96.6667 %
Incorrectly Classified Instances         1                3.3333 %
Correctly Classified Instances          29               96.6667 %
Incorrectly Classified Instances         1                3.3333 %
Correctly Classified Instances          30              100      %
Incorrectly Classified Instances         0                0      %
```

รูป 4.83 ผลลัพธ์การทำเหมืองข้อมูลด้วยชุดข้อมูลดอกไอริส มีการแบ่งแบบ Split 80% โดยใช้อัลกอริทึม NaïveBayes และทำการสุ่มทั้ง 10 ครั้ง ใน Weka ตามลำดับ

แบ่งชุดข้อมูลดอกไอริสแบบแบ่งเป็นชุดข้อมูลเรียนรู้ 70% และชุดข้อมูลทดสอบ 30% ทดลองโดยใช้อัลกอริทึม NaiveBayes ทำการเปลี่ยนค่า Random Seed ทั้งหมด 10 ค่า ได้ผลลัพธ์ดังรูป

Correctly Classified Instances	43	95.5556 %
Incorrectly Classified Instances	2	4.4444 %
Correctly Classified Instances	42	93.3333 %
Incorrectly Classified Instances	3	6.6667 %
Correctly Classified Instances	43	95.5556 %
Incorrectly Classified Instances	2	4.4444 %
Correctly Classified Instances	44	97.7778 %
Incorrectly Classified Instances	1	2.2222 %
Correctly Classified Instances	41	91.1111 %
Incorrectly Classified Instances	4	8.8889 %
Correctly Classified Instances	44	97.7778 %
Incorrectly Classified Instances	1	2.2222 %
Correctly Classified Instances	44	97.7778 %
Incorrectly Classified Instances	1	2.2222 %
Correctly Classified Instances	44	97.7778 %
Incorrectly Classified Instances	1	2.2222 %
Correctly Classified Instances	43	95.5556 %
Incorrectly Classified Instances	2	4.4444 %
Correctly Classified Instances	44	97.7778 %
Incorrectly Classified Instances	1	2.2222 %

รูป 4.84 ผลลัพธ์การทำเหมืองข้อมูลด้วยชุดข้อมูลดอกไอริส มีการแบ่งแบบ Split 70% โดยใช้อัลกอริทึม NaïveBayes และทำการสุ่มทั้ง 10 ครั้ง ใน Weka ตามลำดับ

แบ่งชุดข้อมูลดอกไอริสแบบแบ่งเป็นชุดข้อมูลเรียนรู้ 90% และชุดข้อมูลทดสอบ 10% ทดลองโดยใช้อัลกอริทึม J48 ทำการเปลี่ยนค่า Random Seed ทั้งหมด 10 ค่า ได้ผลลัพธ์ดังรูป

Correctly Classified Instances	15	100 %
Incorrectly Classified Instances	0	0 %
Correctly Classified Instances	14	93.3333 %
Incorrectly Classified Instances	1	6.6667 %
Correctly Classified Instances	14	93.3333 %
Incorrectly Classified Instances	1	6.6667 %
Correctly Classified Instances	15	100 %
Incorrectly Classified Instances	0	0 %
Correctly Classified Instances	14	93.3333 %
Incorrectly Classified Instances	1	6.6667 %
Correctly Classified Instances	15	100 %
Incorrectly Classified Instances	0	0 %
Correctly Classified Instances	15	100 %
Incorrectly Classified Instances	0	0 %
Correctly Classified Instances	13	86.6667 %
Incorrectly Classified Instances	2	13.3333 %
Correctly Classified Instances	15	100 %
Incorrectly Classified Instances	0	0 %
Correctly Classified Instances	15	100 %
Incorrectly Classified Instances	0	0 %

รูป 4.85 ผลลัพธ์การทำเหมืองข้อมูลด้วยชุดข้อมูลดอกไอริส มีการแบ่งแบบ Split 90% โดยใช้อัลกอริทึม J48 และทำการสุ่มทั้ง 10 ครั้ง ใน Weka ตามลำดับ

แบ่งชุดข้อมูลดอกไอริสแบบแบ่งเป็นชุดข้อมูลเรียนรู้ 80% และชุดข้อมูลทดสอบ 20% ทดลองโดยใช้อัลกอริทึม J48 ทำการเปลี่ยนค่า Random Seed ทั้งหมด 10 ค่า ได้ผลลัพธ์ดังรูป

Correctly Classified Instances	30	100 %
Incorrectly Classified Instances	0	0 %
Correctly Classified Instances	29	96.6667 %
Incorrectly Classified Instances	1	3.3333 %
Correctly Classified Instances	29	96.6667 %
Incorrectly Classified Instances	1	3.3333 %
Correctly Classified Instances	28	93.3333 %
Incorrectly Classified Instances	2	6.6667 %
Correctly Classified Instances	28	93.3333 %
Incorrectly Classified Instances	2	6.6667 %
Correctly Classified Instances	29	96.6667 %
Incorrectly Classified Instances	1	3.3333 %
Correctly Classified Instances	30	100 %
Incorrectly Classified Instances	0	0 %
Correctly Classified Instances	28	93.3333 %
Incorrectly Classified Instances	2	6.6667 %
Correctly Classified Instances	30	100 %
Incorrectly Classified Instances	0	0 %
Correctly Classified Instances	29	96.6667 %
Incorrectly Classified Instances	1	3.3333 %

รูป 4.86 ผลลัพธ์การทำเหมืองข้อมูลด้วยชุดข้อมูลดอกไอริส มีการแบ่งแบบ Split 80% โดยใช้อัลกอริทึม J48 และทำการสุ่มทั้ง 10 ครั้ง ใน Weka ตามลำดับ

แบ่งชุดข้อมูลดอกไอริสแบบแบ่งเป็นชุดข้อมูลเรียนรู้ 70% และชุดข้อมูลทดสอบ 30% ทดลองโดยใช้อัลกอริทึม J48 ทำการเปลี่ยนค่า Random Seed ทั้งหมด 10 ค่า ได้ผลลัพธ์ดังรูป

```
Correctly Classified Instances          43               95.5556 %
Incorrectly Classified Instances         2                4.4444 %
Correctly Classified Instances          43               95.5556 %
Incorrectly Classified Instances         2                4.4444 %
Correctly Classified Instances          40               88.8889 %
Incorrectly Classified Instances         5               11.1111 %
Correctly Classified Instances          42               93.3333 %
Incorrectly Classified Instances         3                6.6667 %
Correctly Classified Instances          43               95.5556 %
Incorrectly Classified Instances         2                4.4444 %
Correctly Classified Instances          44               97.7778 %
Incorrectly Classified Instances         1                2.2222 %
Correctly Classified Instances          44               97.7778 %
Incorrectly Classified Instances         1                2.2222 %
Correctly Classified Instances          42               93.3333 %
Incorrectly Classified Instances         3                6.6667 %
Correctly Classified Instances          44               97.7778 %
Incorrectly Classified Instances         1                2.2222 %
Correctly Classified Instances          43               95.5556 %
Incorrectly Classified Instances         2                4.4444 %
```

รูป 4.87 ผลลัพธ์การทำเหมืองข้อมูลด้วยชุดข้อมูลดอกไอริส มีการแบ่งแบบ Split 70% โดยใช้อัลกอริทึม J48 และทำการสุ่มทั้ง 10 ครั้ง ใน Weka ตามลำดับ

4.5.1.2 เบาหวาน (Diabetes.arff)

แบ่งชุดข้อมูลเบาหวานแบบแบ่งเป็นชุดข้อมูลเรียนรู้ 90% และชุดข้อมูลทดสอบ 10% ทดลองโดยใช้อัลกอริทึม NaiveBayes ทำการเปลี่ยนค่า Random Seed ทั้งหมด 10 ค่า ได้ผลลัพธ์ดังรูป

```
Correctly Classified Instances          55               72.3684 %
Incorrectly Classified Instances        21               27.6316 %
Correctly Classified Instances          59               77.6316 %
Incorrectly Classified Instances        17               22.3684 %
Correctly Classified Instances          54               71.0526 %
Incorrectly Classified Instances        22               28.9474 %
Correctly Classified Instances          59               77.6316 %
Incorrectly Classified Instances        17               22.3684 %
Correctly Classified Instances          59               77.6316 %
Incorrectly Classified Instances        17               22.3684 %
Correctly Classified Instances          59               77.6316 %
Incorrectly Classified Instances        17               22.3684 %
Correctly Classified Instances          62               81.5789 %
Incorrectly Classified Instances        14               18.4211 %
Correctly Classified Instances          61               80.2632 %
Incorrectly Classified Instances        15               19.7368 %
Correctly Classified Instances          61               80.2632 %
Incorrectly Classified Instances        15               19.7368 %
```

รูป 4.88 ผลลัพธ์การทำเหมืองข้อมูลด้วยชุดข้อมูลเบาหวาน มีการแบ่งแบบ Split 90% โดยใช้อัลกอริทึม NaïveBayes และทำการสุ่มทั้ง 10 ครั้ง ใน Weka ตามลำดับ

แบ่งชุดข้อมูลเบาหวานแบบแบ่งเป็นชุดข้อมูลเรียนรู้ 80% และชุดข้อมูลทดสอบ 20% ทดลองโดยใช้อัลกอริทึม NaiveBayes ทำการเปลี่ยนค่า Random Seed ทั้งหมด 10 ค่า ได้ผลลัพธ์ดังรูป

```
Correctly Classified Instances         116               75.817  %
Incorrectly Classified Instances        37               24.183  %
Correctly Classified Instances         104               67.9739 %
Incorrectly Classified Instances        49               32.0261 %
Correctly Classified Instances         113               73.8562 %
Incorrectly Classified Instances        40               26.1438 %
Correctly Classified Instances         123               80.3922 %
Incorrectly Classified Instances        30               19.6078 %
Correctly Classified Instances         108               70.5882 %
Incorrectly Classified Instances        45               29.4118 %
Correctly Classified Instances         118               77.1242 %
Incorrectly Classified Instances        35               22.8758 %
Correctly Classified Instances         123               80.3922 %
Incorrectly Classified Instances        30               19.6078 %
Correctly Classified Instances         115               75.1634 %
Incorrectly Classified Instances        38               24.8366 %
Correctly Classified Instances         121               79.085  %
Incorrectly Classified Instances        32               20.915  %
Correctly Classified Instances         118               77.1242 %
Incorrectly Classified Instances        35               22.8758 %
```


รูป 4.89 ผลลัพธ์การทำเหมืองข้อมูลด้วยชุดข้อมูลเบาหวาน มีการแบ่งแบบ Split 80% โดยใช้อัลกอริทึม NaïveBayes และทำการสุ่มทั้ง 10 ครั้ง ใน Weka ตามลำดับ

แบ่งชุดข้อมูลเบาหวานแบบแบ่งเป็นชุดข้อมูลเรียนรู้ 70% และชุดข้อมูลทดสอบ 30% ทดลองโดยใช้อัลกอริทึม NaiveBayes ทำการเปลี่ยนค่า Random Seed ทั้งหมด 10 ค่า ได้ผลลัพธ์ดังรูป

```
Correctly Classified Instances         174               75.6522 %
Incorrectly Classified Instances        56               24.3478 %
Correctly Classified Instances         159               69.1304 %
Incorrectly Classified Instances        71               30.8696 %
Correctly Classified Instances         170               73.913  %
Incorrectly Classified Instances        60               26.087  %
Correctly Classified Instances         180               78.2609 %
Incorrectly Classified Instances        50               21.7391 %
Correctly Classified Instances         172               74.7826 %
Incorrectly Classified Instances        58               25.2174 %
Correctly Classified Instances         173               75.2174 %
Incorrectly Classified Instances        57               24.7826 %
Correctly Classified Instances         173               75.2174 %
Incorrectly Classified Instances        57               24.7826 %
Correctly Classified Instances         175               76.087  %
Incorrectly Classified Instances        55               23.913  %
Correctly Classified Instances         174               75.6522 %
Incorrectly Classified Instances        56               24.3478 %
Correctly Classified Instances         180               78.2609 %
Incorrectly Classified Instances        50               21.7391 %
```

รูป 4.90 ผลลัพธ์การทำเหมืองข้อมูลด้วยชุดข้อมูลเบาหวาน มีการแบ่งแบบ Split 70% โดยใช้อัลกอริทึม NaïveBayes และทำการสุ่มทั้ง 10 ครั้ง ใน Weka ตามลำดับ

แบ่งชุดข้อมูลเบาหวานแบบแบ่งเป็นชุดข้อมูลเรียนรู้ 90% และชุดข้อมูลทดสอบ 10% ทดลองโดยใช้อัลกอริทึม J48 ทำการเปลี่ยนค่า Random Seed ทั้งหมด 10 ค่า ได้ผลลัพธ์ดังรูป

Correctly Classified Instances	55	72.3684 %
Incorrectly Classified Instances	21	27.6316 %
Correctly Classified Instances	55	72.3684 %
Incorrectly Classified Instances	21	27.6316 %
Correctly Classified Instances	51	67.1053 %
Incorrectly Classified Instances	25	32.8947 %
Correctly Classified Instances	51	67.1053 %
Incorrectly Classified Instances	25	32.8947 %
Correctly Classified Instances	57	75 %
Incorrectly Classified Instances	19	25 %
Correctly Classified Instances	62	81.5789 %
Incorrectly Classified Instances	14	18.4211 %
Correctly Classified Instances	52	68.4211 %
Incorrectly Classified Instances	24	31.5789 %
Correctly Classified Instances	62	81.5789 %
Incorrectly Classified Instances	14	18.4211 %
Correctly Classified Instances	59	77.6316 %
Incorrectly Classified Instances	17	22.3684 %
Correctly Classified Instances	56	73.6842 %
Incorrectly Classified Instances	20	26.3158 %

รูป 4.91 ผลลัพธ์การทำเหมืองข้อมูลด้วยชุดข้อมูลเบาหวาน มีการแบ่งแบบ Split 90% โดยใช้อัลกอริทึม J48 และทำการสุ่มทั้ง 10 ครั้ง ใน Weka ตามลำดับ

แบ่งชุดข้อมูลเบาหวานแบบแบ่งเป็นชุดข้อมูลเรียนรู้ 80% และชุดข้อมูลทดสอบ 20% ทดลองโดยใช้อัลกอริทึม J48 ทำการเปลี่ยนค่า Random Seed ทั้งหมด 10 ค่า ได้ผลลัพธ์ดังรูป

Correctly Classified Instances	118	77.1242 %
Incorrectly Classified Instances	35	22.8758 %
Correctly Classified Instances	98	64.0523 %
Incorrectly Classified Instances	55	35.9477 %
Correctly Classified Instances	113	73.8562 %
Incorrectly Classified Instances	40	26.1438 %
Correctly Classified Instances	115	75.1634 %
Incorrectly Classified Instances	38	24.8366 %
Correctly Classified Instances	116	75.817 %
Incorrectly Classified Instances	37	24.183 %
Correctly Classified Instances	121	79.085 %
Incorrectly Classified Instances	32	20.915 %
Correctly Classified Instances	106	69.281 %
Incorrectly Classified Instances	47	30.719 %
Correctly Classified Instances	112	73.2026 %
Incorrectly Classified Instances	41	26.7974 %
Correctly Classified Instances	122	79.7386 %
Incorrectly Classified Instances	31	20.2614 %
Correctly Classified Instances	112	73.2026 %
Incorrectly Classified Instances	41	26.7974 %

รูป 4.92 ผลลัพธ์การทำเหมืองข้อมูลด้วยชุดข้อมูลเบาหวาน มีการแบ่งแบบ Split 80% โดยใช้อัลกอริทึม J48 และทำการสุ่มทั้ง 10 ครั้ง ใน Weka ตามลำดับ

แบ่งชุดข้อมูลเบาหวานแบบแบ่งเป็นชุดข้อมูลเรียนรู้ 70% และชุดข้อมูลทดสอบ 30% ทดลองโดยใช้อัลกอริทึม J48 ทำการเปลี่ยนค่า Random Seed ทั้งหมด 10 ค่า ได้ผลลัพธ์ดังรูป

Correctly Classified Instances	182	79.1304 %
Incorrectly Classified Instances	48	20.8696 %
Correctly Classified Instances	157	68.2609 %
Incorrectly Classified Instances	73	31.7391 %
Correctly Classified Instances	177	76.9565 %
Incorrectly Classified Instances	53	23.0435 %
Correctly Classified Instances	169	73.4783 %
Incorrectly Classified Instances	61	26.5217 %
Correctly Classified Instances	159	69.1304 %
Incorrectly Classified Instances	71	30.8696 %
Correctly Classified Instances	177	76.9565 %
Incorrectly Classified Instances	53	23.0435 %
Correctly Classified Instances	165	71.7391 %
Incorrectly Classified Instances	65	28.2609 %
Correctly Classified Instances	166	72.1739 %
Incorrectly Classified Instances	64	27.8261 %
Correctly Classified Instances	176	76.5217 %
Incorrectly Classified Instances	54	23.4783 %
Correctly Classified Instances	178	77.3913 %
Incorrectly Classified Instances	52	22.6087 %

รูป 4.93 ผลลัพธ์การทำเหมืองข้อมูลด้วยชุดข้อมูลเบาหวาน มีการแบ่งแบบ Split 70% โดยใช้อัลกอริทึม J48 และทำการสุ่มทั้ง 10 ครั้ง ใน Weka ตามลำดับ

นำผลลัพธ์ที่ได้มาหาค่าเฉลี่ยเพื่อวิเคราะห์ผลลัพธ์ต่อไป

4.5.2 เปรียบเทียบการแบ่งชุดข้อมูลที่แตกต่างกันในซอฟต์แวร์ราปิดมายเนอร์

ทำการทดลองโดยการเปลี่ยนชุดข้อมูลที่มีการนำเข้าเป็นชุดข้อมูลเรียนรู้ และชุดข้อมูลทดสอบที่สอดคล้องกัน

4.5.2.1 ดอกไอริส (Iris.arff)

แบ่งชุดข้อมูลดอกไอริสแบบแบ่งเป็นชุดข้อมูลเรียนรู้ 90% และชุดข้อมูลทดสอบ 10% ทดลองโดยใช้อัลกอริทึม Naïve Bayes ทำการเปลี่ยนค่า Random Seed ทั้งหมด 10 ค่า ได้ผลลัพธ์ดังรูป

accuracy: 93.33%	accuracy: 93.33%	accuracy: 100.00%	accuracy: 100.00%	accuracy: 100.00%
accuracy: 100.00%	accuracy: 100.00%	accuracy: 93.33%	accuracy: 100.00%	accuracy: 100.00%

รูป 4.94 ผลลัพธ์การทำเหมืองข้อมูลด้วยชุดข้อมูลดอกไอริส มีการแบ่งแบบ Split 90% โดยใช้อัลกอริทึม Naïve Bayes และทำการสุ่มทั้ง 10 ครั้ง ใน RapidMiner ตามลำดับ

แบ่งชุดข้อมูลดอกไอริสแบบแบ่งเป็นชุดข้อมูลเรียนรู้ 80% และชุดข้อมูลทดสอบ 20% ทดลองโดยใช้อัลกอริทึม Naïve Bayes ทำการเปลี่ยนค่า Random Seed ทั้งหมด 10 ค่า ได้ผลลัพธ์ดังรูป

accuracy: 96.67%	accuracy: 93.33%	accuracy: 100.00%	accuracy: 96.67%	accuracy: 93.33%
accuracy: 96.67%	accuracy: 100.00%	accuracy: 96.67%	accuracy: 96.67%	accuracy: 100.00%

รูป 4.95 ผลลัพธ์การทำเหมืองข้อมูลด้วยชุดข้อมูลดอกไอริส มีการแบ่งแบบ Split 80% โดยใช้อัลกอริทึม Naïve Bayes และทำการสุ่มทั้ง 10 ครั้ง ใน RapidMiner ตามลำดับ

แบ่งชุดข้อมูลดอกไอริสแบบแบ่งเป็นชุดข้อมูลเรียนรู้ 70% และชุดข้อมูลทดสอบ 30% ทดลองโดยใช้อัลกอริทึม Naïve Bayes ทำการเปลี่ยนค่า Random Seed ทั้งหมด 10 ค่า ได้ผลลัพธ์ดังรูป

accuracy: 95.56%	accuracy: 93.33%	accuracy: 97.78%	accuracy: 97.78%	accuracy: 95.56%
accuracy: 97.78%	accuracy: 97.78%	accuracy: 97.78%	accuracy: 95.56%	accuracy: 97.78%

รูป 4.96 ผลลัพธ์การทำเหมืองข้อมูลด้วยชุดข้อมูลดอกไอริส มีการแบ่งแบบ Split 70% โดยใช้อัลกอริทึม Naïve Bayes และทำการสุ่มทั้ง 10 ครั้ง ใน RapidMiner ตามลำดับ

แบ่งชุดข้อมูลดอกไอริสแบบแบ่งเป็นชุดข้อมูลเรียนรู้ 90% และชุดข้อมูลทดสอบ 10% ทดลองโดยใช้อัลกอริทึม Decision Tree ทำการเปลี่ยนค่า Random Seed ทั้งหมด 10 ค่า ได้ผลลัพธ์ดังรูป

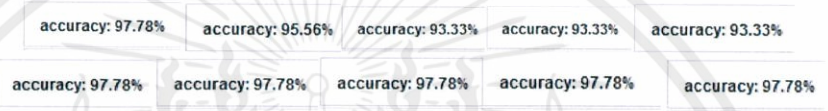
accuracy: 100.00%	accuracy: 93.33%	accuracy: 93.33%	accuracy: 100.00%	accuracy: 93.33%
accuracy: 100.00%	accuracy: 100.00%	accuracy: 93.33%	accuracy: 100.00%	accuracy: 100.00%

รูป 4.97 ผลลัพธ์การทำเหมืองข้อมูลด้วยชุดข้อมูลดอกไอริส มีการแบ่งแบบ Split 90% โดยใช้อัลกอริทึม Decision Tree และทำการสุ่มทั้ง 10 ครั้ง ใน RapidMiner ตามลำดับ

แบ่งชุดข้อมูลดอกไอริสแบบแบ่งเป็นชุดข้อมูลเรียนรู้ 80% และชุดข้อมูลทดสอบ 20% ทดลองโดยใช้อัลกอริทึม Decision Tree ทำการเปลี่ยนค่า Random Seed ทั้งหมด 10 ค่า ได้ผลลัพธ์ดังรูป

รูป 4.98 ผลลัพธ์การทำเหมืองข้อมูลด้วยชุดข้อมูลดอกไอริส มีการแบ่งแบบ Split 80% โดยใช้อัลกอริทึม Decision Tree และทำการสุ่มทั้ง 10 ครั้ง ใน RapidMiner ตามลำดับ

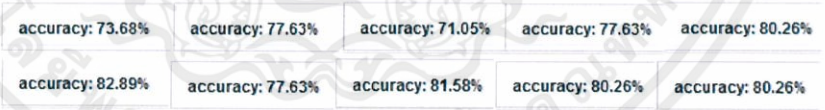
แบ่งชุดข้อมูลดอกไอริสแบบแบ่งเป็นชุดข้อมูลเรียนรู้ 70% และชุดข้อมูลทดสอบ 30% ทดลองโดยใช้อัลกอริทึม Decision Tree ทำการเปลี่ยนค่า Random Seed ทั้งหมด 10 ค่า ได้ผลลัพธ์ดังรูป



รูป 4.99 ผลลัพธ์การทำเหมืองข้อมูลด้วยชุดข้อมูลดอกไอริส มีการแบ่งแบบ Split 70% โดยใช้อัลกอริทึม Decision Tree และทำการสุ่มทั้ง 10 ครั้ง ใน RapidMiner ตามลำดับ

4.5.2.2 เบาหวาน (Diabetes.arff)

แบ่งชุดข้อมูลเบาหวานแบบแบ่งเป็นชุดข้อมูลเรียนรู้ 90% และชุดข้อมูลทดสอบ 10% ทดลองโดยใช้อัลกอริทึม Naïve Bayes ทำการเปลี่ยนค่า Random Seed ทั้งหมด 10 ค่า ได้ผลลัพธ์ดังรูป

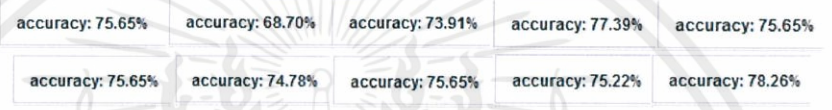


รูป 4.100 ผลลัพธ์การทำเหมืองข้อมูลด้วยชุดข้อมูลเบาหวาน มีการแบ่งแบบ Split 90% โดยใช้อัลกอริทึม Naïve Bayes และทำการสุ่มทั้ง 10 ครั้ง ใน RapidMiner ตามลำดับ

แบ่งชุดข้อมูลเบาหวานแบบแบ่งเป็นชุดข้อมูลเรียนรู้ 80% และชุดข้อมูลทดสอบ 20% ทดลองโดยใช้อัลกอริทึม Naïve Bayes ทำการเปลี่ยนค่า Random Seed ทั้งหมด 10 ค่า ได้ผลลัพธ์ดังรูป

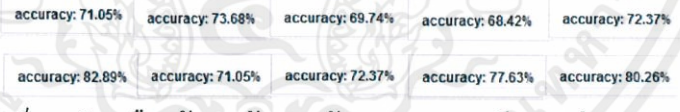
รูป 4.101 ผลลัพธ์การทำเหมืองข้อมูลด้วยชุดข้อมูลเบาหวาน มีการแบ่งแบบ Split 80% โดยใช้อัลกอริทึม Naïve Bayes และทำการสุ่มทั้ง 10 ครั้ง ใน RapidMiner ตามลำดับ

แบ่งชุดข้อมูลเบาหวานแบบแบ่งเป็นชุดข้อมูลเรียนรู้ 70% และชุดข้อมูลทดสอบ 30% ทดลองโดยใช้อัลกอริทึม Naïve Bayes ทำการเปลี่ยนค่า Random Seed ทั้งหมด 10 ค่า ได้ผลลัพธ์ดังรูป



รูป 4.102 ผลลัพธ์การทำเหมืองข้อมูลด้วยชุดข้อมูลเบาหวาน มีการแบ่งแบบ Split 70% โดยใช้อัลกอริทึม Naïve Bayes และทำการสุ่มทั้ง 10 ครั้ง ใน RapidMiner ตามลำดับ

แบ่งชุดข้อมูลเบาหวานแบบแบ่งเป็นชุดข้อมูลเรียนรู้ 90% และชุดข้อมูลทดสอบ 10% ทดลองโดยใช้อัลกอริทึม Decision Tree ทำการเปลี่ยนค่า Random Seed ทั้งหมด 10 ค่า ได้ผลลัพธ์ดังรูป



รูป 4.103 ผลลัพธ์การทำเหมืองข้อมูลด้วยชุดข้อมูลเบาหวาน มีการแบ่งแบบ Split 90% โดยใช้อัลกอริทึม Decision Tree และทำการสุ่มทั้ง 10 ครั้ง ใน RapidMiner ตามลำดับ

แบ่งชุดข้อมูลเบาหวานแบบแบ่งเป็นชุดข้อมูลเรียนรู้ 80% และชุดข้อมูลทดสอบ 20% ทดลองโดยใช้อัลกอริทึม Decision Tree ทำการเปลี่ยนค่า Random Seed ทั้งหมด 10 ค่า ได้ผลลัพธ์ดังรูป

รูป 4.104 ผลลัพธ์การทำเหมืองข้อมูลด้วยชุดข้อมูลเบาหวาน มีการแบ่งแบบ Split 80% โดยใช้อัลกอริทึม Decision Tree และทำการสุ่มทั้ง 10 ครั้ง ใน RapidMiner ตามลำดับ

แบ่งชุดข้อมูลเบาหวานแบบแบ่งเป็นชุดข้อมูลเรียนรู้ 70% และชุดข้อมูลทดสอบ 30% ทดลองโดยใช้อัลกอริทึม Decision Tree ทำการเปลี่ยนค่า Random Seed ทั้งหมด 10 ค่า ได้ผลลัพธ์ดังรูป

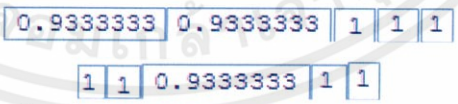
รูป 4.105 ผลลัพธ์การทำเหมืองข้อมูลด้วยชุดข้อมูลเบาหวาน มีการแบ่งแบบ Split 70% โดยใช้อัลกอริทึม Decision Tree และทำการสุ่มทั้ง 10 ครั้ง ใน RapidMiner ตามลำดับ

4.5.3 เปรียบเทียบการแบ่งชุดข้อมูลที่แตกต่างกันในซอฟต์แวร์อาร์ ดาต้ามายนิ่ง

ทำการทดลองโดยการเปลี่ยนชุดข้อมูลที่มีการนำเข้าเป็นชุดข้อมูลเรียนรู้ และชุดข้อมูลทดสอบที่สอดคล้องกัน

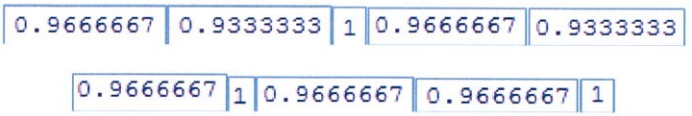
4.5.3.1 ดอกไอริส (Iris.arff)

แบ่งชุดข้อมูลดอกไอริสแบบแบ่งเป็นชุดข้อมูลเรียนรู้ 90% และชุดข้อมูลทดสอบ 10% ทดลองโดยใช้อัลกอริทึม Naïve Bayes ทำการเปลี่ยนค่า Random Seed ทั้งหมด 10 ค่า ได้ผลลัพธ์ดังรูป



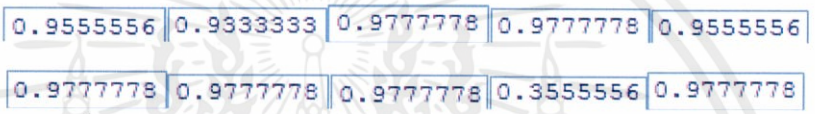
รูป 4.106 ผลลัพธ์การทำเหมืองข้อมูลด้วยชุดข้อมูลดอกไอริส มีการแบ่งแบบ Split 90% โดยใช้อัลกอริทึม Naïve Bayes และทำการสุ่มทั้ง 10 ครั้ง ใน R Data Mining ตามลำดับ

แบ่งชุดข้อมูลดอกไอริสแบบแบ่งเป็นชุดข้อมูลเรียนรู้ 80% และชุดข้อมูลทดสอบ 20% ทดลองโดยใช้อัลกอริทึม Naïve Bayes ทำการเปลี่ยนค่า Random Seed ทั้งหมด 10 ค่า ได้ผลลัพธ์ดังรูป



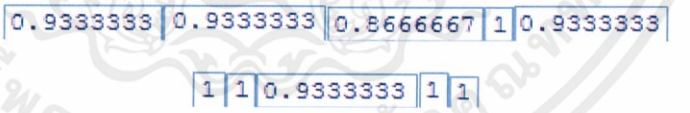
รูป 4.107 ผลลัพธ์การทำเหมืองข้อมูลด้วยชุดข้อมูลดอกไอริส มีการแบ่งแบบ Split 80% โดยใช้อัลกอริทึม Naïve Bayes และทำการสุ่มทั้ง 10 ครั้ง ใน R Data Mining ตามลำดับ

แบ่งชุดข้อมูลดอกไอริสแบบแบ่งเป็นชุดข้อมูลเรียนรู้ 70% และชุดข้อมูลทดสอบ 30% ทดลองโดยใช้อัลกอริทึม Naïve Bayes ทำการเปลี่ยนค่า Random Seed ทั้งหมด 10 ค่า ได้ผลลัพธ์ดังรูป



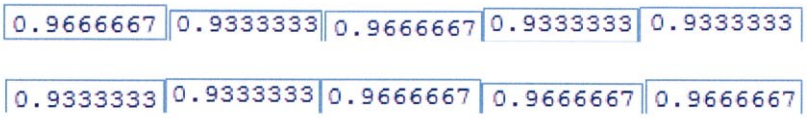
รูป 4.108 ผลลัพธ์การทำเหมืองข้อมูลด้วยชุดข้อมูลดอกไอริส มีการแบ่งแบบ Split 70% โดยใช้อัลกอริทึม Naïve Bayes และทำการสุ่มทั้ง 10 ครั้ง ใน R Data Mining ตามลำดับ

แบ่งชุดข้อมูลดอกไอริสแบบแบ่งเป็นชุดข้อมูลเรียนรู้ 90% และชุดข้อมูลทดสอบ 10% ทดลองโดยใช้อัลกอริทึม Decision Tree ทำการเปลี่ยนค่า Random Seed ทั้งหมด 10 ค่า ได้ผลลัพธ์ดังรูป



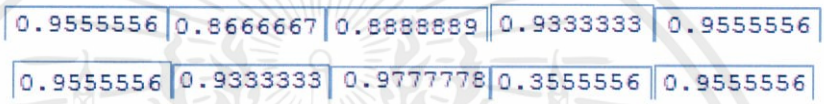
รูป 4.109 ผลลัพธ์การทำเหมืองข้อมูลด้วยชุดข้อมูลดอกไอริส มีการแบ่งแบบ Split 90% โดยใช้อัลกอริทึม Decision Tree และทำการสุ่มทั้ง 10 ครั้ง ใน R Data Mining ตามลำดับ

แบ่งชุดข้อมูลดอกไอริสแบบแบ่งเป็นชุดข้อมูลเรียนรู้ 80% และชุดข้อมูลทดสอบ 20% ทดลองโดยใช้อัลกอริทึม Decision Tree ทำการเปลี่ยนค่า Random Seed ทั้งหมด 10 ค่า ได้ผลลัพธ์ดังรูป



รูป 4.110 ผลลัพธ์การทำเหมืองข้อมูลด้วยชุดข้อมูลดอกไอริส มีการแบ่งแบบ Split 80% โดยใช้อัลกอริทึม Decision Tree และทำการสุ่มทั้ง 10 ครั้ง ใน R Data Mining ตามลำดับ

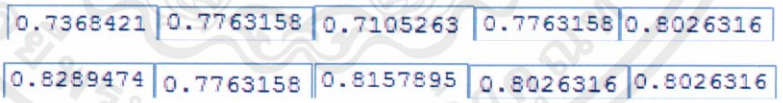
แบ่งชุดข้อมูลดอกไอริสแบบแบ่งเป็นชุดข้อมูลเรียนรู้ 70% และชุดข้อมูลทดสอบ 30% ทดลองโดยใช้อัลกอริทึม Decision Tree ทำการเปลี่ยนค่า Random Seed ทั้งหมด 10 ค่า ได้ผลลัพธ์ดังรูป



รูป 4.111 ผลลัพธ์การทำเหมืองข้อมูลด้วยชุดข้อมูลดอกไอริส มีการแบ่งแบบ Split 70% โดยใช้อัลกอริทึม Decision Tree และทำการสุ่มทั้ง 10 ครั้ง ใน R Data Mining ตามลำดับ

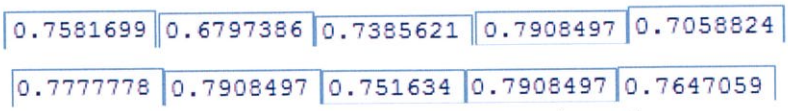
4.5.3.2 เบาหวาน (Diabetes.arff)

แบ่งชุดข้อมูลเบาหวานแบบแบ่งเป็นชุดข้อมูลเรียนรู้ 90% และชุดข้อมูลทดสอบ 10% ทดลองโดยใช้อัลกอริทึม Naïve Bayes ทำการเปลี่ยนค่า Random Seed ทั้งหมด 10 ค่า ได้ผลลัพธ์ดังรูป



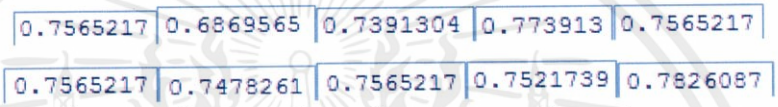
รูป 4.112 ผลลัพธ์การทำเหมืองข้อมูลด้วยชุดข้อมูลเบาหวาน มีการแบ่งแบบ Split 90% โดยใช้อัลกอริทึม Naïve Bayes และทำการสุ่มทั้ง 10 ครั้ง ใน R Data Mining ตามลำดับ

แบ่งชุดข้อมูลเบาหวานแบบแบ่งเป็นชุดข้อมูลเรียนรู้ 80% และชุดข้อมูลทดสอบ 20% ทดลองโดยใช้อัลกอริทึม Naïve Bayes ทำการเปลี่ยนค่า Random Seed ทั้งหมด 10 ค่า ได้ผลลัพธ์ดังรูป



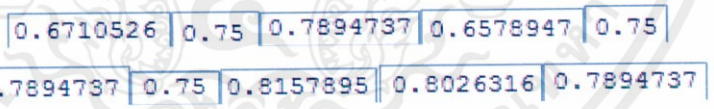
รูป 4.113 ผลลัพธ์การทำเหมืองข้อมูลด้วยชุดข้อมูลเบาหวาน มีการแบ่งแบบ Split 80% โดยใช้อัลกอริทึม Naïve Bayes และทำการสุ่มทั้ง 10 ครั้ง ใน R Data Mining ตามลำดับ

แบ่งชุดข้อมูลเบาหวานแบบแบ่งเป็นชุดข้อมูลเรียนรู้ 70% และชุดข้อมูลทดสอบ 30% ทดลองโดยใช้อัลกอริทึม Naïve Bayes ทำการเปลี่ยนค่า Random Seed ทั้งหมด 10 ค่า ได้ผลลัพธ์ดังรูป



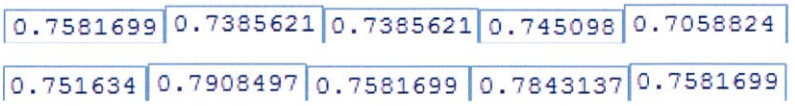
รูป 4.114 ผลลัพธ์การทำเหมืองข้อมูลด้วยชุดข้อมูลเบาหวาน มีการแบ่งแบบ Split 70% โดยใช้อัลกอริทึม Naïve Bayes และทำการสุ่มทั้ง 10 ครั้ง ใน R Data Mining ตามลำดับ

แบ่งชุดข้อมูลเบาหวานแบบแบ่งเป็นชุดข้อมูลเรียนรู้ 90% และชุดข้อมูลทดสอบ 10% ทดลองโดยใช้อัลกอริทึม Decision Tree ทำการเปลี่ยนค่า Random Seed ทั้งหมด 10 ค่า ได้ผลลัพธ์ดังรูป



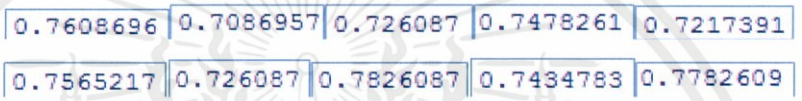
รูป 4.115 ผลลัพธ์การทำเหมืองข้อมูลด้วยชุดข้อมูลเบาหวาน มีการแบ่งแบบ Split 90% โดยใช้อัลกอริทึม Decision Tree และทำการสุ่มทั้ง 10 ครั้ง ใน R Data Mining ตามลำดับ

แบ่งชุดข้อมูลเบาหวานแบบแบ่งเป็นชุดข้อมูลเรียนรู้ 80% และชุดข้อมูลทดสอบ 20% ทดลองโดยใช้อัลกอริทึม Decision Tree ทำการเปลี่ยนค่า Random Seed ทั้งหมด 10 ค่า ได้ผลลัพธ์ดังรูป



รูป 4.116 ผลลัพธ์การทำเหมืองข้อมูลด้วยชุดข้อมูลเบาหวาน มีการแบ่งแบบ Split 80% โดยใช้อัลกอริทึม Decision Tree และทำการสุ่มทั้ง 10 ครั้ง ใน R Data Mining ตามลำดับ

แบ่งชุดข้อมูลเบาหวานแบบแบ่งเป็นชุดข้อมูลเรียนรู้ 70% และชุดข้อมูลทดสอบ 30% ทดลองโดยใช้อัลกอริทึม Decision Tree ทำการเปลี่ยนค่า Random Seed ทั้งหมด 10 ค่า ได้ผลลัพธ์ดังรูป



รูป 4.117 ผลลัพธ์การทำเหมืองข้อมูลด้วยชุดข้อมูลเบาหวาน มีการแบ่งแบบ Split 70% โดยใช้อัลกอริทึม Decision Tree และทำการสุ่มทั้ง 10 ครั้ง ใน R Data Mining ตามลำดับ

4.6 เปรียบเทียบอัลกอริทึมที่เหมาะสมกับชุดข้อมูลแต่ละประเภท

ในการทดลองจะใช้อัลกอริทึมกับแต่ละชุดข้อมูลทั้งหมด 6 อัลกอริทึม ดังนี้

- Rules>ZeroR
- Bayes>NaiveBayes
- Lazy>IBk
- Meta>AdaBoostM1
- Rules>OneR
- Trees>J48

โดยใช้รูปแบบการทดสอบแบบ Cross-validation

4.6.1 ดอกไอริส (Iris.arff)

เป็นข้อมูลเกี่ยวกับดอกไอริสประเภทต่างกัน 3 ประเภท ได้แก่ Iris-setosa, Iris-versicolor และ Iris-verginica ซึ่งจะประกอบไปด้วยข้อมูลแอตทริบิวต์ทั้งความยาวของกลีบเลี้ยงและกลีบดอก และความกว้างของกลีบเลี้ยงและกลีบดอก ซึ่งเราจะใช้ข้อมูลเหล่านี้ในการแบ่งหรือจำแนกดอกไอริสว่าเป็นดอกไอริสประเภทใด โดยข้อมูลแอตทริบิวต์ทั้งหมดประกอบด้วยตัวเลขจำนวนจริง มีจำนวนทั้งหมด 150 Instances

ในการทดลองเราจะทำการหาค่าเบสไลน์ก่อน จะได้ผลลัพธ์ดังรูป

Correctly Classified Instances	50	33.3333 %
Incorrectly Classified Instances	100	66.6667 %

รูป 4.118 ผลลัพธ์เปอร์เซ็นต์ความถูกต้องค่าเบสไลน์ของชุดข้อมูลดอกไอริส

ในขั้นตอนต่อไปเราจะทำการทดลองโดยทำการเรียนรู้ชุดข้อมูลดอกไอริสด้วยอัลกอริทึมอื่นๆ

อัลกอริทึมนาอีฟเบย์ (Naïve Bayes) ในกลุ่มอัลกอริทึมเบย์ (Bayes) ได้ผลลัพธ์ ดังรูป

Correctly Classified Instances	144	96 %
Incorrectly Classified Instances	6	4 %

รูป 4.119 ผลลัพธ์เปอร์เซ็นต์ความถูกต้องจากอัลกอริทึม Naive Bayes ของชุดข้อมูลดอกไอริส

จะพบว่าได้ค่าเปอร์เซ็นต์ความถูกต้อง 96% มากกว่าค่าเบสไลน์

อัลกอริทึมไอบีเค (IBk) ในกลุ่มอัลกอริทึมเลซี่ (Lazy) ได้ผลลัพธ์ ดังรูป

Correctly Classified Instances	143	95.3333 %
Incorrectly Classified Instances	7	4.6667 %

รูป 4.120 ผลลัพธ์เปอร์เซ็นต์ความถูกต้องจากอัลกอริทึม IBk ของชุดข้อมูลดอกไอริส

พบว่าเปอร์เซ็นต์ความถูกต้องได้ค่อนข้างสูงและน้อยกว่าเปอร์เซ็นต์ความถูกต้องที่ได้จากอัลกอริทึม Naive Bayes และ J48 เพียงเล็กน้อยเท่านั้น

อัลกอริทึมอดาบูสเอ็มวัน (AdaBoostM1) ในกลุ่มอัลกอริทึมเมต้า(Meta) ได้ผลลัพธ์ดังรูป

Correctly Classified Instances	140	93.3333 %
Incorrectly Classified Instances	10	6.6667 %

รูป 4.121 ผลลัพธ์เปอร์เซ็นต์ความถูกต้องจากอัลกอริทึม AdaBoostM1 ของชุดข้อมูลดอกไอริส

อัลกอริทึมวันอาร์ (OneR) ในกลุ่มอัลกอริทึมรูล (Rule) ได้ผลลัพธ์ดังรูป

Correctly Classified Instances	138	92 %
Incorrectly Classified Instances	12	8 %

รูป 4.122 ผลลัพธ์เปอร์เซ็นต์ความถูกต้องจากอัลกอริทึม OneR ของชุดข้อมูลดอกไอริส

อัลกอริทึมเจสี่สิบแปด (J48) ในกลุ่มอัลกอริทึมทรี (Tree) ได้ผลลัพธ์ดังรูป

Correctly Classified Instances	144	96	%
Incorrectly Classified Instances	6	4	%

รูป 4.123 ผลลัพธ์เปอร์เซ็นต์ความถูกต้องจากอัลกอริทึม J48 ของชุดข้อมูลดอกไอริส

จะพบว่าได้ค่าเปอร์เซ็นต์ความถูกต้อง 96% มากกว่าค่าเบสไลน์ และอัตราส่วนของจำนวน Instancesที่ทำการจำแนกได้ถูกต้องมากกว่าจำนวน Instances ที่จำแนกผิดเป็นจำนวนมาก

4.6.2 กระจก (Glass.arff)

เป็นข้อมูลเกี่ยวกับองค์ประกอบทางเคมีต่างๆ ที่ใช้ในการทำกระจกประเภทต่างๆ กัน โดยใช้ข้อมูลองค์ประกอบทางเคมีที่เรามีเพื่อทำนายวัสดุชิ้นนี้เป็นกระจกประเภทใด ประกอบด้วย แอตทริบิวต์ทั้งหมด 9 แอตทริบิวต์ ประกอบด้วยค่าการหักเห ปริมาณโซเดียม แมกนีเซียม อลูมิเนียม ซิลิกอน โพแทสเซียม แคลเซียม แบเรียม และเหล็ก โดยแอตทริบิวต์ทั้งหมดจะเป็นข้อมูลประเภท Numeric มีทั้งหมด 214 Instances

หาค่าเบสไลน์ด้วยอัลกอริทึมซีโร่อาร์ (ZeroR)

Correctly Classified Instances	76	35.514	%
Incorrectly Classified Instances	138	64.486	%

รูป 4.124 ผลลัพธ์เปอร์เซ็นต์ความถูกต้องค่าเบสไลน์ของชุดข้อมูลกระจก

จะพบว่าได้เปอร์เซ็นต์ความถูกต้อง 35.51% ซึ่งถือได้ว่าน้อยมากๆ เนื่องจากอัลกอริทึม ZeroR จะทำการทำนายค่า Instances ใหม่ทั้งหมดตามประเภทของจำนวน Instances เดิมที่มีจำนวนเยอะที่สุด แต่ชุดข้อมูลกระจกมีประเภทของแอตทริบิวต์ที่เป็นคลาสจำนวนมาก เมื่อใช้อัลกอริทึม ZeroR ทำให้เมื่อทำการทำนายข้อมูลจึงทำนายเป็นประเภทเดียวเท่านั้นที่ถูกต้อง ประเภทอื่นๆจะทำนายผิดหมดเลย ค่าความถูกต้องที่ได้จึงต่ำมากๆ

ในขั้นตอนต่อไปเราจะทำการทดลองโดยทำการเรียนรู้ชุดข้อมูลกระจกด้วยอัลกอริทึมอื่นๆ

อัลกอริทึมนาอีฟเบย์ (Naïve Bayes) ในกลุ่มอัลกอริทึมเบย์ (Bayes) ได้ผลลัพธ์ ดังรูป

Correctly Classified Instances	104	48.5981 %
Incorrectly Classified Instances	110	51.4019 %

รูป 4.125 ผลลัพธ์เปอร์เซ็นต์ความถูกต้องจากอัลกอริทึม Naïve Bayes ของชุดข้อมูลกระจก

พบว่ามีเปอร์เซ็นต์ความถูกต้องเพียง 48.60% เท่านั้น แม้จะมากกว่าค่าเบสไลน์แต่ค่าเปอร์เซ็นต์ความถูกต้องยังน้อยไปไม่เหมาะกับการนำมาใช้กับชุดข้อมูลนี้

อัลกอริทึมไอบีเค (IBk) ในกลุ่มอัลกอริทึมเลซี่ (Lazy) ได้ผลลัพธ์ ดังรูป

Correctly Classified Instances	151	70.5607 %
Incorrectly Classified Instances	63	29.4393 %

รูป 4.126 ผลลัพธ์เปอร์เซ็นต์ความถูกต้องจากอัลกอริทึม IBk ของชุดข้อมูลกระจก

พบว่าอัลกอริทึมนี้จะให้เปอร์เซ็นต์ความถูกต้องถึง 70.56% ซึ่งจะมากกว่าอัลกอริทึมอื่นๆ ค่อนข้างมาก

อัลกอริทึมอดาบูสเอ็มวัน (AdaBoostM1) ในกลุ่มอัลกอริทึมเมต้า (Meta) ได้ผลลัพธ์ดังรูป

Correctly Classified Instances	159	74.2991 %
Incorrectly Classified Instances	55	25.7009 %

รูป 4.127 ผลลัพธ์เปอร์เซ็นต์ความถูกต้องจากอัลกอริทึม AdaBoostM1 ของชุดข้อมูลกระจก

อัลกอริทึมวันอาร์ (OneR) ในกลุ่มอัลกอริทึมรูล (Rule) ได้ผลลัพธ์ดังรูป

Correctly Classified Instances	124	57.9439 %
Incorrectly Classified Instances	90	42.0561 %

รูป 4.128 ผลลัพธ์เปอร์เซ็นต์ความถูกต้องจากอัลกอริทึม OneR ของชุดข้อมูลกระจก

อัลกอริทึมเจสี่สิบแปด (J48) ในกลุ่มอัลกอริทึมทรี (Tree) ได้ผลลัพธ์ดังรูป

Correctly Classified Instances	143	66.8224 %
Incorrectly Classified Instances	71	33.1776 %

รูป 4.129 ผลลัพธ์เปอร์เซ็นต์ความถูกต้องจากอัลกอริทึม J48 ของชุดข้อมูลกระจก

จะพบว่าเมื่อใช้อัลกอริทึมJ48 กับชุดข้อมูลกระจกจะได้เปอร์เซ็นต์ความถูกต้อง 66.82% ซึ่งมากกว่าค่าเบสไลน์ แต่อย่างไรก็ตามจะเห็นว่าจำนวน Instances ที่มีการจำแนกผิดก็มีจำนวนมากเช่นกัน

4.6.3 มะเร็งเต้านม (Breast-cancer.arff)

เป็นข้อมูลเกี่ยวกับข้อมูลทางการแพทย์ ประกอบไปด้วยแอตทริบิวต์ต่างๆที่เกี่ยวข้องกับสุขภาพของคนไข้ทั้งหมด 9 แอตทริบิวต์ เพื่อการทำนายว่าคนไข้นั้นจะมีโอกาสเป็นมะเร็งเต้านมหรือไม่ ซึ่งแอตทริบิวต์ทั้งหมดจะเป็นข้อมูลประเภท Nominal หรือข้อมูลที่เป็นคำ (word) มีจำนวนทั้งหมด 286 Instances

หาค่าเบสไลน์ด้วยอัลกอริทึมซีโร่อาร์ (ZeroR)

Correctly Classified Instances	201	70.2797 %
Incorrectly Classified Instances	85	29.7203 %

รูป 4.130 ผลลัพธ์เปอร์เซ็นต์ความถูกต้องค่าเบสไลน์ของชุดข้อมูลมะเร็งเต้านม

จะพบว่าจะได้ค่าเบสไลน์ 70.28% ซึ่งถือว่าเป็นค่าเบสไลน์ที่ค่อนข้างสูง

ในขั้นตอนต่อไปเราจะทำการทดลองโดยทำการเรียนรู้ชุดข้อมูลมะเร็งเต้านมด้วยอัลกอริทึมอื่นๆ

อัลกอริทึมนาอีฟเบย์ (Naïve Bayes) ในกลุ่มอัลกอริทึมเบย์ (Bayes) ได้ผลลัพธ์ ดังรูป

Correctly Classified Instances	205	71.6783 %
Incorrectly Classified Instances	81	28.3217 %

รูป 4.131 ผลลัพธ์เปอร์เซ็นต์ความถูกต้องจากอัลกอริทึม Naive Bayes ของชุดข้อมูลมะเร็งเต้านม

พบว่ามีเปอร์เซ็นต์ความถูกต้อง 71.68% มากกว่าค่าเบสไลน์แต่ยังน้อยกว่า J48

อัลกอริทึมไอบีเค (IBk) ในกลุ่มอัลกอริทึมเลซี่ (Lazy) ได้ผลลัพธ์ ดังรูป

Correctly Classified Instances	207	72.3776 %
Incorrectly Classified Instances	79	27.6224 %

รูป 4.132 ผลลัพธ์เปอร์เซ็นต์ความถูกต้องจากอัลกอริทึม IBk ของชุดข้อมูลมะเร็งเต้านม

อัลกอริทึมอดาบูสเอ็มวัน (AdaBoostM1) ในกลุ่มอัลกอริทึมเมต้า (Meta) ได้ผลลัพธ์ ดังรูป

Correctly Classified Instances	199	69.5804 %
Incorrectly Classified Instances	87	30.4196 %

รูป 4.133 ผลลัพธ์เปอร์เซ็นต์ความถูกต้องจากอัลกอริทึม AdaBoostM1 ของชุดข้อมูลมะเร็งเต้านม

อัลกอริทึมวันอาร์ (OneR) ในกลุ่มอัลกอริทึมรูล (Rule) ได้ผลลัพธ์ดังรูป

Correctly Classified Instances	188	65.7343 %
Incorrectly Classified Instances	98	34.2657 %

รูป 4.134 ผลลัพธ์เปอร์เซ็นต์ความถูกต้องจากอัลกอริทึม OneR ของชุดข้อมูลมะเร็งเต้านม

จะพบว่าคาเปอร์เซ็นต์ความถูกต้องที่ได้มีค่าน้อยกว่าค่าเบสไลน์ ดังนั้นอัลกอริทึม OneR จึงไม่เหมาะสมจะนำมาใช้กับชุดข้อมูลมะเร็งเต้านม

อัลกอริทึมเจสี่สิบแปด (J48) ในกลุ่มอัลกอริทึมทรี (Tree) ได้ผลลัพธ์ดังรูป

Correctly Classified Instances	216	75.5245 %
Incorrectly Classified Instances	70	24.4755 %

รูป 4.135 ผลลัพธ์เปอร์เซ็นต์ความถูกต้องจากอัลกอริทึม J48 ของชุดข้อมูลมะเร็งเต้านม

พบว่ามีค่าเปอร์เซ็นต์ความถูกต้อง 75.52% มากกว่าค่าเบสไลน์ไม่มากเท่าใดนัก

4.6.4 เบาหวาน (Diabetes.arff)

เป็นข้อมูลทางการแพทย์เกี่ยวกับข้อมูลสุขภาพโดยทั่วๆไปของคนไข้ ประกอบด้วยแอตทริบิวต์ทั้งหมดจำนวน 8 แอตทริบิวต์ อันประกอบไปด้วย จำนวนครั้งของการตั้งครรภ์ ความเข้มข้นของกลูโคสพลาสม่า ความดันเลือด ความหนาของผิวหนัง ระดับอิซูลิน ดัชนีมวลกาย ความน่าจะเป็นจากเครือญาติ และอายุ ซึ่งจะนำแอตทริบิวต์เหล่านี้มาทำการเรียนรู้เพื่อสร้างโมเดลทำนายว่าคนไข้มีโอกาสเป็นโรคเบาหวานหรือไม่ แอตทริบิวต์เหล่านี้มีประเภทข้อมูลเป็น numeric ทั้งหมด ประกอบด้วยInstancesทั้งหมด 768 Instances

หาค่าเบสไลน์ด้วยอัลกอริทึมซีโร่อาร์ (ZeroR)

Correctly Classified Instances	500	65.1042 %
Incorrectly Classified Instances	268	34.8958 %

รูป 4.136 ผลลัพธ์เปอร์เซ็นต์ความถูกต้องค่าเบสไลน์ของชุดข้อมูลเบาหวาน

พบว่าจะมีค่าเบสไลน์เท่ากับ 65.10%

ในขั้นตอนต่อไปเราจะทำการทดลองโดยทำการเรียนรู้ชุดข้อมูลเบาหวานด้วยอัลกอริทึมอื่นๆ

อัลกอริทึมนาอีฟเบย์ (Naïve Bayes) ในกลุ่มอัลกอริทึมเบย์ (Bayes) ได้ผลลัพธ์ ดังรูป

Correctly Classified Instances	586	76.3021 %
Incorrectly Classified Instances	182	23.6979 %

รูป 4.137 ผลลัพธ์เปอร์เซ็นต์ความถูกต้องจากอัลกอริทึม Naive Bayes ของชุดข้อมูลเบาหวาน

พบว่ามีเปอร์เซ็นต์ความถูกต้อง 76.30% มากกว่าค่าเบสไลน์และค่าเปอร์เซ็นต์ความถูกต้องที่ได้จากอัลกอริทึม J48

อัลกอริทึมไอบีเค (IBk) ในกลุ่มอัลกอริทึมเลซี่ (Lazy) ได้ผลลัพธ์ ดังรูป

Correctly Classified Instances	539	70.1823 %
Incorrectly Classified Instances	229	29.8177 %

รูป 4.138 ผลลัพธ์เปอร์เซ็นต์ความถูกต้องจากอัลกอริทึม IBk ของชุดข้อมูลเบาหวาน

พบว่ามีเปอร์เซ็นต์ความถูกต้องเพียง 70.18% ซึ่งถือว่ามากกว่าค่าเบสไลน์สามารถนำมาใช้งานได้ แต่ยังมีค่าเปอร์เซ็นต์ที่ทำนายผิดพลาดอยู่มาก

อัลกอริทึมอดาบูสเอ็มวัน (AdaBoostM1) ในกลุ่มอัลกอริทึมเมต้า(Meta) ได้ผลลัพธ์ดังรูป

Correctly Classified Instances	556	72.3958 %
Incorrectly Classified Instances	212	27.6042 %

รูป 4.139 ผลลัพธ์เปอร์เซ็นต์ความถูกต้องจากอัลกอริทึม AdaBoostM1 ของชุดข้อมูลเบาหวาน

อัลกอริทึมวันอาร์ (OneR) ในกลุ่มอัลกอริทึมรูล (Rule) ได้ผลลัพธ์ดังรูป

Correctly Classified Instances	549	71.4844 %
Incorrectly Classified Instances	219	28.5156 %

รูป 4.140 ผลลัพธ์เปอร์เซ็นต์ความถูกต้องจากอัลกอริทึม OneR ของชุดข้อมูลเบาหวาน

อัลกอริทึมเจสี่สิบแปด (J48) ในกลุ่มอัลกอริทึมทรี (Tree) ได้ผลลัพธ์ดังรูป

Correctly Classified Instances	567	73.8281 %
Incorrectly Classified Instances	201	26.1719 %

รูป 4.141 ผลลัพธ์เปอร์เซ็นต์ความถูกต้องจากอัลกอริทึม J48 ของชุดข้อมูลเบาหวาน

จะพบว่ามีค่าเปอร์เซ็นต์ความถูกต้อง 73.83% มากกว่าค่าเบสไลน์ค่อนข้างมาก เมื่อดูจำนวน Instances ที่ทำนายถูกต้องจะพบว่ามีค่าที่ทำนายถูกต้องเพิ่มมากขึ้นจากเบสไลน์

4.6.5 ถั่วเหลือง (Soybean.arff)

เป็นข้อมูลเกี่ยวกับการเป็นโรคต่างๆในต้นถั่วเหลือง ซึ่งจะมีแอตทริบิวต์ที่ประกอบไปด้วยสภาพแวดล้อมต่างๆเช่น ฝน อุณหภูมิ ลูกเห็บ และประกอบไปด้วยลักษณะของลำต้น และใบเพื่อใช้ในการจำแนกว่าแต่ละต้นมีโอกาสเป็นโรคใด ทั้งหมด 35 แอตทริบิวต์ โดยเป็นข้อมูลประเภท Nominal ทั้งหมด ข้อมูลถั่วเหลืองประกอบไปด้วยInstancesทั้งหมดจำนวน 683 Instances

หาค่าเบสไลน์ด้วยอัลกอริทึมซีโร่อาร์ (ZeroR)

Correctly Classified Instances	92	13.47 %
Incorrectly Classified Instances	591	86.53 %

รูป 4.142 ผลลัพธ์เปอร์เซ็นต์ความถูกต้องค่าเบสไลน์ของชุดข้อมูลถั่วเหลือง

จะเห็นได้ว่าเนื่องจากชุดข้อมูลถั่วเหลืองมีค่าของแอตทริบิวต์คลาสเป็นจำนวนมากถึง 19 แอตทริบิวต์ ทำให้เมื่อใช้อัลกอริทึม ZeroR ทำให้ได้เปอร์เซ็นต์ความถูกต้องที่ค่อนข้างต่ำมากๆ สำหรับชุดข้อมูลถั่วเหลืองจึงได้เปอร์เซ็นต์ความถูกต้องเพียง 13.47%

ในขั้นตอนต่อไปเราจะทำการทดลองโดยทำการเรียนรู้ชุดข้อมูลถั่วเหลืองด้วยอัลกอริทึมอื่นๆ

อัลกอริทึมนาอีฟเบย์ (Naïve Bayes) ในกลุ่มอัลกอริทึมเบย์ (Bayes) ได้ผลลัพธ์ ดังรูป

Correctly Classified Instances	635	92.9722 %
Incorrectly Classified Instances	48	7.0278 %

รูป 4.143 ผลลัพธ์เปอร์เซ็นต์ความถูกต้องจากอัลกอริทึม Naïve Bayes ของชุดข้อมูลถั่วเหลือง

ผลลัพธ์ที่ได้มีเปอร์เซ็นต์ความถูกต้องถึง 92.97% มากกว่าอัลกอริทึมJ48เพียงเล็กน้อย

อัลกอริทึมไอบีเค (IBk) ในกลุ่มอัลกอริทึมเลซี่ (Lazy) ได้ผลลัพธ์ ดังรูป

Correctly Classified Instances	623	91.2152 %
Incorrectly Classified Instances	60	8.7848 %

รูป 4.144 ผลลัพธ์เปอร์เซ็นต์ความถูกต้องจากอัลกอริทึม IBk ของชุดข้อมูลถั่วเหลือง

อัลกอริทึมอดาบูสเอ็มวัน(AdaBoostM1) ในกลุ่มอัลกอริทึมเมต้า(Meta) ได้ผลลัพธ์ ดังรูป

Correctly Classified Instances	634	92.8258 %
Incorrectly Classified Instances	49	7.1742 %

รูป 4.145 ผลลัพธ์เปอร์เซ็นต์ความถูกต้องจากอัลกอริทึม AdaBoostM1 ของชุดข้อมูลถั่วเหลือง

อัลกอริทึมวันอาร์ (OneR) ในกลุ่มอัลกอริทึมรูล (Rule) ได้ผลลัพธ์ดังรูป

Correctly Classified Instances	273	39.9707 %
Incorrectly Classified Instances	410	60.0293 %

รูป 4.146 ผลลัพธ์เปอร์เซ็นต์ความถูกต้องจากอัลกอริทึม OneR ของชุดข้อมูลถั่วเหลือง

ผลลัพธ์ที่ได้จากการใช้อัลกอริทึม OneR กับชุดข้อมูลถั่วเหลืองพบว่าเปอร์เซ็นต์ความถูกต้องที่ได้แม้จะมากกว่าค่าเบสไลน์ แต่ความถูกต้องยังน้อยอยู่มากๆ เพียง 39.97%

อัลกอริทึมเจสี่สิบแปด (J48) ในกลุ่มอัลกอริทึมทรี (Tree) ได้ผลลัพธ์ดังรูป

Correctly Classified Instances	625	91.5081 %
Incorrectly Classified Instances	58	8.4919 %

รูป 4.147 ผลลัพธ์เปอร์เซ็นต์ความถูกต้องจากอัลกอริทึม J48 ของชุดข้อมูลถั่วเหลือง

จะพบว่าเปอร์เซ็นต์ความถูกต้องที่ได้มีค่าค่อนข้างสูงมากถึง 91.51%

4.6.6 การลงคะแนนเสียงเลือกตั้ง (Vote.arff)

เป็นข้อมูลเกี่ยวกับการเลือกตั้งของประเทศอเมริกาสองพรรคที่แข่งขันกัน โดยจะมีการเก็บข้อมูลจากประชาชนว่าชอบนโยบายใด ไม่ชอบนโยบายใดบ้าง และสุดท้ายประชาชนเลือกพรรคใด ประกอบด้วยแอตทริบิวต์ทั้งหมดจำนวน 16 แอตทริบิวต์ โดยแต่ละแอตทริบิวต์จะเป็นข้อมูลประเภท nominal เพื่อการทำนายว่าประชาชนจะเลือกพรรคใด มีทั้งหมด 435 Instances

หาค่าเบสไลน์ด้วยอัลกอริทึมซีโร่อาร์ (ZeroR)

Correctly Classified Instances	267	61.3793 %
Incorrectly Classified Instances	168	38.6207 %

รูป 4.148 ผลลัพธ์เปอร์เซ็นต์ความถูกต้องค่าเบสไลน์ของชุดข้อมูลการลงคะแนนเสียงเลือกตั้ง

ในขั้นตอนต่อไปเราจะทำการทดลองโดยทำการเรียนรู้ชุดข้อมูลการลงคะแนนเสียงเลือกตั้งด้วยอัลกอริทึมอื่นๆ

อัลกอริทึมนาอีฟเบย์ (Naïve Bayes) ในกลุ่มอัลกอริทึมเบย์ (Bayes) ได้ผลลัพธ์ ดังรูป

Correctly Classified Instances	392	90.1149 %
Incorrectly Classified Instances	43	9.8851 %

รูป 4.149 ผลลัพธ์เปอร์เซ็นต์ความถูกต้องจากอัลกอริทึม Naive Bayes ของชุดข้อมูลการลงคะแนนเสียงเลือกตั้ง

พบว่ามีค่าเปอร์เซ็นต์ความถูกต้อง 90.11% สูงกว่าค่าเบสไลน์ค่อนข้างมาก

อัลกอริทึมไอบีเค (IBk) ในกลุ่มอัลกอริทึมเลซี่ (Lazy) ได้ผลลัพธ์ ดังรูป

Correctly Classified Instances	402	92.4138 %
Incorrectly Classified Instances	33	7.5862 %

รูป 4.150 ผลลัพธ์เปอร์เซ็นต์ความถูกต้องจากอัลกอริทึม IBk ของชุดข้อมูลการลงคะแนนเสียงเลือกตั้ง

พบว่ามีค่าเปอร์เซ็นต์ความถูกต้องสูงถึง 92.41%

อัลกอริทึมอดาบูสเอ็มวัน (AdaBoostM1) ในกลุ่มอัลกอริทึมเมต้า (Meta) ได้ผลลัพธ์ดังรูป

Correctly Classified Instances	417	95.8621 %
Incorrectly Classified Instances	18	4.1379 %

รูป 4.151 ผลลัพธ์เปอร์เซ็นต์ความถูกต้องจากอัลกอริทึม AdaBoostM1 ของชุดข้อมูลการลงคะแนนเสียงเลือกตั้ง

อัลกอริทึมวันอาร์ (OneR) ในกลุ่มอัลกอริทึมรูล (Rule) ได้ผลลัพธ์ดังรูป

Correctly Classified Instances	416	95.6322 %
Incorrectly Classified Instances	19	4.3678 %

รูป 4.152 ผลลัพธ์เปอร์เซ็นต์ความถูกต้องจากอัลกอริทึม OneR ของชุดข้อมูลการลงคะแนนเสียงเลือกตั้ง

อัลกอริทึมเจสี่สิบแปด (J48) ในกลุ่มอัลกอริทึมทรี (Tree) ได้ผลลัพธ์ดังรูป

Correctly Classified Instances	419	96.3218 %
Incorrectly Classified Instances	16	3.6782 %

รูป 4.153 ผลลัพธ์เปอร์เซ็นต์ความถูกต้องจากอัลกอริทึม J48 ของชุดข้อมูลการลงคะแนนเสียงเลือกตั้ง

จะพบได้ว่าหากใช้อัลกอริทึม J48 กับชุดข้อมูลการลงคะแนนเสียงเลือกตั้งแล้วทำให้ค่าเปอร์เซ็นต์ความถูกต้องสูงถึง 96.32%

4.6.7 สินเชื่อ (Credit-g.arff)

เป็นข้อมูลเกี่ยวกับข้อมูลทั่วไปต่างๆของลูกค้า และข้อมูลการทำธุรกรรมในอดีตของลูกค้า เพื่อจำแนกว่าเป็นลูกค้าที่ดีขององค์กรหรือไม่ ประกอบด้วยแอตทริบิวต์ต่างๆ 20 แอตทริบิวต์ มีทั้งข้อมูลประเภท Nominal และ Numeric ประกอบด้วย Instances ทั้งหมด 1000 Instances

หาค่าเบสไลน์ด้วยอัลกอริทึมซีโร่อาร์ (ZeroR)

Correctly Classified Instances	700	70	%
Incorrectly Classified Instances	300	30	%

รูป 4.154 ผลลัพธ์เปอร์เซ็นต์ความถูกต้องค่าเบสไลน์ของชุดข้อมูลสินเชื่อ

พบว่ามีค่าเบสไลน์อยู่ที่ 70%

ในขั้นตอนต่อไปเราจะทำการทดลองโดยทำการเรียนรู้ชุดข้อมูลสินเชื่อด้วยอัลกอริทึมอื่นๆ

อัลกอริทึมนาอีฟเบย์ (Naïve Bayes) ในกลุ่มอัลกอริทึมเบย์ (Bayes) ได้ผลลัพธ์ ดังรูป

Correctly Classified Instances	754	75.4	%
Incorrectly Classified Instances	246	24.6	%

รูป 4.155 ผลลัพธ์เปอร์เซ็นต์ความถูกต้องจากอัลกอริทึม Naive Bayes ของชุดข้อมูลสินเชื่อ

พบว่ามีค่าเปอร์เซ็นต์ความถูกต้องสูงกว่าอัลกอริทึม J48 โดยมีค่าเปอร์เซ็นต์ความถูกต้องอยู่ที่ 75.4%

อัลกอริทึมไอบีเค (IBk) ในกลุ่มอัลกอริทึมเลซี่ (Lazy) ได้ผลลัพธ์ ดังรูป

Correctly Classified Instances	720	72	%
Incorrectly Classified Instances	280	28	%

รูป 4.156 ผลลัพธ์เปอร์เซ็นต์ความถูกต้องจากอัลกอริทึม IBk ของชุดข้อมูลสินเชื่อ

จะเห็นได้ว่ามีเปอร์เซ็นต์ความถูกต้องสูงกว่าเบสไลน์เพียงเล็กน้อยเพียง 72%

อัลกอริทึมอดาบูสเอ็มวัน(AdaBoostM1) ในกลุ่มอัลกอริทึมเมต้า (Meta) ได้ผลลัพธ์ดังรูป

Correctly Classified Instances	696	69.6	%
Incorrectly Classified Instances	304	30.4	%

รูป 4.157 ผลลัพธ์เปอร์เซ็นต์ความถูกต้องจากอัลกอริทึม AdaBoostM1 ของชุดข้อมูลสินเชื่อ

อัลกอริทึมวันอาร์ (OneR) ในกลุ่มอัลกอริทึมรูล (Rule) ได้ผลลัพธ์ดังรูป

Correctly Classified Instances	661	66.1	%
Incorrectly Classified Instances	339	33.9	%

รูป 4.158 ผลลัพธ์เปอร์เซ็นต์ความถูกต้องจากอัลกอริทึม OneR ของชุดข้อมูลสินเชื่อ

พบว่ามีเปอร์เซ็นต์ความถูกต้องน้อยกว่าค่าเบสไลน์ จึงไม่เหมาะสมจะใช้อัลกอริทึม OneR กับชุดข้อมูลสินเชื่อ

อัลกอริทึมเจสี่สิบแปด (J48) ในกลุ่มอัลกอริทึมทรี (Tree) ได้ผลลัพธ์ดังรูป

Correctly Classified Instances	705	70.5	%
Incorrectly Classified Instances	295	29.5	%

รูป 4.159 ผลลัพธ์เปอร์เซ็นต์ความถูกต้องจากอัลกอริทึม J48 ของชุดข้อมูลสินเชื่อ

พบว่ามีค่าเปอร์เซ็นต์ความถูกต้องมากกว่าค่าเบสไลน์เพียงนิดเดียวคือเท่ากับ 70.5%

4.7 เปรียบเทียบประสิทธิภาพของแต่ละซอฟต์แวร์

ในการทดลองเปรียบเทียบประสิทธิภาพของซอฟต์แวร์เพื่อเปรียบเทียบความสามารถในการทำนายผลให้แม่นยำของซอฟต์แวร์ทั้ง 3 ตัว ได้แก่ เวก้า, ราปิดมายเนอร์ และอาร์ ดาต้ามายนิ่ง โดยทำการทดลองกับชุดข้อมูล 7 ชุดที่ทำการแบ่งเป็นชุดข้อมูลเรียนรู้ 90% และชุดข้อมูลทดสอบ 10% ทำการเฉลี่ยเปอร์เซ็นต์ความถูกต้องจากการทดลองทั้งหมด 10 Random seed โดยชุดข้อมูลทั้ง 7 ชุด ได้แก่

1) ดอกไอริส (Iris.arff)
2) กระจก (Glass.arff)
3) มะเร็งเต้านม (Breast-cancer.arff)
4) เบาหวาน (Diabetes.arff)
5) ถั่วเหลือง (Soybean.arff)
6) การลงคะแนนเสียงเลือกตั้ง (Vote.arff)
7) สินเชื่อ (Credit-g.arff)

จากการทดลอง ได้ผลการทดลองดังตารางต่อไปนี้

ตาราง 4.1 เปอร์เซ็นต์ความถูกต้องเฉลี่ยจากการทดลองด้วยชุดข้อมูลดอกไอริส

Algorithms	Software		
	Weka	RapidMiner	R Data Mining
Naïve Bayes	97.33%	98.00%	98.00%
Decision tree	96.67%	97.33%	96.00%
AdaBoost	95.33%	97.33%	97.33%

ตาราง 4.2 เปอร์เซ็นต์ความถูกต้องเฉลี่ยจากการทดลองด้วยชุดข้อมูลกระจก

Algorithms	Software		
	Weka	RapidMiner	R Data Mining
Naïve Bayes	46.67%	42.38%	36.67%
Decision tree	63.33%	61.90%	55.24%
AdaBoost	70.00%	66.19%	70.95%

ตาราง 4.3 เปอร์เซ็นต์ความถูกต้องเฉลี่ยจากการทดลองด้วยชุดข้อมูลมะเร็งเต้านม

Algorithms	Software		
	Weka	RapidMiner	R Data Mining
Naïve Bayes	76.43%	74.64%	72.15%
Decision tree	76.07%	72.50%	-
AdaBoost	67.86%	61.79%	71.79%

ตาราง 4.4 เปอร์เซ็นต์ความถูกต้องเฉลี่ยจากการทดลองด้วยชุดข้อมูลเบาหวาน

Algorithms	Software		
	Weka	RapidMiner	R Data Mining
Naïve Bayes	77.89%	78.29%	78.29%
Decision tree	73.69%	73.95%	75.66%
AdaBoost	74.61%	73.95%	73.95%

ตาราง 4.5 เปอร์เซ็นต์ความถูกต้องเฉลี่ยจากการทดลองด้วยชุดข้อมูลถั่วเหลือง

Algorithms	Software		
	Weka	RapidMiner	R Data Mining
Naïve Bayes	91.03%	93.97%	91.91%
Decision tree	89.85%	91.18%	-
AdaBoost	94.27%	48.53%	94.41%

ตาราง 4.6 เปอร์เซ็นต์ความถูกต้องเฉลี่ยจากการทดลองด้วยชุดข้อมูลการลงคะแนนเสียงเลือกตั้ง

Algorithms	Software		
	Weka	RapidMiner	R Data Mining
Naïve Bayes	88.37%	88.84%	88.37%
Decision tree	94.88%	94.42%	95.81%
AdaBoost	94.65%	50.00%	94.65%

ตาราง 4.7 เปอร์เซ็นต์ความถูกต้องเฉลี่ยจากการทดลองด้วยชุดข้อมูลสินเชื่อ

Algorithms	Software		
	Weka	RapidMiner	R Data Mining
Naïve Bayes	74.50%	75.10%	75.10%
Decision tree	69.80%	71.90%	67.29%
AdaBoost	70.50%	66.80%	73.63%

บทที่ 5
บทสรุปและข้อเสนอแนะ

5.1 สรุปและวิเคราะห์ผล

5.1.1 สรุปประสิทธิภาพในการสร้างโมเดลของซอฟต์แวร์

จากการทดลองซอฟต์แวร์ทั้ง 3 ตัว ซึ่งได้แก่ Weka, RapidMiner และ R Data Mining ร่วมกับอัลกอริทึม 3 ตัว ซึ่งได้แก่ Naïve Bayes, Decision tree และ AdaBoost กับชุดข้อมูล 7 ชุด สามารถสรุปผลได้ว่า ซอฟต์แวร์แต่ละตัวมีประสิทธิภาพในการสร้างโมเดลสำหรับทำนายได้ใกล้เคียงกัน โดยสามารถเห็นได้ชัดว่าซอฟต์แวร์ Weka จะมีจุดเด่นในการใช้อัลกอริทึม J48 (Decision tree) ซึ่งเป็นอัลกอริทึมตัวสำคัญของ Weka เนื่องจากมีการปรับปรุงอัลกอริทึมมาจาก C4.5 เพื่อใช้งานกับ Weka โดยเฉพาะ และสำหรับ RapidMiner และ R Data Mining มีวิธีการคำนวณแต่ละอัลกอริทึมเหมือนกัน ทำให้ผู้ทำการทดลองสรุปได้ว่า ซอฟต์แวร์แต่ละตัวมีความถนัดในการใช้อัลกอริทึมต่างๆ ไม่เท่ากันและจากการสังเกตผลของเปอร์เซ็นต์ความถูกต้องของโมเดลที่ใช้วัดประสิทธิภาพของซอฟต์แวร์นั้น พบว่าไม่มีซอฟต์แวร์ใดที่มีผลโดดเด่นไปกว่าซอฟต์แวร์อื่นๆ จึงสามารถสรุปได้ว่า ผลของเปอร์เซ็นต์ความถูกต้องของโมเดลนั้นขึ้นอยู่กับแต่ละอัลกอริทึมที่ผู้ใช้เลือกใช้และขึ้นอยู่กับลักษณะที่ต่างกันของชุดข้อมูล เช่น ลักษณะของชุดข้อมูล, จำนวน Instance หรือจำนวนแอตทริบิวต์ในชุดข้อมูล เป็นต้น

5.1.2 จุดอ่อนจุดแข็งของแต่ละซอฟต์แวร์

สรุปจุดอ่อนและจุดแข็งของแต่ละซอฟต์แวร์ได้ดังนี้

ตาราง 5.1 แสดง Performance ในด้านต่างๆ ของซอฟต์แวร์จากการใช้งานจริง

FEATURES AVAILABILITY	Software		
	Weka	RapidMiner	R Data Mining
ความสะดวกในการใช้งาน	ใช้งานง่าย การตั้งค่าในแต่ละอัลกอริทึมทำได้ง่าย มีการแบ่งประเภทของการทำเหมืองข้อมูลอย่างชัดเจน หากมีการ Error ซอฟท์แวร์จะ	มี Tutorial สอนในระดับเบื้องต้น แต่ในการใช้งานจริง การเลือกใช้แต่โมดูลยังมีความซับซ้อนอยู่มาก หากผู้ใช้งานไม่มีความเข้าใจการ	การใช้งานค่อนข้างเข้าใจยากสำหรับผู้เริ่มต้น ใช้ภาษา R ในการเขียนคำสั่งทุกอย่าง เกิด Error ได้ง่ายมาก ซึ่งปัญหา Error อาจจะไม่ได้

	มีการแจ้งเตือนอย่างชัดเจน	ทำงานของแต่ละโมดูลที่ดีพอ จะทำให้กระบวนการเกิด Error ได้ง่าย ผู้ใช้สามารถสร้างกระบวนการที่ต่อเนื่องกันได้ในการรันเพียงครั้งเดียว	อยู่ที่โค้ด แต่อาจอยู่ที่ชุดข้อมูล ซึ่งยากต่อการแก้ไข
รูปแบบการนำเสนอผลการวิเคราะห์	การทำนายค่าแบบแผนภาพต้นไม้ สามารถแสดงโมเดลแผนภาพช่วยให้เข้าใจได้ง่าย การแสดงผลอื่นๆจะแสดงในรูปแบบของกราฟสองมิติ พร้อมทั้งมีการใช้สีเพื่อจำแนกความแตกต่างของแต่ละ class ในการแสดงสถิติต่างๆ ของการเรียนรู้มีรูปแบบการแสดงผลที่เข้าใจง่าย	มีการแสดงผลการทดสอบในรูปของ Metrix เป็นหลัก สามารถบันทึกโมเดลออกเป็นไฟล์ภาพประเภทต่างๆได้ ในส่วนของการแสดงผลอื่นๆทั้งแบบกราฟหรือโมเดล ผู้ใช้สามารถทำการต่อโมดูลให้แสดงผลได้ตามใจ โดยสามารถแสดงผลทั้งหมดได้ในการรันเพียงครั้งเดียว	สามารถ plot ออกมาเป็นกราฟในรูปแบบที่ต้องการได้โดยการเขียนโค้ดคำสั่ง และสามารถ Export เป็นไฟล์ได้
ความสามารถในการรองรับรูปแบบไฟล์ที่หลากหลาย	รองรับไฟล์ได้หลากหลายสกุลไฟล์	มีข้อจำกัดในการ importข้อมูลเข้าใช้งานในไฟล์บางประเภทเนื่องจากซอฟท์แวร์ตัวนี้มีการแบ่งระดับของผู้ใช้งานไว้หลายระดับ ซึ่งแต่ละระดับ	รองรับได้หลากหลายสกุลไฟล์ โดยการ install แพ็กเกจเพิ่ม

		จะใช้งานซอฟท์แวร์ได้ไม่เท่ากัน	
ความหลากหลายของอัลกอริทึมที่ใช้ในการจำแนกข้อมูล	มีจำนวนอัลกอริทึมสำหรับการจำแนกกว่า100อัลกอริทึม โดยอัลกอริทึมที่โดดเด่นอย่าง J48 ก็มีการพัฒนาสำหรับนำมาใช้ใน Weka โดยเฉพาะ	มีอัลกอริทึมสำหรับทำการจำแนกค่อนข้างหลากหลาย โดยมีการจำแนกเป็นหมวดหมู่	สามารถรองรับอัลกอริทึมได้หลากหลาย เกือบทุกอัลกอริทึมที่ใช้ในการทำ Machine learning ทำได้โดยการ install แพ็กเกจที่ต้องการ
ความเร็วในการประมวลผล	ใช้ระยะเวลาในการประมวลผลได้เร็วมาก	หากเป็นข้อมูลขนาดใหญ่การประมวลผลจะช้าเพียงเล็กน้อย	ค่อนข้างใช้เวลานานในการประมวลผลถ้าเทียบกับ Weka และ RapidMiner
การใช้งานทรัพยากรเครื่องคอมพิวเตอร์	ไม่กินทรัพยากรเครื่อง	หากเป็นกระบวนการทำงานที่ใหญ่โปรแกรมจะทำงานช้าลงตามลำดับ	ค่อนข้างกินทรัพยากรเครื่องมาก โปรแกรมค้างบ่อยครั้ง

5.1.3 อัลกอริทึมที่เหมาะสมกับข้อมูลแต่ละประเภท

ตาราง 5.2 แสดงการสรุปผลลัพธ์การทดลองแต่ละอัลกอริทึมกับแต่ละชุดข้อมูล

Datasets	Algorithms		
	Naïve Bayes	Decision Tree	AdaBoost
Iris.arff	96.00%	96.00%	93.33%
Glass.arff	48.60%	66.82%	74.30%
Breast-cancer.arff	71.68%	75.52%	69.58%
Diabetes.arff	76.30%	73.83%	72.40%
Soybean.arff	92.97%	91.51%	92.83%
Vote.arff	90.11%	96.32%	95.86%
Credit-g.arff	75.40%	70.50%	69.60%

จากการศึกษาเปรียบเทียบผลลัพธ์ความถูกต้องของการทดลองชุดข้อมูลแต่ละตัวกับอัลกอริทึมทั้งสามตัวซึ่งได้แก่ Naïve Bayes, Decision tree และ AdaBoost สรุปได้ดังนี้

1. Naïve Bayes ค่อนข้างเหมาะกับข้อมูลที่มีจำนวนแอตทริบิวต์มาก เนื่องจากการคำนวณของอัลกอริทึม Naïve Bayes จะทำการคำนวณค่าความน่าจะเป็นของ Instance ใหม่จากความน่าจะเป็นของทุกๆ แอตทริบิวต์ใน Instance เดิม
2. Decision tree เหมาะสำหรับข้อมูลที่ประกอบด้วยแอตทริบิวต์ที่มีความสำคัญลดหลั่นเป็นลำดับ เนื่องจากอัลกอริทึมในกลุ่ม Decision tree รวมทั้ง J48 มีการคำนวณค่า Gain จากชุดข้อมูลเดิมในการเลือก Root และ Leaf ตามลำดับความสำคัญของแอตทริบิวต์
3. AdaBoost เหมาะสำหรับข้อมูลที่ประกอบด้วยแอตทริบิวต์ที่เป็นคลาสซึ่งมีจำนวน Instance ของคลาสหนึ่งมากกว่าคลาสอื่นๆ มาก หรือเรียกอีกอย่างหนึ่งว่าจำนวน Instance ของแต่ละคลาสมีความแตกต่างกันมาก ชุดข้อมูลลักษณะนี้จะเหมาะกับอัลกอริทึม AdaBoost เนื่องจากหลักการของอัลกอริทึมนี้คือการให้น้ำหนักกับแอตทริบิวต์ของคลาสที่มีจำนวน Instance น้อยกว่า

5.1.4 การแบ่งชุดข้อมูลที่มีประสิทธิภาพ

ในการทำการทดลอง ชุดข้อมูลที่นำมาทำการทดลองจะต้องมีการแบ่งข้อมูลออกเป็น Training set และ Test set โดยแบ่งจากชุดข้อมูลต้นฉบับ ซึ่งจะทำการแบ่งออกเป็น 3 แบบคือ Split 90%, Split 80% และ Split 70% โดยยกตัวอย่างการนำข้อมูลที่แบ่งด้วยลักษณะต่างๆ ไปทดลองกับอัลกอริทึมจะได้เปอร์เซ็นต์ความถูกต้องของโมเดลเป็นดังนี้

ตาราง 5.3 ผลการทดลองด้วยชุดข้อมูล Iris.arff กับอัลกอริทึม Naïve Bayes

โปรแกรมที่ใช้ทดลอง	Split 90%	Split 80%	Split 70%
Weka	97.33%	96.67%	96.00%
RapidMiner	98.00%	97.00%	96.67%
R Data Mining	98.00%	97.00%	96.67%

ตาราง 5.4 ผลการทดลองด้วยชุดข้อมูล Diabetes.arff กับอัลกอริทึม Decision tree

โปรแกรมที่ใช้ทดลอง	Split 90%	Split 80%	Split 70%
Weka	73.69%	74.05%	74.17%
RapidMiner	73.95%	70.98%	73.00%
R Data Mining	75.66%	75.30%	74.52%

จากการทดลองสามารถสรุปผลได้ว่า การแบ่งชุดข้อมูลแบบ Split 90% จะให้ความแม่นยำมากที่สุดเป็นส่วนใหญ่ เนื่องจากมีข้อมูลที่ใช้เป็นเป็นตัวอย่างในการสร้างโมเดลจำนวนมากที่สุด รองลงมาคือ Split 80% และ Split 70% ตามลำดับ

5.2 ปัญหาอุปสรรคและแนวทางการแก้ไข

5.2.1 ปัญหาและอุปสรรคในการทดลองด้วยเวก้า

เวก้าเป็นซอฟต์แวร์ตัวแรกที่ทำการศึกษาเกี่ยวกับเรื่องการทำเหมืองข้อมูล ทำให้ต้องใช้ความพยายามอย่างมากในการศึกษาเรื่องการทำงานของอัลกอริทึมต่างๆ อย่างละเอียด รวมไปถึงการทดลองข้อมูลแบบต่างๆ เช่น Cross-validation, Supplied test set และ Percentage split เป็นต้น และต้องศึกษาเรื่องวิธีการทำเหมืองข้อมูลเพื่อให้เข้าใจหลักการทำงานโดยใช้เวก้าเป็นตัวเริ่มต้น

5.2.2 ปัญหาและอุปสรรคในการทดลองด้วยราปิดมายเนอร์

1. อุปสรรคในการขอ License ราปิดมายเนอร์ที่เปิดให้ใช้ฟรีไม่รองรับการอิมพอร์ตไฟล์ ARFF ต้องทำการแจ้งอีเมล์เพื่อขอ License สำหรับเวอร์ชั่นที่สามารถรองรับไฟล์ ARFF ได้ ซึ่งแต่ละ License ใช้งานได้เพียง 14 วัน จึงต้องหาอีเมล์ใหม่มาขอ License ทุกๆ 14 วัน จนกว่าจะทดลองเสร็จ
2. ปัญหาในการเลือกใช้โมดูล ถ้าไม่มีความเข้าใจในการทำงานหรือฝึกฝนให้ใช้งานจนชำนาญจะทำให้ไม่รู้ว่าควรจะเลือกใช้โมดูลในหมวดใดมาต่อ

5.2.3 ปัญหาและอุปสรรคในการทดลองด้วยอาร์ ดาต้ามายนิ่ง

1. ปัญหาการเริ่มใช้งานอาร์ ดาต้ามายนิ่ง การเริ่มใช้งานค่อนข้างยาก ต้องทำการศึกษาเยอะ เนื่องจากการใช้งานค่อนข้างอิสระ ทำการแก้ปัญหาโดยการศึกษาข้อมูลและตัวอย่างการใช้งานต่างๆ มากขึ้น
2. การใช้งานแต่ละอัลกอริทึมมีการเขียนโค้ดภาษาอาร์แตกต่างกัน ทำให้ต้องศึกษาใหม่ทุกครั้งหากต้องการทดลองอัลกอริทึมที่ไม่เคยใช้มาก่อน
3. ในการใช้อัลกอริทึม Decision tree มีข้อจำกัดค่อนข้างมาก เนื่องจากชุดข้อมูลบางชุดในบาง Instance มีบางแอตทริบิวต์ที่ข้อมูลหายไป ทำให้ไม่สามารถใช้อัลกอริทึม Decision tree ได้ โดยชุดข้อมูลที่ไม่สามารถใช้อัลกอริทึม Decision tree ได้แก่ Breast-cancer.arff, Credit-g.arff และ Soybean.arff

5.2.4 ปัญหาและอุปสรรคในการควบคุมชุดข้อมูลที่นำมาใช้

เนื่องจากการทดลองในหลายซอฟต์แวร์จำเป็นต้องใช้ชุดข้อมูลที่เหมือนกัน เพื่อที่จะสามารถเปรียบเทียบผลการทดลองได้ ปัญหาเดียวของการควบคุมชุดข้อมูลที่นำมาใช้คือ ผู้ทำโครงงานต้องนำชุดข้อมูลฉบับเต็มมาแบ่งเป็น Training set และ Test set ด้วยตนเอง ซึ่งชุดข้อมูลที่ใช้มีจำนวนมาก และในแต่ละชุดข้อมูลยังมีการแบ่งหลายแบบรวมแล้วกว่า 210 ชุดข้อมูลที่นำมาทดลอง ทำให้ใช้เวลามากในการเตรียมชุดข้อมูลนี้

5.3 แนวทางการพัฒนาต่อ

1. สามารถนำไปศึกษาต่อในด้านการแบ่งกลุ่มข้อมูล (Clustering) หรือการจำแนกโดยการใช้กฎความสัมพันธ์ (Association rule)
2. ศึกษาเจาะลึกเพิ่มเติมไปยังอัลกอริทึมการจำแนกข้อมูลแต่ละตัว เพื่อดูการทำงานและความเหมาะสมระหว่างชุดข้อมูลกับแต่ละอัลกอริทึม
3. ทำการศึกษาเพิ่มเติมจากชุดข้อมูลอื่นๆ ที่เป็นชุดข้อมูลจริง หรือทำเป็น Case study ตามที่สนใจ

บรรณานุกรม

Witten, I.H. Frank, E. and Hall, M.A. **Data Mining: Practical Machine Learning Tools and Techniques.** Massachusetts : Morgan Kaufmann.

Wu, X. and Kumar, V. 2009. **The Top Ten Algorithms in Data Mining.** Florida : Taylor & Francis Group.

มหาวิทยาลัยขอนแก่น. n.d. **AdaboostSVM-Based Techniques for Image Classification.** [Online]. Available : http://tar.thailis.or.th/bitstream/123456789/24/1/CIT2010_03.pdf.

อดุลย์ ยิ้มงาม. 2008. **Computer Center – การทำเหมืองข้อมูล (Data Mining).** [Online]. Available : http://compcenter.bu.ac.th/index.php?option=com_content&task=view&id=75&Itemid=172.

เอกสิทธิ์ พัชรวงศ์ศักดา. 2014. **ขั้นตอนการสร้างโมเดล Decision Tree.** [Online]. Available : http://dataminingtrend.com/2014/decision-tree-model.

เอกสิทธิ์ พัชรวงศ์ศักดา. 2014. **บทที่ 1 แนะนำการใช้งาน RapidMiner Studio 6 | Data Mining Trend.** [Online]. Available : http://dataminingtrend.com/2014/rapidminer-studio-6/chapter1.

Ott T. 2015. **Tutorials – Neural Market Trends.** [Online]. Available : http://www.neuralmarkettrends.com/tutorials.

RapidMiner. 2014. **Operater Manual – RapidMiner Documentation.** [Online]. Available : http://docs.rapidminer.com/studio/operators/

RapidMiner. 2014. **RapidMiner Studio – RapidMiner Documentation.** [Online]. Available : http://docs.rapidminer.com/studio.