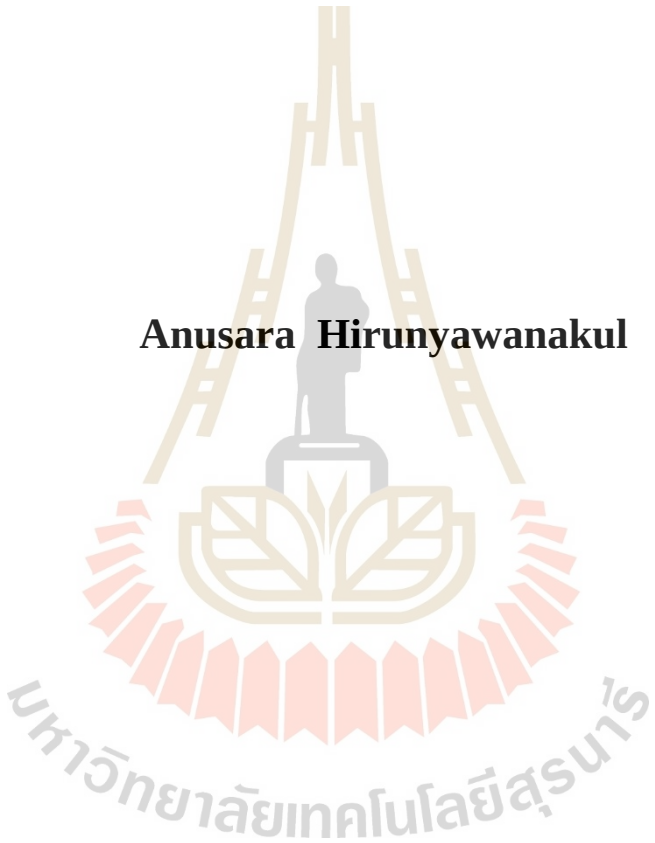
เทคนิคการเพิ่มประสิทธิภาพการคาดการณ์ผลผลิตในอุตสาหกรรมการผลิตฮาร์ดดิสก์ไดรฟ์

นางสาวอนุสรา หิรัญวนากุล

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรดุษฎีบัณฑิต
สาขาวิชาวิศวกรรมโทรคมนาคมและคอมพิวเตอร์
มหาวิทยาลัยเทคโนโลยีสุรนารี
ปีการศึกษา 2563

TECHNIQUES TO IMPROVE YIELD PREDICTION IN HARD DISK DRIVE MANUFACTURING INDUSTRY

Anusara Hirunyawanakul



A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy in Telecommunication and Computer Engineering

Suranaree University of Technology

Academic Year 2020

เทคนิคการเพิ่มประสิทธิภาพการคาดการณ์ผลผลิตในอุตสาหกรรม
การผลิตฮาร์ดดิสก์ไดรฟ์

มหาวิทยาลัยเทคโนโลยีสุรนารี อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาดุษฎีบัณฑิต

คณะกรรมการสอบวิทยานิพนธ์

(อ. ดร. กีระชาติ สุขสุทธิ์)
ประธานกรรมการ

(รศ. ดร. นิตยา เกิดประสพ)
กรรมการ (อาจารย์ที่ปรึกษาวิทยานิพนธ์)

(รศ. ดร. กิตติศักดิ์ เกิดประสพ)
กรรมการ

(ผศ. ร.อ. ดร. ประโยชน์ คำสวัสดิ์)
กรรมการ

(ผศ. ดร. วิภาวี อุสาหะ)
กรรมการ

(รศ. ร.อ. ดร.กนต์ธร ชำนิประศาสน์)
รองอธิการบดีฝ่ายวิชาการและพัฒนาความเป็นสากล

(รศ. ดร. พรศิริ จงกล)
คณบดีสำนักวิชาวิศวกรรมศาสตร์

อนุสรา หิรัญวนากุล: เทคนิคการเพิ่มประสิทธิภาพการคาดการณ์ผลผลิตในอุตสาหกรรมการผลิตฮาร์ดดิสก์ไดรฟ์ (TECHNIQUES TO IMPROVE YIELD PREDICTION IN HARD DISK DRIVE MANUFACTURING INDUSTRY) อาจารย์ที่ปรึกษา : รองศาสตราจารย์ ดร.นิตยา เกิดประสพ, 182 หน้า.

ในวงการอุตสาหกรรมการผลิตฮาร์ดดิสก์ไดรฟ์นั้นการคาดการณ์ผลผลิต (Yield Prediction) เป็นงานที่มีความสำคัญที่สุดอย่างหนึ่ง ความแม่นยำของการคาดการณ์ผลผลิตมีผลกระทบและบทบาทอย่างสูงในธุรกิจนี้ การคาดการณ์ผลผลิตให้มีความแม่นยำอาจจะเกิดอุปสรรคได้จากหลายปัจจัย โดยหนึ่งในปัจจัยสำคัญที่สุดคือการที่ข้อมูลการผลิตและการทดสอบผลิตภัณฑ์นั้นมีจำนวนมหาศาลทำให้มีความยากต่อการนำไปใช้งาน นอกเหนือจากนั้นแล้วข้อมูลเหล่านั้นมีความไม่สมดุลของข้อมูลระหว่างฮาร์ดดิสก์ไดรฟ์ที่ผ่านและไม่ผ่านการทดสอบอยู่ในระดับที่สูงมาก ด้วยเหตุนี้เองการสร้างโมเดลสำหรับการคาดการณ์ผลผลิตที่แม่นยำนั้นจึงเป็นเรื่องที่นับได้ว่ามีความท้าทายเป็นอย่างยิ่ง งานวิจัยนี้จึงได้นำเสนอการเพิ่มประสิทธิภาพในการทำงานของการคาดการณ์ผลผลิตในอุตสาหกรรมการผลิตฮาร์ดดิสก์ไดรฟ์โดยการนำความรู้ทางด้านการเรียนรู้ของเครื่องและวิธีการต่าง ๆ ทางด้านการทำเหมืองข้อมูล (Data Mining) เข้ามาช่วยจัดการปัญหา ทั้งการนำเทคนิคการทำข้อมูลให้สมดุล (Data Balancing) การนำอัลกอริทึมการเรียนรู้และวิธีการทางสถิติมาใช้ในการคัดเลือกคุณลักษณะจากผลลัพธ์ของการทดสอบผลิตภัณฑ์ รวมไปถึงการนำอัลกอริทึมการเรียนรู้มาใช้ในการสร้างโมเดลการคาดการณ์ผลผลิตของกระบวนการ นอกเหนือจากนั้นแล้วงานวิจัยนี้ยังได้นำเสนอเทคนิคใหม่คือการรวมกลุ่มข้อมูลมาประยุกต์ใช้ในการคำนวณผลผลิตโดยจะเป็นการรวมกลุ่มของข้อมูลในลักษณะของการแบ่งกลุ่มข้อมูลตามจำนวนค่าคงที่ที่ได้กำหนดไว้แทนที่วิธีการดั้งเดิมที่อ้างอิงการรวมกลุ่มของข้อมูลเป็นรายสัปดาห์เพื่อการคำนวณผลผลิต การทดสอบประสิทธิภาพใช้เกณฑ์การวัดค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ย (Mean Absolute Error: MAE) และค่ารากที่สองของความคลาดเคลื่อนกำลังสองเฉลี่ย (Root Mean Squared Error: RMSE) เป็นหลัก

จากผลการดำเนินงานวิจัยพบว่าเทคนิคใหม่ที่นำเสนอการทำสมดุลข้อมูลโดยใช้วิธีการ Data Balancing by k-Means Clustering k-Nearest Neighbors and Re-Sampling (DBC-2KAR) ซึ่งเป็นการประยุกต์อัลกอริทึมทางด้านการเรียนรู้ของเครื่องร่วมกับการสุ่มข้อมูลเข้าด้วยกันนั้นสามารถทำให้ข้อมูลเกิดความสมดุลและส่งผลให้โมเดลการเรียนรู้ในด้านการจำแนกสามารถคัดเลือกคุณลักษณะที่สำคัญออกมาได้ ขั้นตอนถัดมาในส่วนของการนำคุณลักษณะที่ได้มาดำเนินการสร้างโมเดลการคาดการณ์ผลผลิตด้วยอัลกอริทึม Artificial Neural Network และ Multiple Linear Regression ผนวกเข้ากับการรวมกลุ่มข้อมูลด้วยวิธีการรวมกลุ่มข้อมูลด้วยค่าคงที่

ด้วยวิธีการที่ได้กล่าวมาข้างต้นนั้นจะให้ผลลัพธ์ที่ดีกว่าวิธีการดั้งเดิมที่วิศวกรกระบวนการใช้งานอยู่ในปัจจุบัน โดยวิธีการที่ดีที่สุดได้แก่การคัดเลือกคุณลักษณะด้วยอัลกอริทึม Genetic Algorithm ร่วมกับการสร้างโมเดลการเรียนรู้ด้วยอัลกอริทึม Multiple Linear Regression โดยใช้การรวมกลุ่มแบบค่าคงที่ซึ่งจะให้ค่า MAE และ RMSE อยู่ที่ 0.559 และ 0.732 ตามลำดับ ถ้าหากเปรียบเทียบกับวิธีการดั้งเดิมแล้วค่าความผิดพลาดที่วัดได้จะลดลงไปถึง 60% โดยในส่วนของการทดลองการเพิ่มค่าคงที่ของการรวมกลุ่มข้อมูลนั้นพบว่าส่งผลให้มีค่าความผิดพลาดลดลงตั้งแต่ค่าคงที่เท่ากับ 1,000 และน้อยที่สุดเมื่อมีค่าเท่ากับ 10,000 จากนั้นแม้ว่าจะเพิ่มค่าคงที่การรวมกลุ่มข้อมูลไปจนถึง 40,000 ประสิทธิภาพในการคาดการณ์ผลผลิตจะอยู่ในระดับคงที่

สาขาวิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อนักศึกษา_____

ปีการศึกษา 2563 ลายมือชื่ออาจารย์ที่ปรึกษา_____

ANUSARA HIRUNYAWANAKUL : TECHNIQUES TO IMPROVE YIELD PREDICTION IN HARD DISK DRIVE MANUFACTURING INDUSTRY. THESIS ADVISOR : ASSOC. PROF. NITTAYAKERDPRASOP, Ph.D., 182 PP.

YIELD PREDICTION IN MANUFACTURING/MACHINE LEARNING/ IMBALANCED DATA/DATA BALANCING/FEATURE SELECTION/ GENETIC ALGORITHM/MULTIPLE LINEAR REGRESSION

In the field of hard disk drive manufacturing, yield prediction is one of the most important tasks. An accurate yield prediction has high contribution and influence on this business. There are many factors to obstruct the process of yield prediction modeling. One of key problems is tremendous amount of features collected from manufacturing and testing process. This problem creates difficulty for the data usage. Moreover, imbalance ratio between passed and failed hard disk drive units in the manufacturing data is extremely high. This issue makes the creation of an accurate yield prediction model a challenging task. This research introduces techniques to improve yield prediction performance in hard disk drive manufacturing by applying knowledge of machine learning and several techniques of data mining to solve the problem. Techniques of data balancing and the application of learning algorithm and statistics are used in data preparation and feature selection process. Machine learning techniques are also applied in modeling for yield prediction step. This research also introduces the novel method in data grouping step called data aggregation by consistency quantity to replace the original technique that data are grouped by week.

Performance has been evaluated through the Mean Square Error (MAE) and Root Mean Square Error (RMSE) measurements.

The experimental result shows that the proposed method, which is a combination of machine learning algorithms and resampling techniques called DBC-2KAR (Data Balancing k-Means Clustering, k-Nearest Neighbors and Resampling), is able to turn a highly imbalanced dataset to be a balanced and efficient one. Efficiency is due to the fact that the classification learning model can find the important attributes in feature selection step. These important attributes are to be used in the subsequent step for yield prediction modeling with algorithm Artificial Neural Network and Multiple Linear Regression combined with the novel technique called Data Aggregate by Fixed number that can provide better result than the traditional method adopted by current process engineers. The best methodology is the combination of feature selection using genetic algorithm and yield prediction modeling with multiple linear regression. RMSE and MAE of this combination are 0.559 and 0.732, respectively. In terms of error rate reduction from traditional method, this method can reduce the error rate by 60%. In the portion of experiment to increase fixed number of Data Aggregation, we found that errors of yield prediction are getting lower since the fixed number is 1,000. The lowest error of yield prediction is provided when fixed numbers is 10,000 and results are steady even if the fixed number has been increased up to 40,000.

School of Computer Engineering Student's Signature 

Academic Year 2020 Advisor's Signature 

กิตติกรรมประกาศ

การที่วิทยานิพนธ์นี้สำเร็จลุล่วงด้วยดี ผู้วิจัยขอกราบขอบพระคุณ บุคคล และกลุ่มบุคคลต่าง ๆ ที่ได้กรุณาให้คำปรึกษา แนะนำ ช่วยเหลืออย่างดียิ่ง ทั้งในด้านวิชาการ และด้านการดำเนินงานวิจัยดังต่อไปนี้

รองศาสตราจารย์ ดร.กิตติศักดิ์ เกิดประสพ และรองศาสตราจารย์ ดร.นิตยา เกิดประสพ อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่ให้โอกาส ให้การอบรม สั่งสอนชี้แนะ ให้กำลังใจ รวมไปถึงให้คำแนะนำเกี่ยวกับรูปแบบในการนำเสนอผลการศึกษา และตรวจแก้ไขวิทยานิพนธ์จนเสร็จสมบูรณ์

ท่านคณะกรรมการผู้ทรงคุณวุฒิทุกท่านที่เสียสละเวลาทำหน้าที่กรรมการสอบ รวมไปจนถึงการให้คำแนะนำเพื่อให้วิทยานิพนธ์ฉบับนี้สมบูรณ์มากยิ่งขึ้น

มหาวิทยาลัยเทคโนโลยีสุรนารีที่ได้เปิดโอกาสให้สำหรับการศึกษาต่อ รวมถึงการให้การสนับสนุนทุนการศึกษา และทุนอุดหนุนในการนำเสนอและเผยแพร่ผลงานวิจัย

ผู้ช่วยศาสตราจารย์. ดร.นันทวุฒิ คะอังกุ และนายวุฒิกร เตชะเวชเจริญ ที่ช่วยให้คำปรึกษาในด้านการดำเนินงานวิจัย ให้กำลังใจและช่วยเหลือในการทำวิจัย รวมไปถึงการช่วยตรวจทานความถูกต้องในการเขียนวิทยานิพนธ์

นายอนุพงษ์ บรรจงการ ดร.สุภาพร บุญฤทธิ์ นายอภิรักษ์ วรกานตพล และนายภูมิรพี ภูมิค้า ตลอดจนนักศึกษาร่วมชั้นเรียนทั้งปริญญาโทและปริญญาเอก ทั้งในอดีตที่สำเร็จการศึกษาไปแล้ว และที่ยังศึกษาอยู่ในปัจจุบัน ที่ช่วยให้คำแนะนำ ให้ความช่วยเหลือในด้านต่าง ๆ

คุณปราณี กฐินใหม่ เลขานุการสาขาวิชาวิศวกรรมคอมพิวเตอร์ ที่ให้ความช่วยเหลือในการประสานงานระหว่างศึกษา

ขอกราบขอบพระคุณ บิดา มารดา ที่ให้กำเนิด อบรม เลี้ยงดูด้วยความรัก และส่งเสริมการศึกษาเป็นอย่างดีโดยตลอด ทำให้ผู้วิจัยมีความรู้ ความสามารถ จนทำให้ผู้วิจัยประสบความสำเร็จในชีวิตเรื่อยมา นอกจากนี้ขอขอบคุณครู อาจารย์ทั้งในอดีตและปัจจุบันที่ให้ความรู้แก่ผู้วิจัยจนประสบความสำเร็จในชีวิต

ท้ายที่สุดที่จะลืมไม่ได้ขอขอบคุณน้องสาวอันเป็นที่รัก และบุคคลอันเป็นที่รักของครอบครัวผู้ได้ล่วงลับไปแล้ว ซึ่งถือได้ว่าเป็นอีกหนึ่งกำลังใจที่ยิ่งใหญ่ ทำให้ผู้วิจัยมีจิตใจเข้มแข็งไม่ย่อท้อต่ออุปสรรคต่าง ๆ จนสามารถทำให้วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงด้วยดี

อนุสรา หิรัญวนากุล

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

ตารางที่ **หน้า**

สารบัญตาราง (ต่อ)

ตารางที่ **หน้า**

สารบัญรูป

รูปที่ **หน้า**

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มาของปัญหาการวิจัย

ในยุคปัจจุบันนี้ถึงแม้อุปกรณ์บันทึกข้อมูลในรูปแบบโซลิดสเตทไดรฟ์ (Solid State Drive) นั้นจะมีการพัฒนารุดหน้าไปอย่างรวดเร็ว แต่ฮาร์ดดิสก์ไดรฟ์ยังคงเป็นอุปกรณ์ที่มีบทบาทสำคัญที่สุดในการจัดเก็บข้อมูลจำนวนมหาศาลอยู่เสมอ ด้วยเหตุผลของความน่าเชื่อถือในการจัดเก็บข้อมูล (Reliability) ที่สูงกว่าโซลิดสเตทไดรฟ์ และราคาต่อความจุ (Cost per Terabyte) ซึ่งเมื่อเทียบกันแล้วยังถือว่าถูกกว่าโซลิดสเตทไดรฟ์อยู่มากพอสมควร และด้วยปัจจัยในด้านการพัฒนาของเทคโนโลยีทำให้ข้อมูลดิจิทัลที่มาจากแหล่งต่าง ๆ มีขนาดใหญ่ขึ้น มีรายละเอียดมากขึ้น ส่งผลให้ต้องใช้พื้นที่ในการจัดเก็บที่มากขึ้นตามไปด้วย ทำให้ความต้องการในการใช้ฮาร์ดดิสก์ไดรฟ์นั้นมีแนวโน้มที่สูงขึ้นอย่างต่อเนื่อง ด้วยเหตุนี้เองอุตสาหกรรมการผลิตฮาร์ดดิสก์ไดรฟ์จึงยังคงเป็นอุตสาหกรรมที่ยังคงได้รับความสนใจและได้รับการพัฒนาอย่างต่อเนื่อง

กระบวนการผลิตฮาร์ดดิสก์ไดรฟ์มีขั้นตอนการประกอบชิ้นส่วนต่าง ๆ ที่มากมาย รวมไปถึงการทดสอบผลิตภัณฑ์ที่มีกระบวนการซับซ้อนและใช้เวลายาวนาน โดยฮาร์ดดิสก์ไดรฟ์หนึ่งยูนิตอาจใช้เวลาในการประกอบชิ้นส่วนรวมไปถึงการทดสอบผลิตภัณฑ์ที่ยาวนานกว่า 3 เดือน ฮาร์ดดิสก์ไดรฟ์ที่สามารถผ่านการทดสอบผลิตภัณฑ์ไปได้จะเรียกว่า Passed Unit และฮาร์ดดิสก์ไดรฟ์ที่ไม่สามารถผ่านการทดสอบผลิตภัณฑ์ได้นั้นจะเรียกว่า Failed Unit การที่มีจำนวน Passed Unit เป็นจำนวนมากจะส่งผลให้ผลผลิตของกระบวนการหรือยีลด์ (Yield) มีมากขึ้นตามไปด้วย เนื่องจากผลผลิตของกระบวนการจะถูกคำนวณมาจากจำนวนของฮาร์ดดิสก์ไดรฟ์ที่ผ่านการทดสอบต่อจำนวนของฮาร์ดดิสก์ไดรฟ์ที่เข้าสู่กระบวนการทดสอบนั้น ๆ ยกตัวอย่างเช่น มีฮาร์ดดิสก์ไดรฟ์เข้าสู่กระบวนการทดสอบ 100 ยูนิตและมีฮาร์ดดิสก์ไดรฟ์ที่ผ่านกระบวนการทดสอบ 80 ยูนิต กรณีนี้จะมียีลด์เท่ากับ 0.8 หรือจะกล่าวได้ว่ากระบวนการทดสอบนี้มีความสามารถในการผลิตฮาร์ดดิสก์ไดรฟ์สำเร็จร้อยละ 80

การคาดการณ์ผลผลิตของกระบวนการ (Yield Prediction) เป็นงานที่มีความสำคัญและส่งผลกระทบโดยตรงต่อการวางแผนงานในหลายภาคส่วน ไม่ว่าจะเป็น การวางแผนการผลิต การวางแผนงานด้านการสั่งซื้อวัตถุดิบ การวางแผนงานในกระบวนการทดสอบผลิตภัณฑ์ การวางแผนส่งมอบผลิตภัณฑ์แก่ลูกค้า รวมไปถึงการตั้งราคาสินค้าเพื่อจัดจำหน่าย เป็นต้น จากความเชื่อมโยง

เหล่านี้ทำให้ความแม่นยำของการคาดการณ์ผลผลิตของกระบวนการมีความสำคัญเป็นอย่างมาก ตัวอย่างเช่น หากการคาดการณ์ผลผลิตโดยใช้ค่ายีลด์ที่สูงเกินกว่าความเป็นจริงแล้ว จะส่งผลกระทบทำให้แผนกวางแผนการจัดการวัตถุดิบ (Material Planner) วางแผนการสั่งซื้อวัตถุดิบไม่เพียงพอต่อความสามารถในการผลิตที่เกิดขึ้นจริง ซึ่งจะส่งผลกระทบต่อเนื่องทำให้ไม่สามารถส่งมอบผลิตภัณฑ์ได้ตรงตามตารางการจัดส่งสินค้าให้แก่ลูกค้า และอาจทำให้ความน่าเชื่อถือของบริษัทลดลงตามไปด้วย ในทางตรงกันข้ามถ้าหากการคาดการณ์ผลผลิตนั้นใช้ค่ายีลด์ที่ต่ำกว่าความเป็นจริง ในกรณีนี้จะส่งผลกระทบให้แผนกวางแผนการจัดการวัตถุดิบสั่งวัตถุดิบมามากเกินกว่าความต้องการในการส่งมอบผลิตภัณฑ์ให้กับลูกค้า ซึ่งจะก่อให้เกิดเหตุการณ์ที่กระแสเงินสด (Cash Flow) ลดลงไปจากการที่มีสินค้าคงคลังมากเกินความจำเป็น นอกจากนี้แล้วประเด็นทางด้านการตั้งราคาสินค้านั้นเป็นสิ่งที่ละเอียดอ่อนและมีความเชื่อมโยงกับความแม่นยำในการคาดการณ์ยีลด์สูงมาก เนื่องจากการที่สามารถคาดการณ์ล่วงหน้าได้ว่าจะมียีลด์ของการผลิตเป็นเท่าใดนั้น จะทำให้สามารถคำนวณต้นทุนได้ชัดเจนมากยิ่งขึ้น จะเห็นได้ว่าความสำคัญของการคาดการณ์ยีลด์ให้แม่นยำนั้นจะเป็นการมุ่งเน้นสนับสนุนในด้านของการช่วยทำให้ผู้บริหารระดับสูงสามารถตัดสินใจได้ดีขึ้น รวมไปถึงการกำหนดทิศทางในอนาคตของบริษัทอีกด้วย โดยนับได้ว่าเป็นความสำคัญในระดับการบริหารธุรกิจ (Business Management Level) ซึ่งจะแตกต่างจากลักษณะงานด้านการทำ Yield Improvement ที่มีความสำคัญเพียงแค่ในระดับปฏิบัติการ (Operating Level) ถึงแม้ว่าจะสามารถวัดผลออกมาเป็นตัวเลขจำนวนเงินได้ก็ตาม

อุปสรรคในการคาดการณ์ผลผลิตที่แม่นยำอาจจะเกิดได้จากหลายปัจจัย ตัวอย่างเช่น ปัญหาจากการส่งมอบวัตถุดิบแต่ละชิ้นส่วนจากผู้ผลิตชิ้นส่วนรายย่อย (Supplier) ไม่เป็นไปตามแผนงาน ปัญหาจากการคำนวณและคาดการณ์ที่ไม่แม่นยำจากตัววิศวกรเองอันเนื่องมาจากการที่มีความยึดติดกับวิธีการเดิมที่ใช้ได้ดีในอดีต และอุปสรรคที่สำคัญที่สุดคือปัญหาทางด้านข้อมูลการผลิต และการทดสอบผลิตภัณฑ์นั้นมีจำนวนมหาศาลซึ่งมีความไม่สมดุลระหว่าง Passed Unit และ Failed Unit อยู่ในระดับที่สูงมาก ด้วยเหตุนี้เองการคัดเลือกคุณลักษณะเพื่อให้มีความเหมาะสมต่อการนำมาใช้ในการคาดการณ์ผลผลิตของกระบวนการนั้นจึงเป็นเรื่องที่นับได้ว่ามีความท้าทายเป็นอย่างยิ่ง

การทำข้อมูลให้สมดุลนั้นนับได้ว่าเป็นการสร้างรากฐานที่ดีให้กับกระบวนการคัดเลือกคุณลักษณะ เนื่องจากความไม่สมดุลของข้อมูลนั้นเป็นหนึ่งในสาเหตุหลักในการลดทอนประสิทธิภาพของการนำอัลกอริทึมการเรียนรู้ด้านการจำแนก (Classification) มาประยุกต์ใช้ในงานด้านการคัดเลือกคุณลักษณะ ซึ่งปัญหาเรื่องประสิทธิภาพของการจำแนกถูกลดทอนลงจากความไม่สมดุลของข้อมูลนั้นสามารถพบเห็นได้จากงานวิจัยมากมาย (กีระชาติ สุขสุทธิ์, 2559; Martin-Diaz

et al., 2016; Sun et al., 2015; Chawla, 2009) โดยประสิทธิภาพของการจำแนกที่ลดลงนั้นเป็นเหตุมาจากโมเดลการเรียนรู้พยายามจำแนกข้อมูลเพื่อให้มีความแม่นยำมากที่สุด ซึ่งการกระทำดังกล่าวจะก่อให้เกิดความโน้มเอียงที่จะทำให้โมเดลที่ได้รับเลือกทำนายคลาสเป้าหมายให้เป็นคลาสส่วนมาก (Majority Class) โดยอาจจะละทิ้งการทำนายคลาสเป้าหมายที่เป็นคลาสส่วนน้อย (Minority Class) ด้วยเหตุนี้เองจึงทำให้โมเดลการจำแนกให้ค่าความสำคัญ (Important Factor) ของแต่ละคุณลักษณะโน้มเอียงตามการทำนายนั้น ซึ่งจะทำให้การคัดเลือกคุณลักษณะมีประสิทธิภาพที่ลดลงตามไปด้วย โดยในชุดข้อมูลที่มีค่าอัตราส่วนความไม่สมดุล (Imbalance Ratio) อยู่ในระดับที่สูงมีโอกาสทำให้เกิดเหตุการณ์ที่โมเดลไม่สามารถคัดเลือกคุณลักษณะได้ เนื่องจากโมเดลการเรียนรู้จะเลือกทำนายข้อมูลทั้งหมดของคลาสเป้าหมายให้เป็นคลาสส่วนมากแทน ซึ่งถูกอธิบายเพิ่มเติมในรูปที่ 1.1 และ รูปที่ 1.2 โดยที่รูปที่ 1.3 และ 1.4 จะเป็นการอธิบายในตัวอย่างข้อมูลที่ได้รับการทำสมดุลข้อมูลเรียบร้อยแล้ว ส่วนรายละเอียดเรื่องความไม่สมดุลของข้อมูลและการจัดการชุดข้อมูลที่ไม่สมดุลนั้นจะถูกกล่าวไว้ในบทที่ 2

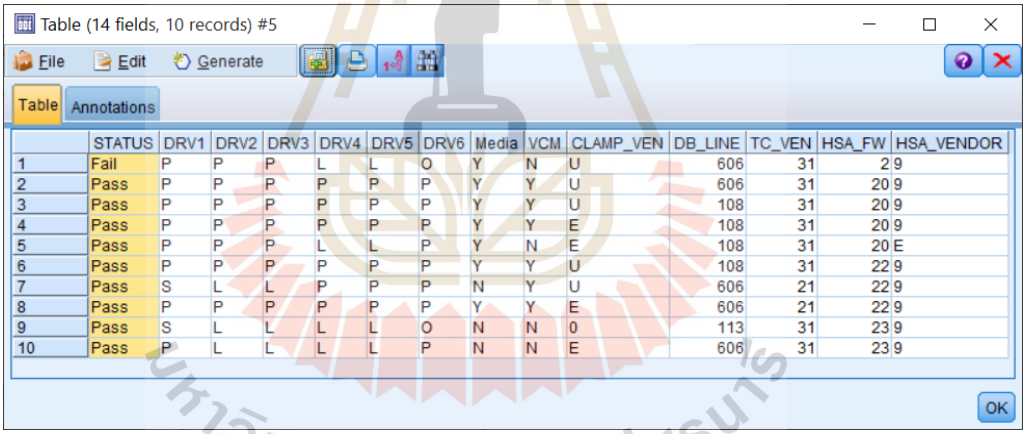


Table (14 fields, 10 records) #5

File Edit Generate

Table Annotations

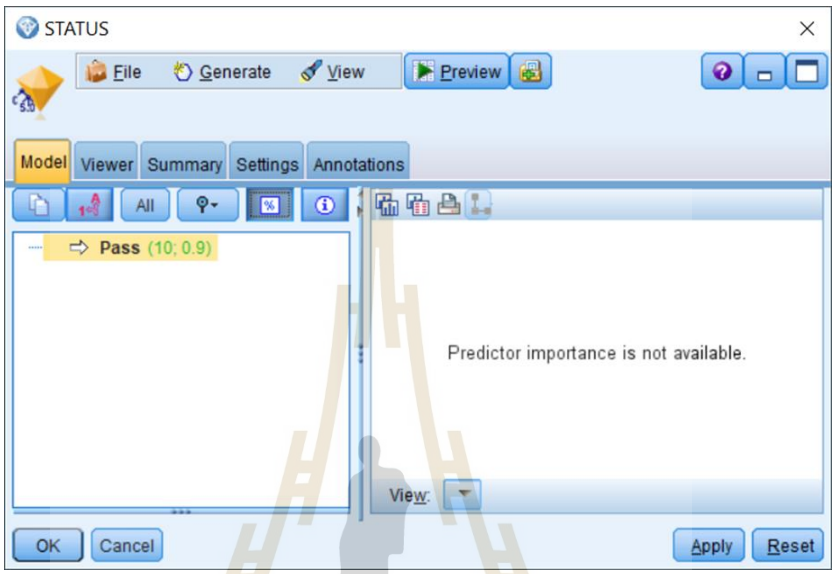
	STATUS	DRV1	DRV2	DRV3	DRV4	DRV5	DRV6	Media	VCM	CLAMP_VEN	DB_LINE	TC_VEN	HSA_FW	HSA_VENDOR
1	Fail	P	P	P	L	L	O	Y	N	U	606	31	29	
2	Pass	P	P	P	P	P	P	Y	Y	U	606	31	209	
3	Pass	P	P	P	P	P	P	Y	Y	U	108	31	209	
4	Pass	P	P	P	P	P	P	Y	Y	E	108	31	209	
5	Pass	P	P	P	L	L	P	Y	N	E	108	31	20	E
6	Pass	P	P	P	P	P	P	Y	Y	U	108	31	229	
7	Pass	S	L	L	P	P	P	N	Y	U	606	21	229	
8	Pass	P	P	P	P	P	P	Y	Y	E	606	21	229	
9	Pass	S	L	L	L	L	O	N	N	0	113	31	239	
10	Pass	P	L	L	L	L	P	N	N	E	606	31	239	

OK

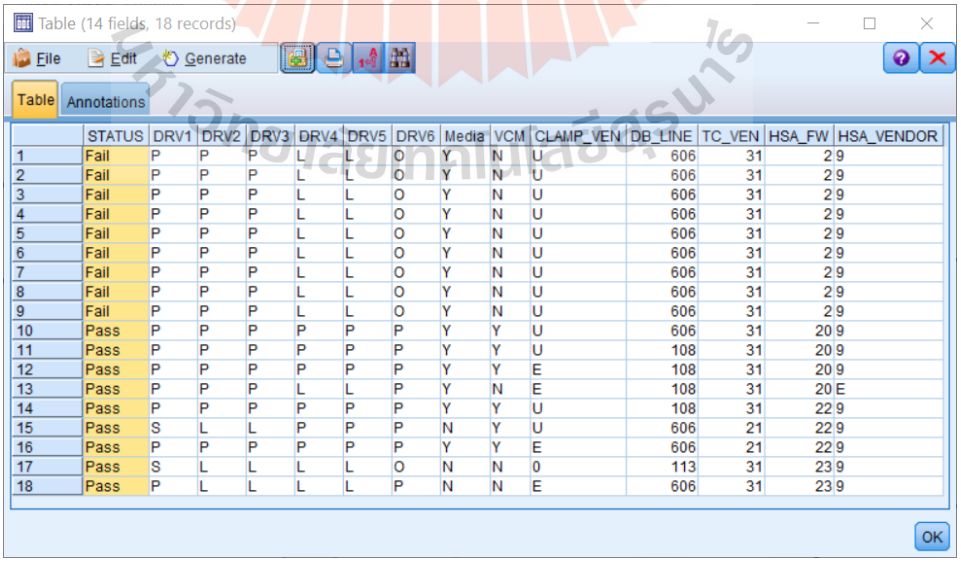
รูปที่ 1.1 ตัวอย่างชุดข้อมูลฮาร์ดดิสก์ไดรฟ์ซึ่งมีระดับความไม่สมดุลเท่ากับ 10

รูปที่ 1.1 แสดงตัวอย่างชุดข้อมูลฮาร์ดดิสก์ไดรฟ์ซึ่งมีระดับความไม่สมดุลเท่ากับ 10 โดยมีจำนวนข้อมูลฮาร์ดดิสก์ไดรฟ์ 10 ยูนิต มีคลาสเป้าหมายคือ STATUS ซึ่งมีคลาสส่วนมากคือ STATUS = Pass จำนวน 9 ยูนิต และคลาสส่วนน้อยคือ STATUS = Fail จำนวน 1 ยูนิต และมีจำนวนคุณลักษณะ 13 คุณลักษณะ โดยชุดข้อมูลตัวอย่างนี้ถูกนำไปสร้างโมเดลการจำแนกด้วยอัลกอริทึม C5.0 ในโปรแกรม IBM SPSS Modeler ผลลัพธ์ที่ได้จากการดำเนินการพบว่าตัวโมเดล

ไม่สามารถทำการหาคุณลักษณะสำคัญที่ส่งผลต่อการทำนายได้ เนื่องจากโมเดลเลือกที่จะทำนายคลาสเป้าหมายของข้อมูลทั้งหมดให้เป็น Pass โดยมีค่าความแม่นยำอยู่ที่ 0.9 ดังที่ได้แสดงไว้ในรูปที่ 1.2



รูปที่ 1.2 โมเดลที่ได้รับจากอัลกอริทึม C5.0 เพื่อใช้ในการคัดเลือกคุณลักษณะเมื่อใช้ข้อมูลฮาร์ดดิสก์ไดรฟ์ซึ่งมีระดับความไม่สมดุลเท่ากับ 10

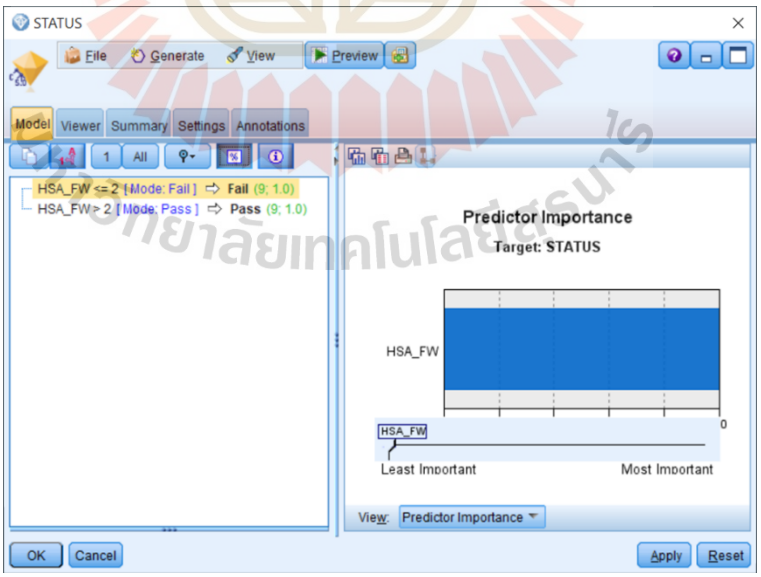


	STATUS	DRV1	DRV2	DRV3	DRV4	DRV5	DRV6	Media	VCM	CLAMP_VEN	DB_LINE	TC_VEN	HSA_FW	HSA_VENDOR
1	Fail	P	P	P	L	L	O	Y	N	U	606	31	29	
2	Fail	P	P	P	L	L	O	Y	N	U	606	31	29	
3	Fail	P	P	P	L	L	O	Y	N	U	606	31	29	
4	Fail	P	P	P	L	L	O	Y	N	U	606	31	29	
5	Fail	P	P	P	L	L	O	Y	N	U	606	31	29	
6	Fail	P	P	P	L	L	O	Y	N	U	606	31	29	
7	Fail	P	P	P	L	L	O	Y	N	U	606	31	29	
8	Fail	P	P	P	L	L	O	Y	N	U	606	31	29	
9	Fail	P	P	P	L	L	O	Y	N	U	606	31	29	
10	Pass	P	P	P	P	P	P	Y	Y	U	606	31	209	
11	Pass	P	P	P	P	P	P	Y	Y	U	108	31	209	
12	Pass	P	P	P	P	P	P	Y	Y	E	108	31	209	
13	Pass	P	P	P	L	L	P	Y	N	E	108	31	20E	
14	Pass	P	P	P	P	P	P	Y	Y	U	108	31	229	
15	Pass	S	L	L	P	P	P	N	Y	U	606	21	229	
16	Pass	P	P	P	P	P	P	Y	Y	E	606	21	229	
17	Pass	S	L	L	L	L	O	N	N	0	113	31	239	
18	Pass	P	L	L	L	L	P	N	N	E	606	31	239	

รูปที่ 1.3 ตัวอย่างชุดข้อมูลฮาร์ดดิสก์ไดรฟ์ซึ่งผ่านกระบวนการทำสมดุลข้อมูล

รูปที่ 1.3 แสดงการนำชุดข้อมูลฮาร์ดดิสก์ไดรฟ์ในรูปที่ 1.1 มาผ่านกระบวนการทำสมดุลข้อมูลด้วยวิธีสุ่มเพิ่มอย่างง่าย โดยจะเป็นการทำซ้ำข้อมูลที่มีค่า STATUS = Fail เพิ่มเข้าไปอีกจำนวน 8 ยูนิต ซึ่งจะทำให้ชุดข้อมูลใหม่นี้มีจำนวนข้อมูลทั้งหมด 18 ยูนิต และมีจำนวนคุณลักษณะ 13 คุณลักษณะ โดยมีจำนวนข้อมูลในคลาสเป้าหมาย STATUS = Pass และ STATUS = Fail มีจำนวนเท่ากันคือ 9 ยูนิต หลังจากนั้นได้นำชุดข้อมูลใหม่นี้ไปสร้างโมเดลการจำแนกด้วยอัลกอริทึม C5.0 เช่นเดียวกับชุดข้อมูลแรก ผลลัพธ์คือโมเดลสามารถใช้คุณลักษณะที่มีอยู่มาใช้ในการจำแนกข้อมูลได้ โดยใช้คุณลักษณะ HSA_FW เพื่อแบ่งข้อมูลทั้งสองคลาสโดยจำแนกข้อมูล Pass จำนวน 9 ยูนิต และข้อมูล Fail จำนวน 9 ยูนิตได้ถูกต้องทั้งหมด จึงมีค่าความแม่นยำของโมเดลอยู่ที่ 1.0 หรือ 100% และคุณลักษณะที่สำคัญจากโมเดลที่ได้รับ คือคุณลักษณะ HSA FW ซึ่งได้แสดงไว้ในรูปที่ 1.4

จากตัวอย่างข้อมูลข้างต้นซึ่งมีระดับความไม่สมดุลอยู่ที่ 10 มีข้อมูลเพียง 10 ยูนิต ยังได้รับผลกระทบจากการไม่ทำสมดุลข้อมูลก่อนเข้าสู่อัลกอริทึมจำแนก เพื่อจะนำโมเดลที่ได้รับไปใช้ในการคัดเลือกคุณลักษณะ ซึ่งข้อมูลจริงที่นำมาใช้ในงานวิจัยนี้มีระดับความไม่สมดุลที่สูงกว่ามากและมีจำนวนข้อมูลที่นำมาใช้มากถึง 10 ล้านยูนิตซึ่งนั่นหมายความว่าจะทำให้มีโอกาสสูงที่โมเดลไม่สามารถหาคุณลักษณะที่สำคัญมาใช้ในการสร้างโมเดลเพื่อจำแนกได้ ดังนั้นผู้วิจัยจึงนำเสนอการทำสมดุลข้อมูลในรูปแบบใหม่ซึ่งถือเป็นอีกหนึ่งกระบวนการที่มีความสำคัญในงานวิจัยนี้



รูปที่ 1.4 โมเดลที่ได้รับจากอัลกอริทึม C5.0 เพื่อใช้ในการคัดเลือกคุณลักษณะเมื่อใช้ข้อมูลฮาร์ดดิสก์ไดรฟ์ซึ่งผ่านการทำสมดุลข้อมูล

ถึงแม้ว่าในหลายปีที่ผ่านมาจะมีการนำองค์ความรู้ทางด้านปัญญาประดิษฐ์ (Artificial Intelligence) การเรียนรู้ของเครื่อง (Machine Learning) และการทำเหมืองข้อมูล (Data Mining) เข้ามาประยุกต์ใช้ในการทำงานในด้านการคาดการณ์ผลผลิต แต่โดยส่วนมากพบว่าจะเป็นการนำไปประยุกต์ใช้ในการคาดการณ์ผลผลิตด้านอุตสาหกรรมการเกษตรรวมไปถึงการแปรรูปผลิตภัณฑ์จากการเกษตร ในส่วนของอุตสาหกรรมที่เกี่ยวกับชิ้นส่วนอิเล็กทรอนิกส์นั้นพบงานวิจัยในปริมาณที่น้อย โดยจะเป็นอุตสาหกรรมการผลิตแผ่นวงจร (Printed Circuit Board) (Lee et al., 2015) อุตสาหกรรมการผลิตแผ่นเวเฟอร์ (Wafer) (Yuan et al., 2011) อุตสาหกรรมการผลิต TFT-LCD (Thin Film Technology Liquid Crystal Display) (Lee and Tsai, 2019) และอุตสาหกรรมการผลิตชิ้นส่วนเซมิคอนดักเตอร์ (Semi-conductor Components) (Chen, 2017; An et al., 2009) ในภาคส่วนของอุตสาหกรรมการผลิตฮาร์ดดิสก์ไดรฟ์นั้นพบว่างานวิจัยส่วนใหญ่จะมุ่งเน้นไปในการทำงานด้านการหาปัจจัยหรือสาเหตุที่ทำให้ผลิตภัณฑ์ไม่ผ่านการทดสอบ (Failure Root Cause Analysis) ซึ่งจากที่ผู้วิจัยได้ทำการค้นคว้างานวิจัยมายังไม่พบงานวิจัยที่ทำการวิจัยหรือค้นคว้าเกี่ยวกับเรื่องการคาดการณ์ผลผลิตในอุตสาหกรรมการผลิตฮาร์ดดิสก์ไดรฟ์โดยตรง

วิธีการดั้งเดิมของการคาดการณ์ผลผลิตที่นิยมใช้งานอยู่ในปัจจุบันนั้น จะเริ่มจากการคัดเลือกคุณลักษณะโดยอาศัยจากความรู้ และประสบการณ์ที่ได้รับการสืบทอดต่อกันมา จากนั้นจึงนำคุณลักษณะเหล่านั้นไปใช้งานเพื่อทำการคาดการณ์ผลผลิต ในส่วนของการคำนวณผลผลิตนั้นจะใช้การรวมกลุ่มข้อมูล (Data Aggregation) ที่อ้างอิงตามระยะเวลาของปฏิทิน ซึ่งการรวมกลุ่มข้อมูลเป็นรายสัปดาห์จะเป็นที่นิยมมากที่สุด จากการศึกษาข้อมูลการผลิตและทดสอบผลิตภัณฑ์ฮาร์ดดิสก์ไดรฟ์ ทางผู้วิจัยพบว่าวิธีการดั้งเดิมนี้มีโอกาสส่งผลเสียในกรณีที่จำนวนข้อมูลของแต่ละสัปดาห์นั้นมีความแปรปรวนสูง รวมไปถึงคุณลักษณะที่ได้รับคัดเลือกมานั้นอาจจะล้าสมัยเกินไปไม่เหมาะสมกับการทำงานในการคาดการณ์ผลผลิตของผลิตภัณฑ์รุ่นใหม่ ซึ่งบ่อยครั้งการคาดการณ์ผลผลิตด้วยวิธีการดั้งเดิมนี้ให้ผลลัพธ์ที่มีความแม่นยำที่ต่ำและไม่คงที่ จึงทำให้เกิดผลกระทบที่ตามมาคือการต้องวางแผนงานแบบเผื่อเหลือเผื่อขาด หรือจำเป็นต้องใส่ปัจจัยด้านความปลอดภัย (Safety Factor) ไว้เป็นค่าที่สูง ซึ่งเป็นการสิ้นเปลืองโดยเปล่าประโยชน์

จากสิ่งที่ได้กล่าวมาข้างต้นทางผู้วิจัยได้เล็งเห็นถึงโอกาสในการเพิ่มประสิทธิภาพในการทำงานทางด้านการคาดการณ์ผลผลิตในอุตสาหกรรมการผลิตฮาร์ดดิสก์ไดรฟ์ โดยการนำความรู้ทางด้านการเรียนรู้ของเครื่อง และแนวคิดในเรื่องของการรวมกลุ่มข้อมูลเข้ามาช่วยและได้ทำการเรียบเรียงไว้เป็นหัวข้อดังต่อไปนี้

1. การจัดการกับปัญหาชุดข้อมูลที่ไม่สมดุล โดยการนำเสนอเทคนิควิธีการใหม่ในการทำสมดุลข้อมูลชื่อว่า DBC-2KAR ซึ่งจะเป็นการนำอัลกอริทึม k-Means Clustering, k-Nearest Neighbors และ Re-Sampling มาช่วยในการจัดการข้อมูล

2. การนำวิธีการทางสถิติและอัลกอริทึมทางการเรียนรู้ของเครื่อง รวมทั้งหมด 7 ชนิด มาใช้ในการคัดเลือกคุณลักษณะ โดยพิจารณาจากผลลัพธ์ของการทดสอบผลิตภัณฑ์ ได้แก่ วิธีการคัดเลือกคุณลักษณะทางสถิติโดยใช้ Chi-Square เป็นเกณฑ์, วิธีการคัดเลือกคุณลักษณะทางสถิติโดยใช้ Information Gain เป็นเกณฑ์, อัลกอริทึมต้นไม้ตัดสินใจ C5.0, อัลกอริทึม Classification and Regression Tree (CART), อัลกอริทึม Support Vectors Machine (SVM), อัลกอริทึม Stepwise Regression และ อัลกอริทึม Genetic Algorithm (GA) โดยจะนำคุณลักษณะที่ได้รับการคัดเลือกจากแต่ละชนิด แยกกันไปใช้สร้างตัวแบบ (หรือโมเดล) ของการคาดการณ์ผลผลิต

3. การนำอัลกอริทึมการเรียนรู้ได้แก่ Multiple Linear Regression (MLR) และ Artificial Neural Networks (ANN) มาประยุกต์ใช้งานในการสร้างโมเดลการคาดการณ์ผลผลิต

4. การนำเสนอเทคนิคใหม่ของการรวมกลุ่มข้อมูล โดยจะเป็นการรวมกลุ่มของข้อมูลในลักษณะของการแบ่งกลุ่มข้อมูลตามจำนวนค่าคงที่ที่ได้กำหนดไว้ แทนที่วิธีการดั้งเดิมที่อ้างอิงการรวมกลุ่มของข้อมูลเพื่อการคำนวณผลผลิตด้วยการแบ่งข้อมูลเป็นรายสัปดาห์

1.2 วัตถุประสงค์ของงานวิจัย

งานวิจัยชิ้นนี้มีวัตถุประสงค์ของการทำงานดังต่อไปนี้

1. เพื่อทำการพัฒนาและเพิ่มประสิทธิภาพในการจัดการปัญหาความไม่สมดุลของข้อมูลด้วยวิธีการ DBC-2KAR

2. เพื่อทำการศึกษาอัลกอริทึมที่มีประสิทธิภาพในการคัดเลือกคุณลักษณะ

3. เพื่อศึกษาการนำคุณลักษณะที่สำคัญมาใช้ในการคาดการณ์ผลผลิต ด้วยการสร้างโมเดลการคาดการณ์ผลผลิตด้วยอัลกอริทึมการเรียนรู้ รวมถึงการเปรียบเทียบประสิทธิภาพของโมเดล

4. เพื่อทำการพัฒนาและปรับปรุงประสิทธิภาพในการสร้างโมเดลการคาดการณ์ผลผลิตเพื่อให้ได้ผลลัพธ์ที่ดีที่สุด โดยศึกษาจากการเปรียบเทียบจากความสัมพันธ์ของ “การเพิ่มค่าคงที่ของจำนวนในการรวมข้อมูล”

1.3 ขอบเขตของงานวิจัย

1. ข้อมูลที่ใช้ในงานวิจัยเป็นชุดข้อมูลจริงของกระบวนการผลิตและทดสอบผลิตภัณฑ์ฮาร์ดดิสก์ไดรฟ์ โดยมีช่วงเวลาการเก็บข้อมูลของกระบวนการผลิตประมาณ 3 ปี

2. อัลกอริทึมที่ใช้สำหรับการทำสมดุลข้อมูล ได้แก่ k-Means Clustering, k-Nearest Neighbors, Over-Sampling และ Under-Sampling

3. อัลกอริทึมและวิธีการทางสถิติที่นำมาใช้ในขั้นตอนการคัดเลือกคุณลักษณะมี 7 ชนิด ได้แก่ Decision Tree-C5 model, CART, SVM, Stepwise Regression, GA, Information Gain และ Chi-Square

4. การประเมินประสิทธิภาพการคัดเลือกคุณลักษณะ จะใช้ค่าน้ำหนักของแต่ละคุณลักษณะที่ส่งผลต่อคลาสเป้าหมายโดยการคัดเลือกจำนวนคุณลักษณะที่นำมาใช้ขึ้นอยู่กับแต่ละวิธีการ นอกเหนือจากนั้นยังมีการวัดค่าความถูกต้องในการจำแนก และเวลาที่ใช้ในการประมวลผล (Computation time) ในวิธีการที่สามารถวัดได้เพื่อใช้ประกอบการพิจารณาเพิ่มเติม

5. อัลกอริทึมที่ใช้ในการเปรียบเทียบเพื่อทำการสร้างโมเดลการคาดการณ์ผลผลิตมี 2 ชนิด ได้แก่ Multiple Linear Regression และ Artificial Neural Networks โดยในส่วนของเกณฑ์การวัดที่ใช้ในการประเมินประสิทธิภาพมี 2 มาตรวัด ได้แก่ Mean Absolute Error (MAE) และ Root Mean Square Error (RMSE)

1.4 ประโยชน์ที่คาดว่าจะได้รับ

จากการศึกษาและพัฒนางานวิจัยนี้ ผู้วิจัยคาดหวังว่ากระบวนการต่าง ๆ ในงานวิจัยนี้ ไม่ว่าจะเป็นการทำสมดุลข้อมูล กระบวนการคัดเลือกคุณลักษณะจากการสร้างตัวแบบโมเดลจากอัลกอริทึมการเรียนรู้ของเครื่อง จะสามารถนำไปใช้งานเพื่อช่วยเพิ่มประสิทธิภาพให้การคาดการณ์ผลผลิตของกระบวนการต่าง ๆ ให้ดียิ่งขึ้น ไม่ว่าจะเป็นในอุตสาหกรรมการผลิตฮาร์ดดิสก์ไดรฟ์ หรือแม้แต่ในอุตสาหกรรมอื่น ๆ นอกเหนือจากนั้นแล้วทางผู้วิจัยคาดหวังเป็นอย่างยิ่งว่าการรวมกลุ่มของข้อมูล ด้วยค่าคงที่จะสามารถนำไปประยุกต์ใช้เพื่อเป็นการเพิ่มประสิทธิภาพในการคาดการณ์ผลผลิตได้อีกทางหนึ่ง รวมไปถึงการเป็นแนวทางในการศึกษาแก่ผู้ที่สนใจเพื่อพัฒนาต่อยอดแนวคิดในการเพิ่มประสิทธิภาพของการคาดการณ์ผลผลิต

บทที่ 2
ปริทัศน์วรรณกรรมและงานวิจัยที่เกี่ยวข้อง

เนื้อหาในบทนี้จะเริ่มต้นด้วยการกล่าวถึงการจำแนกข้อมูล (Data Classification) อธิบายถึงลักษณะของชุดข้อมูลไม่สมดุล (Imbalanced Data) และการแก้ปัญหาในข้อมูลที่ไม่สมดุล ซึ่งการแก้ปัญหาชุดข้อมูลไม่สมดุลนี้จะมีทั้งวิธีการจัดการในระดับข้อมูล (Data-Level Methods) และวิธีการจัดการในระดับอัลกอริทึม (Algorithm-Level Methods) โดยงานวิจัยนี้จะนำอัลกอริทึม k-Means Clustering, k-NN, Oversampling และ Undersampling มาประยุกต์ใช้ในการจัดการความไม่สมดุลของข้อมูล ในส่วนถัดมาจะกล่าวถึงการคัดเลือกคุณลักษณะ (Feature Selection) ด้วยวิธีการที่นิยมใช้ ได้แก่ Decision Tree, Support Vectors Machine, Stepwise Regression, Genetic Algorithms, Information Gain และ Chi-Square ตามด้วยอัลกอริทึมที่นิยมใช้ในการสร้างโมเดลการเรียนรู้การคาดการณ์ผลผลิต ได้แก่ Multiple Linear Regression และ Artificial Neural Networks ต่อด้วยวิธีการประเมินประสิทธิภาพที่ใช้ในงานวิจัยนี้ คือ MAE และ RMSE ส่วนสุดท้ายในบทนี้จะกล่าวถึงงานวิจัยที่เกี่ยวข้อง

2.1 การจำแนกข้อมูล (Data Classification)

การจำแนกข้อมูลเป็นงานที่ได้รับความนิยมในงานทางด้านการทำเหมืองข้อมูล (Data Mining) และการเรียนรู้ของเครื่อง (Machine Learning) เนื่องจากเป็นโจทย์ที่พบเจอได้บ่อยและเกี่ยวข้องกับชีวิตประจำวันของมนุษย์ไม่มากก็น้อย ตัวอย่างเช่น เมื่อมีลูกค้าไปขอกู้ยืมเงินจากธนาคารแห่งหนึ่ง ทางธนาคารจะรู้ได้อย่างไรว่าบุคคลนี้จะเป็นผู้กู้ที่ดีหรือเป็นผู้กู้ที่ไม่ดี และควรจะพิจารณาให้กู้ยืมหรือไม่ ซึ่งธนาคารเองก็จะต้องมีการศึกษาข้อมูลจากผู้กู้ในอดีตรวมไปถึงประวัติการชำระหนี้ของผู้กู้เหล่านั้น โดยสิ่งสำคัญจะต้องมีข้อมูลมากเพียงพอที่จะสร้างเป็นโมเดลเพื่อทำนายรูปแบบของผู้กู้ที่ควรให้กู้ยืมเงิน และผู้กู้แบบใดที่ไม่ควรเสี่ยงที่จะให้กู้ยืมเงิน ซึ่งถ้าหากผู้กู้มีคุณสมบัติหลายอย่างตรงตามโมเดลของผู้กู้ที่ดี ธนาคารก็จะอนุมัติเงินกู้ยืมได้ง่ายและรวดเร็วมากยิ่งขึ้น จากตัวอย่างที่กล่าวมาข้างต้น การจำแนกข้อมูล คือ กระบวนการสร้างโมเดลเพื่อหาความสัมพันธ์ระหว่างข้อมูลนำเข้าและคลาสเป้าหมายของข้อมูลนั้น ๆ โดยจุดประสงค์เพื่อใช้ในการจำแนกประเภท หรือจำแนกคลาสเป้าหมายจากข้อมูลที่มีอยู่เดิม เพื่อที่จะนำไปพยากรณ์ข้อมูลใหม่ที่เกิดขึ้นในอนาคต (Tang at el., 2014; Sun at el., 2007; Sun at el., 2009)

2.2 ข้อมูลไม่สมดุล (Imbalanced Data)

ข้อมูลไม่สมดุล คือข้อมูลที่มีสมาชิกในแต่ละกลุ่มไม่เท่ากัน ซึ่งสามารถพบได้ทั่วไปในชีวิตประจำวัน (He and Garcia, 2008) ตัวอย่างเช่น ข้อมูลทางด้านการแพทย์ที่มีข้อมูลของผู้ป่วยเป็นโรคมะเร็งน้อยกว่าผู้ป่วยที่ไม่เป็นโรคมะเร็ง ข้อมูลทางด้านธนาคารที่มีข้อมูลลูกค้าบัตรเครดิตซึ่งมีการใช้จ่ายแบบปกติมากกว่าข้อมูลลูกค้าที่ใช้จ่ายไม่ปกติเพราะถูกโจรกรรมข้อมูลบัตรเครดิต และข้อมูลทางด้านอุตสาหกรรมการผลิตที่มีข้อมูลของผลิตภัณฑ์ที่ดีมากกว่าผลิตภัณฑ์ที่เสียหายจากการผลิต เป็นต้น

หากชุดข้อมูลใดมีจำนวนสมาชิกในแต่ละกลุ่มที่มีความต่างกันของจำนวนสมาชิกมาก ๆ ก็จะยิ่งทำให้ข้อมูลนั้นเกิดความไม่สมดุลของข้อมูลมากตามไปด้วย (Liu et al., 2008) โดยสามารถแสดงได้ในรูปแบบอัตราส่วนหรือระดับของความไม่สมดุลของข้อมูล (Imbalanced Ratio) คำนวณได้จากสมการที่ 2.1

$$Imbalanced\ Ratio = \frac{\text{จำนวนสมาชิกในคลาสส่วนมาก}}{\text{จำนวนสมาชิกในคลาสส่วนน้อย}} \tag{2.1}$$

สมมติมีชุดข้อมูลผู้ป่วยโรคร้ายแรงอยู่ 1 ชุดข้อมูล ซึ่งมีอยู่ 2 คลาส คือ คลาสผู้ป่วยที่เป็นโรคมะเร็ง จำนวน 5 คน และคลาสผู้ป่วยที่ไม่เป็นมะเร็ง จำนวน 1000 คน จากสมการที่ 2.1 สามารถคำนวณระดับความไม่สมดุลของชุดข้อมูลนี้ได้เป็น

$$\frac{\text{จำนวนสมาชิกในคลาสของคนที่ไม่เป็นโรคมะเร็ง}}{\text{จำนวนสมาชิกในคลาสของคนที่เป็นโรคมะเร็ง}} = \frac{1000}{5} = 200$$

ดังนั้นระดับความไม่สมดุลของข้อมูลชุดนี้เท่ากับ 200 ซึ่งหากระดับของความไม่สมดุลมีค่าสูงมาก ๆ จะถือได้ว่าข้อมูลดังกล่าวเป็นข้อมูลที่มีระดับความไม่สมดุลอยู่ในเกณฑ์ที่สูง (Zong et al., 2013) ระดับความไม่สมดุลที่สูงมาก ๆ นี้เองจะส่งผลทำให้การแก้ปัญหาในการจำแนกข้อมูลมีความยากมากยิ่งขึ้น

2.3 การแก้ปัญหาข้อมูลไม่สมดุล

การแก้ปัญหาข้อมูลไม่สมดุลนั้นได้รับความสนใจในวงกว้างและมีวิธีการที่หลากหลายโดยสามารถแบ่งวิธีออกเป็น 2 กลุ่ม ดังนี้

2.3.1 Data-Level Methods

Data-Level Methods คือ วิธีการที่มุ่งเน้นไปที่การจัดการกับข้อมูลโดยตรง ภายใต้แนวคิดที่เรียบง่าย คือ เมื่อเจอปัญหาข้อมูลไม่สมดุลก็ต้องหาวิธีการที่ทำให้ข้อมูลเหล่านั้นมีสมดุลให้มากขึ้นเพื่อให้เหมาะสมกับการใช้อัลกอริทึมพื้นฐานในการจำแนกประเภทข้อมูลต่อไป ซึ่งส่วนใหญ่จะอยู่หลังจากกระบวนการนี้เสร็จสิ้นแล้ว ซึ่งเรียกว่า เทคนิคการการสุ่มตัวอย่างซ้ำ (Resampling) คือ เทคนิคการเลือกตัวอย่างจากคลาสหนึ่งให้มากกว่าหรือน้อยกว่าอีกคลาสหนึ่ง เพื่อชดเชยความไม่สมดุลที่มีอยู่ในชุดข้อมูล (Kotsiantis et al., 2006; Japkowicz, 2000) แบ่งได้เป็น 3 ประเภท คือ

1. Under-sampling คือวิธีการสุ่มด้วยการลดจำนวนสมาชิกในคลาสที่มีจำนวนสมาชิกมากกว่า เรียกว่า คลาสส่วนมาก (Majority Class) ให้มีจำนวนใกล้เคียง หรือมีความสมดุลกับคลาสที่มีจำนวนสมาชิกน้อยกว่า เรียกว่า คลาสส่วนน้อย (Minority Class) (Chawla, 2009) ซึ่งมีหลายเทคนิคที่ได้รับความนิยม ได้แก่ Random Under Sampling (RUS) และ Tomek links เป็นต้น โดยข้อเสียของวิธีการนี้คืออาจทำให้พลาดข้อมูลที่สำคัญบางอย่างไป

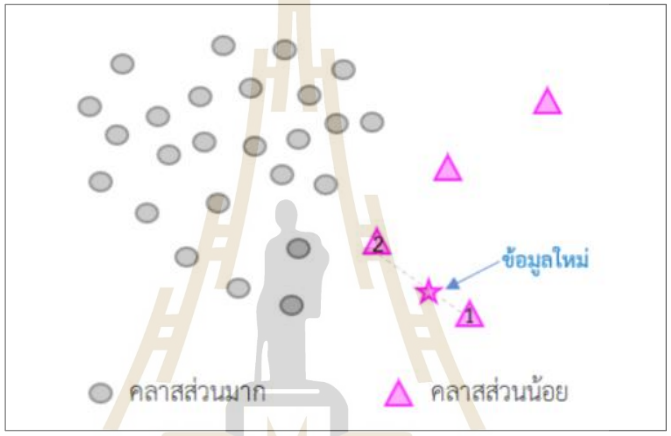
RUS คือ วิธีการสุ่มด้วยการลดข้อมูลรูปแบบหนึ่ง ซึ่งถือได้ว่าเป็นวิธีการสุ่มด้วยการลดข้อมูลที่ง่ายที่สุดมีพื้นฐานมาจากสถิติ หลักการคือสุ่มข้อมูลในคลาสส่วนมากออกมา จากนั้นลบข้อมูลชุดนี้ทิ้งไป และทำแบบนี้ไปจนครบจำนวนข้อมูลที่ต้องการลด

2. Over-sampling คือวิธีการสุ่มด้วยการเพิ่มจำนวนสมาชิกในคลาสส่วนน้อยให้มีจำนวนข้อมูลที่สมดุลกับข้อมูลคลาสส่วนมาก (Barandela et al., 2004) ซึ่งมี 2 เทคนิคที่ได้รับความนิยมคือ Random Oversampling (ROS) และ Synthetic Majority Oversampling Technique (SMOTE) ข้อเสียของวิธีการนี้คืออาจทำให้สร้างข้อมูลที่ไม่เป็นประโยชน์เพิ่มขึ้นมา และเสี่ยงต่อการทำให้เกิดเหตุการณ์ Overfitting ขึ้นได้

ROS คือวิธีการสุ่มด้วยการเพิ่มข้อมูลรูปแบบหนึ่ง ซึ่งถือได้ว่าเป็นวิธีการสุ่มด้วยการเพิ่มข้อมูลที่ง่ายที่สุด โดยอาศัยพื้นฐานจากหลักการทางสถิติ คือสุ่มข้อมูลในคลาสส่วนน้อยขึ้นมา หลังจากนั้นคัดลอกข้อมูลชุดดังกล่าวเพิ่มขึ้นมา โดยทำแบบนี้ไปจนครบจำนวนที่ต้องการจะเพิ่มข้อมูล

SMOTE คือวิธีการสังเคราะห์ข้อมูลเพิ่มซึ่งได้รับความนิยม และถือว่ามีประสิทธิภาพมากวิธีการหนึ่ง ซึ่งถูกพัฒนาขึ้นโดย Chawala (Chawla et al., 2004) คือ จะสุ่มสร้างข้อมูลสังเคราะห์จากคลาสส่วนน้อยโดยเริ่มจากการสุ่มตัวอย่างจากคลาสส่วนน้อยมาหนึ่งตัว จากนั้นพิจารณาข้อมูลที่ใกล้ที่สุด และทำการสังเคราะห์ข้อมูลใหม่ขึ้นมาระหว่างข้อมูลที่สุ่มไว้กับข้อมูลที่ใกล้ที่สุด ดังรูปที่ 2.1 แสดงตัวอย่างการสุ่มสร้างข้อมูลใหม่ด้วยเทคนิค SMOTE จะเห็นได้

ว่าชุดข้อมูลวงกลมสีเทาคือคลาสส่วนมาก และชุดข้อมูลสามเหลี่ยมสีชมพูคือคลาสส่วนน้อย โดยเทคนิค SMOTE จะสังเคราะห์ข้อมูลเพิ่มในคลาสส่วนน้อย สมมติให้ข้อมูลสามเหลี่ยมหมายเลข 1 เป็นข้อมูลที่ถูกสุ่มขึ้นมาเพื่อใช้เป็นตัวตั้งต้นในการหาข้อมูลที่ใกล้ที่สุดมา 1 ตัว ในที่นี้คือสามเหลี่ยมหมายเลข 2 หลังจากนั้นจะทำการสุ่มสังเคราะห์ข้อมูลใหม่ขึ้นมาระหว่างข้อมูลสองตัวนี้ จึงได้ข้อมูลดาวสีชมพูเป็นข้อมูลที่สังเคราะห์ขึ้นตัวแรก จากนั้นจะทำกระบวนการนี้ซ้ำไปมาจนกว่าจะครบจำนวนข้อมูลที่ต้องการสุ่มเพิ่ม ซึ่งวิธีการนี้ยังสามารถหลีกเลี่ยงการเกิดปัญหา Overfitting ได้อีกด้วย (Han et al., 2005)



รูปที่ 2.1 ตัวอย่างแสดงการสุ่มสร้างข้อมูลใหม่ด้วยเทคนิค SMOTE

3. Hybrid คือวิธีที่นำการสุ่มด้วยการลดและการสุ่มด้วยการเพิ่มข้อมูลมาทำงานร่วมกันเพื่อให้เกิดประสิทธิภาพมากยิ่งขึ้น (Ramentol et al., 2012) เช่น วิธีการหาค่ากลางขึ้นมา จากนั้นจะทำการสุ่มลดจำนวนสมาชิกในคลาสส่วนมาก และสุ่มเพิ่มจำนวนสมาชิกในคลาสส่วนน้อย เป็นต้น

2.3.2 Algorithm-Level Methods

วิธีการนี้จะมุ่งเน้นไปที่การปรับค่าน้ำหนัก (Weight) บางอย่างภายในตัวอัลกอริทึม โดยจุดประสงค์เพื่อต้องการลดบทบาทของสมาชิกในคลาสส่วนมากในชุดข้อมูลที่ไม่สมดุลลงมา และเพิ่มบทบาทให้กับสมาชิกในคลาสส่วนน้อยในระหว่างกระบวนการเรียนรู้ ซึ่งจะส่งผลให้โมเดลที่ได้มาหลังจากกระบวนการนี้มีประสิทธิภาพที่สูงขึ้นกว่าโมเดลโดยทั่วไป (Krawczyk, 2016) วิธีการที่ได้รับความนิยมอย่างมาก คือ Cost-Sensitive Learning (Weiss et al., 2007; Thai-Nghe et al., 2010; López et al., 2012) โดยหลักการคือการกำหนดค่าใช้จ่ายในการจำแนกผิด

(Misclassification Cost) ให้แก่อัลกอริทึมตามสัดส่วนที่สูงขึ้นของระดับความไม่สมดุล กล่าวคือ หากระดับความไม่สมดุลอยู่ในเกณฑ์ที่สูง แสดงว่าข้อมูลในคลาสส่วนน้อยมีน้อยกว่าคลาสส่วนมากในปริมาณที่มาก ดังนั้นค่าใช้จ่ายในการจำแนกผิดของคลาสส่วนน้อยก็ควรจะปรับให้สูงตามไปด้วย การกระทำเช่นนี้จะทำให้อัลกอริทึมที่กำหนดค่าใช้จ่ายในการจำแนกผิดของแต่ละคลาสไว้ตั้งแต่แรกนั้นจะมีความระมัดระวังในการจำแนกผิดในข้อมูลคลาสส่วนน้อยเพิ่มมากกว่าเดิม กระบวนการดังกล่าวส่งผลให้โมเดลที่ได้มีประสิทธิภาพในการจำแนกคลาสไม่สมดุลได้ดียิ่งขึ้น

		ค่าที่โมเดลทำนาย	
		คลาสส่วนมาก	คลาสส่วนน้อย
ค่าจริง	คลาสส่วนมาก	0	1
	คลาสส่วนน้อย	100	0

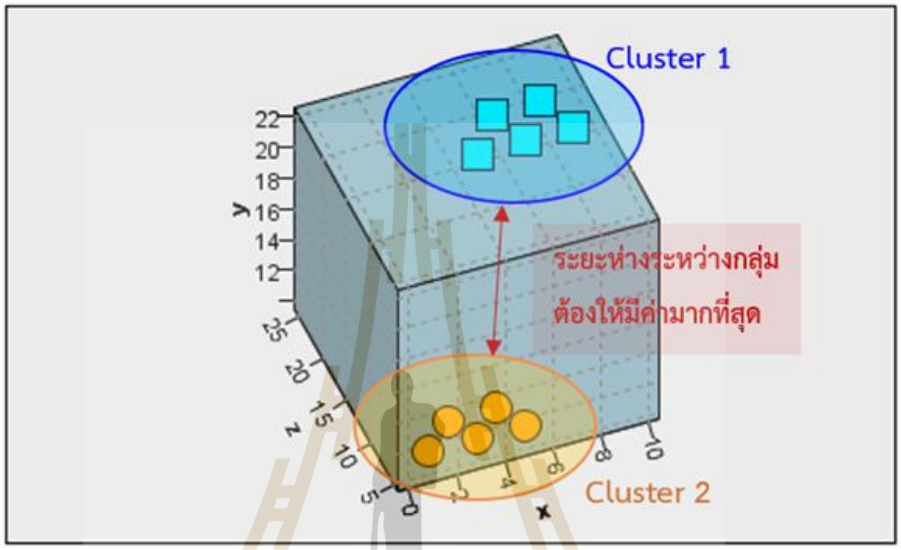
รูปที่ 2.2 ตัวอย่างการกำหนดค่าใช้จ่ายในการจำแนกผิดให้แก่อัลกอริทึมกับข้อมูลที่มีสองคลาส

จากรูปที่ 2.2 เป็นตัวอย่างการกำหนดค่าใช้จ่ายในการจำแนกผิดให้แก่อัลกอริทึมในชุดข้อมูลที่มีสองคลาส สามารถอธิบายได้ว่าหากข้อมูลจริงและข้อมูลที่โมเดลทำนายนั้น ถูกต้องตรงกัน จะถือว่าไม่มีค่าใช้จ่ายในการจำแนกผิด (จากรูปถูกกำหนดไว้เป็นเลข 0) แต่ถ้าหากค่าจริงของข้อมูลเป็นคลาสส่วนมากแล้วโมเดลทำนายให้เป็นคลาสส่วนน้อย ความเสียหายในส่วนนี้ยังถือว่าไม่ร้ายแรงมากนักเพราะคลาสส่วนมากนั้นมีจำนวนมากอยู่แล้วทำนายผิดไปบ้างก็ไม่ถือว่ากระทบมาก จึงใส่ค่าความเสียหายในการจำแนกผิดเพียงเล็กน้อยได้ (ในรูปจึงใส่ค่าความเสียหายไว้ที่ตัวเลข 1) ส่วนในกรณีที่ข้อมูลจริงเป็นคลาสส่วนน้อยแต่โมเดลทำนายให้เป็นคลาสส่วนมาก ซึ่งเป็นกรณีที่ไม่สมควรเกิดขึ้นมากที่สุด เพราะคลาสส่วนน้อยมีจำนวนน้อยอยู่แล้ว ถ้าหากจำแนกผิดอีกก็จะถือว่ามีค่าความเสียหายในการจำแนกผิดมากกว่ากรณีอื่น (ในรูปจึงใส่ค่าความเสียหายไว้ที่ตัวเลข 100) จุดประสงค์เพื่อให้โมเดลระวังการทำนายผิดในส่วนนี้ให้มากที่สุดนั่นเอง

2.4 k-Means Clustering

อัลกอริทึม k-Means Clustering นี้อยู่ในหมวดหมู่ของการเรียนรู้แบบไม่มีผู้ฝึกสอน (Unsupervised Learning) ซึ่งนั่นหมายถึงการที่ไม่จำเป็นต้องบอกว่าข้อมูลชุดนี้เป็นข้อมูลคลาสใด เนื่องจากไม่ใช่อัลกอริทึมในการจำแนกประเภท (Lin et al., 2017; Le, 2013; Kanungo et al., 2002)

เพียงระบุว่าต้องการให้จัดกลุ่มทั้งหมดกี่กลุ่ม และหลังจากนั้นเป็นหน้าที่ของอัลกอริทึมที่จะไปจัดกลุ่มข้อมูลมาให้โดยอัตโนมัติ หลักการทำงานของ k-Means Clustering คือการจัดข้อมูลที่มีความคล้ายคลึงกันไว้ในกลุ่มหรือคลัสเตอร์เดียวกัน ส่วนคลัสเตอร์ที่ต่างกันก็จะพยายามทำให้ข้อมูลที่อยู่ในคลัสเตอร์เหล่านั้นมีความแตกต่างกันมากที่สุดเท่าที่จะทำได้ (Liang et al., 2012) ดังรูปที่ 2.3



รูปที่ 2.3 ตัวอย่างการจัดกลุ่มข้อมูลด้วยอัลกอริทึม k-Means Clustering เมื่อ k มีค่าเท่ากับ 2

การพิจารณาผลลัพธ์ที่ได้จากการจัดกลุ่มด้วย k-Means Clustering ว่าในรายละเอียดของแต่ละคลัสเตอร์มีสิ่งใดเป็นความคล้ายคลึงกัน หรือมีสิ่งใดใกล้เคียงกันนั้น เพื่อการตีความที่ถูกต้องและรัดกุมมากยิ่งขึ้น ในบางครั้งต้องอาศัยผู้เชี่ยวชาญในสายงานนั้น ๆ มาร่วมพิจารณาก่อนจะนำผลลัพธ์ที่ได้จากการจัดกลุ่มไปใช้หรืออธิบายผู้ที่มีอำนาจในการตัดสินใจต่อไป และถือว่าเป็นความท้าทายอีกหนึ่งอย่างของโมเดลในกลุ่มการเรียนรู้แบบไม่มีผู้ฝึกสอน โดยขั้นตอนการทำงานของ k-Means Clustering (Basu et al., 2008; Wu et al., 2012) สามารถอธิบายได้ดังนี้

1. สุ่มข้อมูลให้เท่ากับจำนวน k ที่ระบุไว้ เพื่อใช้เป็นตัวแทนของแต่ละคลัสเตอร์

2. หาระยะห่างของสมาชิกทุกตัวเทียบกับตัวแทนที่ได้ในขั้นตอนที่ 1 ด้วยมาตรวัดระยะทาง

3. จัดกลุ่มสมาชิกให้อยู่ในคลัสเตอร์ที่เหมาะสม โดยพิจารณาจากระยะทางระหว่างสมาชิกตัวใดมีระยะทางใกล้กับตัวแทนของคลัสเตอร์ใดที่สุด ก็จะกำหนดให้เป็นสมาชิกของคลัสเตอร์นั้น ๆ

4. เมื่อจัดกลุ่มให้สมาชิกทุกตัวแล้ว หาตัวแทนในแต่ละคลัสเตอร์ใหม่ ด้วยการคำนวณค่ากึ่งกลางหรือค่าเฉลี่ย โดยคำนวณจากสมาชิกทุกตัวในคลัสเตอร์นั้น ๆ

5. ทำซ้ำในขั้นตอนที่ 3 และ 4 ตามลำดับ จนกว่าตัวแทนในแต่ละคลัสเตอร์จะไม่เปลี่ยนแปลง หรือสมาชิกในแต่ละคลัสเตอร์ไม่มีการเปลี่ยนแปลง หรือครบจำนวนรอบที่กำหนดไว้จึงจะหยุดการทำงานของอัลกอริทึม

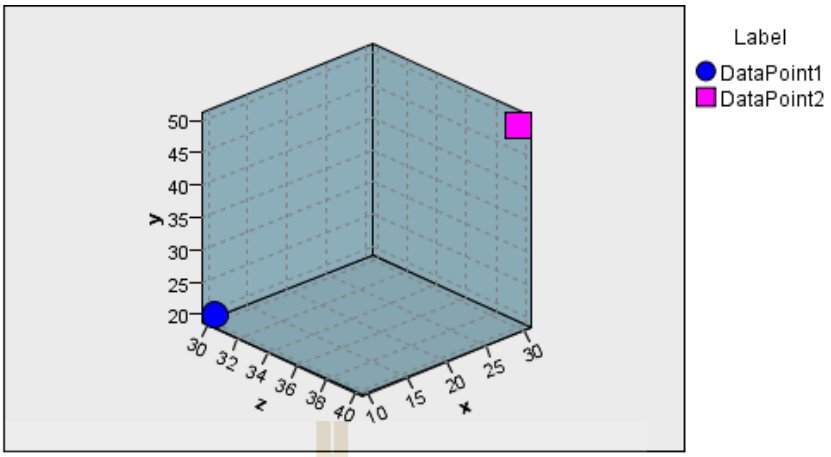
มาตรวัดระยะทางที่ถูกกล่าวถึงในขั้นตอนการทำงาน k-Means Clustering (ขั้นตอนที่ 2) เพื่อใช้หาระยะทางระหว่างข้อมูลกับตัวแทนของแต่ละคลัสเตอร์นั้นมีหลายมาตรวัด แต่มีหลักการที่เหมือนกัน ได้แก่ ระยะทางที่คำนวณได้ต้องไม่ติดลบ ถ้าหากข้อมูลนั้นเป็นตำแหน่งเดียวกัน (เหมือนกัน) ระยะทางต้องเป็นศูนย์ และการคำนวณระยะทางจากจุด a ไปยังจุด b ต้องเท่ากับจุด b ไปยังจุด a (ไปกลับต้องเท่ากัน) มาตรวัดระยะทางที่เป็นที่นิยม ได้แก่ Euclidean Distance (Su and Chou, 2001) และ Manhattan Distance (Singh et al., 2013) ซึ่งมีวิธีการคำนวณดังต่อไปนี้

การคำนวณระยะห่างระหว่างข้อมูลด้วยวิธีแบบ Euclidean Distance (Elkan, 2003) เป็นการคำนวณระยะห่างอย่างง่ายซึ่งกระทำในแนวเส้นตรงโดยไม่มีค่าติดลบ หากต้องการคำนวณระยะห่างระหว่าง 2 ข้อมูล คือ a และ b ด้วยวิธีแบบยูคลิด จะสามารถคำนวณได้ดังสมการที่ 2.2

$$Distance1(a,b) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + \cdots + (a_n - b_n)^2} \qquad (2.2)$$

การคำนวณระยะห่างระหว่างข้อมูลด้วยวิธีแบบ Manhattan Distance เป็นมาตรวัดที่มีการนำค่าที่คำนวณได้ในหนึ่งเรคคอร์ดมารวมกัน ซึ่งเป็นผลรวมด้านประกอบของ Euclidean Distance (Kapil and Chawla, 2016; Thakare and Bagal, 2015) ถ้าหากต้องการคำนวณระยะห่างระหว่าง 2 ข้อมูล คือ a และ b ด้วยวิธีแบบแมนฮัตตันจะสามารถคำนวณได้ดังสมการที่ 2.3

$$Distance2(a,b) = |a_1 - b_1| + |a_2 - b_2| + \ldots + |a_n - b_n| \qquad (2.3)$$



รูปที่ 2.4 ตัวอย่างข้อมูลที่ใช้คำนวณมาตรวัดระยะทาง

จากรูปที่ 2.4 เป็นการแสดงตัวอย่างของข้อมูลที่ 1 และข้อมูลที่ 2 ในรูปแบบสามมิติ (x, y, z) โดยข้อมูลที่ 1 อยู่ที่ตำแหน่ง x=10, y=20, z=30 และข้อมูลที่ 2 อยู่ที่ตำแหน่ง x=30, y=50, z=40 สามารถนำมาหาระยะทางแบบยูคลิดและแมนฮัตตันดังตารางที่ 2.1

ตารางที่ 2.1 ตัวอย่างการคำนวณมาตรวัดระยะทางระหว่างข้อมูลแบบยูคลิดและแบบแมนฮัตตัน

ระยะทางแบบยูคลิด	ระยะทางแบบแมนฮัตตัน
Distance 1 (ข้อมูล1, ข้อมูล2) $= \sqrt{(10-30)^2 + (20-50)^2 + (30-40)^2}$ $= 37.4165$	Distance 2 (ข้อมูล1, ข้อมูล2) $= \|10-30\| + \|20-50\| + \|30-40\|$ $= 60$

จากตารางที่ 2.1 จะเห็นได้ว่าจากข้อมูลชุดเดียวกัน หากคำนวณระยะทางแบบยูคลิดจะได้ระยะห่าง คือ 37.42 หน่วย แต่หากใช้การคำนวณระยะแบบแมนฮัตตันจะได้ระยะห่าง คือ 60 หน่วย ซึ่งการคำนวณระยะห่างที่แตกต่างกันนี้เองจะส่งผลต่ออัลกอริทึมที่ต้องใช้หลังจากการคำนวณนี้ด้วยเช่นกัน

2.5 k-Nearest Neighbors

เคเนียร์เรสเนเบอร์ คืออัลกอริทึมที่ได้รับความนิยมเพื่อใช้ในการจำแนกข้อมูล เนื่องจากมีวิธีการทำงานที่ง่ายไม่ซับซ้อนและยังให้ผลลัพธ์ที่ดี (Fukunage and Narendra, 1975) บางครั้งอัลกอริทึมนี้ก็ถูกเรียกว่า Lazy Learning เนื่องจากไม่มีการสร้างโมเดลเตรียมไว้ล่วงหน้า หากมี

ข้อมูลใหม่ที่ต้องการให้อัลกอริทึมนี้จำแนกประเภทเพิ่มเข้ามา ก็อาศัยเพียงพิจารณาข้อมูลที่มีอยู่เดิม ซึ่งอยู่ใกล้กับข้อมูลใหม่ที่สุดจำนวน k ตัว ว่าข้อมูลเหล่านั้นอยู่ในประเภทใดบ้าง แล้วจึงจัดการให้ข้อมูลใหม่ที่กำลังพิจารณาอยู่นั้น ถูกจัดอยู่ในประเภทเดียวกับเสียงข้างมากของ k ที่กำหนดไว้ ซึ่งมีขั้นตอนการทำงานดังต่อไปนี้ (Zhang and Zhou, 2005)

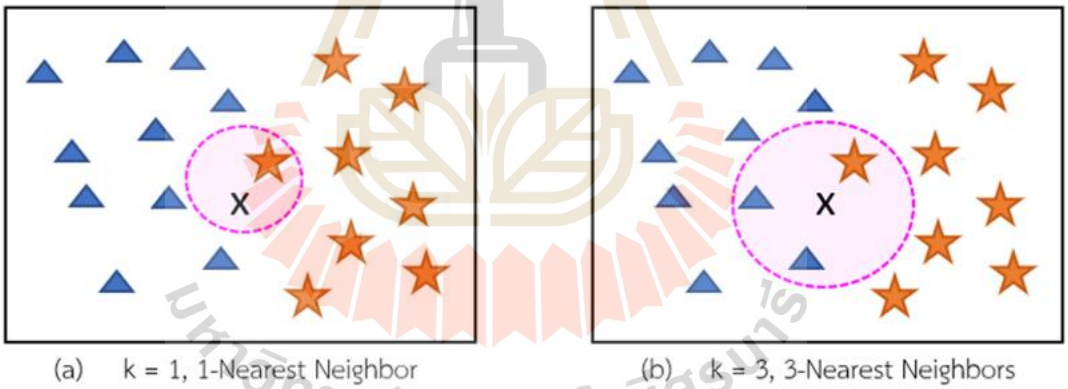
1. กำหนดค่า k ซึ่งเป็นจำนวนข้อมูลที่ใกล้ที่สุดที่จะนำมาพิจารณา (นิยมกำหนดเป็นเลขคี่)

2. คำนวณระยะห่างระหว่างข้อมูลที่ต้องการพิจารณาหรือต้องการจำแนก กับข้อมูลเดิมที่มีอยู่ทั้งหมด ผ่านฟังก์ชันการคำนวณระยะทาง (มาตรวัดระยะที่เป็นที่นิยม ได้แก่ Euclidean Distance และ Manhattan Distance)

3. จัดเรียงลำดับตามระยะห่างของข้อมูล และเลือกพิจารณาจากข้อมูลที่ใกล้ที่สุด k ตัว (ตามค่า k ที่ถูกกำหนดไว้แล้วในขั้นตอนที่ 1)

4. รวมผลเฉลยคลาสเป้าหมายของข้อมูลที่ใกล้ที่สุด

5. จัดประเภทให้แก่ข้อมูลที่กำลังพิจารณา ให้เป็นไปตามคลาสเป้าหมายของข้อมูลที่ใกล้ที่สุดที่มีจำนวนมากที่สุด



รูปที่ 2.5 ตัวอย่างการจำแนกข้อมูลด้วยอัลกอริทึมเคเนียร์เรสเนเบอร์โดยค่า k = 1 และ k = 3

จากรูปที่ 2.5 แสดงให้เห็นถึงการจำแนกประเภทข้อมูลด้วยอัลกอริทึมเคเนียร์เรสเนเบอร์ จากตัวอย่างมีประชากรอยู่ 2 ประเภท คือ คลาสสามเหลี่ยมสีน้ำเงิน และคลาสดาวสีส้ม เมื่อมีข้อมูลใหม่เข้ามาซึ่งในที่นี้ คือ X หากมีการกำหนดค่า k ให้เป็น 1 (รูปซ้ายมือ) จะพิจารณาข้อมูลตัวที่ใกล้ที่สุด 1 ตัว ซึ่งพบว่าเป็นคลาสดาวสีส้ม ดังนั้น ข้อมูลใหม่ X ก็จะถูกจัดอยู่ในประเภทดาวสีส้มด้วย และจากโจทย์เดียวกันแต่กำหนดให้ k เป็น 3 (รูปขวามือ) อัลกอริทึมก็จะพิจารณาข้อมูลตัวที่ใกล้ที่สุด 3 ตัว พบว่ามีคลาสดาวสีส้ม 1 ตัว คลาสสามเหลี่ยมสีน้ำเงิน 2 ตัว ซึ่งพบว่าคลาสสามเหลี่ยมสี

น้ำเงินของข้อมูลตัวที่ใกล้ที่สุดมีจำนวนมากกว่า ดังนั้น ข้อมูลใหม่ X ก็จะถูกจัดให้อยู่ในคลาสสามเหลี่ยมสีน้ำเงินด้วย

จะเห็นได้ว่าโจทย์ปัญหาเดียวกัน แต่ถ้ากำหนดค่า k ให้แตกต่างกันก็ส่งผลถึงการจำแนกข้อมูลใหม่ที่แตกต่างกันด้วย ดังนั้นจึงมีงานวิจัยอีกหลายงานที่กล่าวถึงการหาค่า k ที่เหมาะสมและการใช้มาตรวัดระยะที่เหมาะสมกับข้อมูลเฉพาะด้านอีกมากมาย

2.6 การคัดเลือกคุณลักษณะ (Feature Selection)

การคัดเลือกคุณลักษณะหรือการเลือกฟีเจอร์ เป็นเทคนิคหนึ่งที่นิยมนำมาใช้ในงานทางด้านการเรียนรู้ของเครื่องและสถิติ (Turabieh et al., 2019; Suppers et al., 2018; Liu and Zhou 2017) เพื่อลดมิติของข้อมูลทำให้สามารถลดเวลาและลดการใช้งานทรัพยากรของเครื่องลงได้ โดยหลักการคือการเลือกคุณลักษณะที่มีความสำคัญออกมาจากคุณลักษณะทั้งหมดที่มีอยู่ในชุดข้อมูลนั้น เพื่อเป็นประโยชน์ต่อการนำไปใช้งานในกระบวนการถัดไป เช่น การจำแนกประเภท หรือการคาดการณ์ผลลัพธ์ เป็นต้น

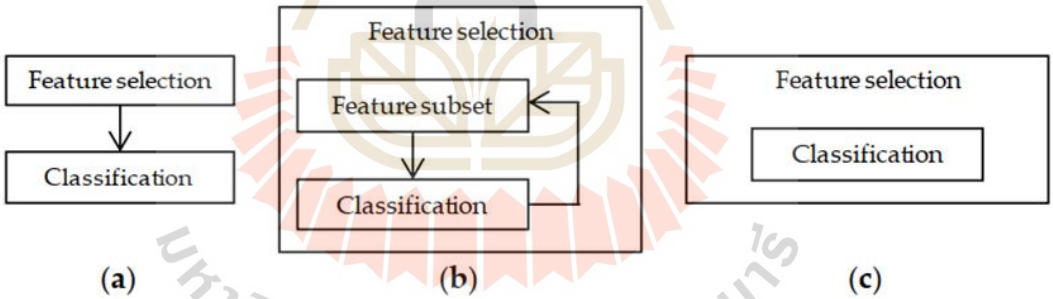
การคัดเลือกคุณลักษณะสามารถแบ่งออกได้เป็น 3 ประเภท ได้แก่ 1) Filter Method 2) Wrapper Method และ 3) Embedded Method ตามที่แสดงในรูปที่ 2.6 โดยมีรายละเอียดดังต่อไปนี้

1. Filter Method เป็นกระบวนการเลือกคุณลักษณะด้วยวิธีฟิลเตอร์ มีหลักการคือจะพิจารณาว่าแต่ละคุณลักษณะเป็นอิสระจากกัน และพยายามหาคุณลักษณะซึ่งมีผลกระทบหรือมีความสำคัญต่อคุณลักษณะเอาต์พุตให้ได้มากที่สุด จากนั้นคุณลักษณะทั้งหลายเหล่านั้นจะถูกนำมาจัดลำดับความสำคัญจากมากไปน้อย โดยจะมีการกำหนดระดับ Threshold หากคุณลักษณะใดมีความสำคัญไม่ถึง Threshold ที่กำหนดไว้ก็จะถูกตัดทิ้งไปเพราะถือว่าไม่มีความสำคัญ ส่วนเกณฑ์ที่นิยมนำมาใช้ในการคัดเลือกคุณลักษณะแบบฟิลเตอร์ ได้แก่ Chi-Square และ Information Gain ซึ่งข้อดีของ Filter Method คือ สามารถคำนวณได้ง่าย รวดเร็ว และลดปัญหาการเกิด Overfitting ได้ ส่วนข้อจำกัดของวิธีการนี้ คือ เมื่อนำไปใช้ในกระบวนการต่อจากนี้ เช่น การนำคุณลักษณะที่ได้รับเลือกมาสร้างโมเดลการจำแนกประเภท ผลลัพธ์ที่ได้อาจจะไม่ได้ดีเสมอไป เนื่องจากตอนเลือกคุณลักษณะมีการพิจารณาทีละคุณลักษณะ แต่กรณีการสร้างโมเดลต้องใช้หลายคุณลักษณะร่วมกัน ซึ่งอาจทำให้ได้ผลลัพธ์ที่ดีหรือไม่ดีก็ได้ และวิธีการนี้ไม่ค่อยนิยมนำมาใช้กับข้อมูลจำนวนมาก

2. Wrapper Method เป็นกระบวนการคัดเลือกคุณลักษณะที่มีแนวคิดมาจากความต้องการที่จะลดปัญหาของ Filter Method ในเรื่องของการพิจารณาคุณลักษณะได้เพียงคราวละหนึ่งคุณลักษณะเท่านั้น โดยหลักการของ Wrapper Method คือจะนำหลาย ๆ คุณลักษณะมาพิจารณาร่วมกัน โดยอยู่ในรูปแบบของเซ็ตของคุณลักษณะ (Feature Set) จากนั้นพยายามหาเซ็ตคุณลักษณะ

ที่มีความสำคัญต่อคุณลักษณะเอาต์พุตมากที่สุด โดยการนำเซ็ตคุณลักษณะเหล่านั้นมาสร้างโมเดลการเรียนรู้จากนั้นจะใช้การประเมินประสิทธิภาพของโมเดล เป็นตัววัดประสิทธิภาพคุณลักษณะในแต่ละเซ็ต หลังจากนั้นทำการเลือกคุณลักษณะทั้งเซ็ตนั้นมาใช้งานในกระบวนการต่อไป โดย Wrapper Method สามารถทำได้ 2 แบบ ได้แก่ 1) Forward Stepwise คือ การค่อย ๆ นำคุณลักษณะมาใส่เพิ่มทีละคุณลักษณะจนได้เซ็ตที่ดีที่สุด และ 2) Backward Stepwise คือ การนำทุกคุณลักษณะที่มีมาอยู่ในเซ็ตแล้วค่อย ๆ นำคุณลักษณะออกไปจากเซ็ตทีละคุณลักษณะจนได้เซ็ตที่ดีที่สุด ข้อดีของ Wrapper Method คือ เป็นวิธีที่ง่ายและมีประสิทธิภาพสูง ข้อเสีย คือ ใช้เวลามากกว่า Filter Method และอาจเกิดปัญหา Overfitting ได้

3. Embedded Method เป็นกระบวนการเลือกคุณลักษณะด้วยวิธีฝังตัว มีแนวคิดจากความต้องการที่จะลดข้อจำกัดของ Filter Method และ Wrapper Method โดย Embedded Method มีหลักการคือนำการคัดเลือกคุณลักษณะให้มาเป็นส่วนหนึ่งของโมเดลในการจำแนกประเภทด้วย ข้อดีของวิธีการนี้ คือ ใช้เวลาน้อยกว่าวิธีการ Wrapper Method และมีประสิทธิภาพในการจำแนกมากกว่า Filter Method ส่วนข้อเสียของวิธีการ Embedded Method คือ คุณลักษณะที่ได้มีความเฉพาะเจาะจงมาก ทำให้อาจเกิดปัญหา Overfitting ได้

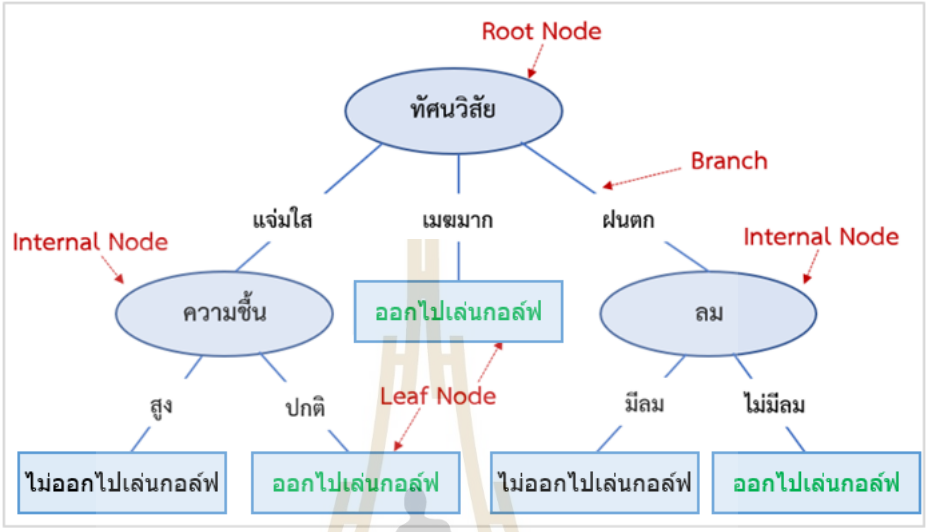


รูปที่ 2.6 การคัดเลือกคุณลักษณะทั้ง 3 ประเภท (a) Filter Method, (b) Wrapper Method, and (c) Embedded Method (Reference : Suppers et al., 2018)

2.7 ต้นไม้ตัดสินใจ (Decision Tree)

ต้นไม้ตัดสินใจ คืออัลกอริทึมที่รู้จักกันอย่างกว้างขวางในการจำแนกประเภทข้อมูล ด้วยเหตุผลเด่นทางด้านการตีความผลลัพธ์ของการจำแนกซึ่งง่ายต่อการทำความเข้าใจ ต้นไม้ตัดสินใจจะเป็นอัลกอริทึมของการหารูปแบบของสิ่งที่ต้องการจำแนกโดยที่โครงสร้างโมเดลของการจำแนกข้อมูลนั้นอยู่ในรูปแบบของลำดับชั้น (Hierarchy) (Safavian and Landgrebe, 1991) ซึ่งอ้างอิงโครงสร้างมาจากต้นไม้ ประกอบด้วยโหนด (Node) และกิ่ง (Branch) โดยโหนดสามารถแบ่ง

ออกเป็น 3 ประเภท ได้แก่ โหนดราก (Root Node) โหนดภายใน (Internal Node) และโหนดใบ (Leaf node) ดังรูปที่ 2.7



รูปที่ 2.7 ตัวอย่างโครงสร้างต้นไม้ตัดสินใจ

จากรูปที่ 2.7 แสดงตัวอย่างโครงสร้างต้นไม้ตัดสินใจโดยที่โหนดรากและโหนดภายใน คือ คุณลักษณะ ที่อัลกอริทึมเลือกมาใช้เพื่อช่วยในการสร้างต้นไม้ตัดสินใจ (Srivastava et al., 1999) โหนดราก คือ คุณลักษณะแรกที่อัลกอริทึมเลือกมาใช้สร้างต้นไม้ เมื่อเลือกคุณลักษณะแรกได้แล้ว ลำดับต่อไปจะเป็นกิ่งที่มาเชื่อมต่อกับโหนดรากนั้น จำนวนกิ่งที่มาเชื่อมต่อจะเท่ากับค่าที่เป็นไปได้ทั้งหมดในคุณลักษณะที่โหนดรากนั้นมี (ซึ่งในแต่ละกิ่งจะมีระบุไว้) เชื่อมต่อกันลงมาเป็นลำดับชั้น และหากกิ่งใดมีสมาชิกทั้งหมดเป็นคลาสเดียวกันแล้ว กิ่งนั้น ๆ จะถูกเชื่อมต่อกับโหนดใบเพื่อบอกว่าเป็นคลาสเป้าหมายประเภทใด หรือกล่าวได้ว่าโหนดใบโหนดนี้ได้สิ้นสุดแล้ว เพราะจะไม่มีกิ่งหรือโหนดใดมาต่อกับโหนดใบได้อีก ตรงกันข้ามหากกิ่งที่ต่อมาจากโหนดรากยังไม่สามารถที่จะระบุความเป็นคลาสเดียวกันได้ ก็จะต้องมีการพิจารณาเลือกคุณลักษณะที่เหลือมาสร้างเป็นโหนดภายในต่อจากกิ่งนี้ลงไปอีกชั้น และอาจจะต้องมีกระบวนการทำซ้ำแบบนี้ไปเรื่อย ๆ เพื่อหาเส้นทางจากโหนดรากมายังโหนดใบ หรือจากโหนดรากมายังโหนดภายใน (หนึ่งโหนด หรือมากกว่า) ต่อมายังโหนดใบ เพื่อต้องการที่จะจำแนกข้อมูลทุกตัวให้ได้ แต่ปัญหาก็คือหากปล่อยให้การสร้างต้นไม้ตัดสินใจเป็นไปตามเงื่อนไขดังที่กล่าวมาข้างต้น อาจทำให้เกิดปัญหาหรือทำให้ได้ต้นไม้ตัดสินใจที่ใหญ่และลึกมาก ซึ่งอาจจะใช้เวลานานและเสี่ยงต่อการเกิด Overfitting ซึ่งเป็นผลทำให้ยากต่อการนำไปใช้ประโยชน์ ดังนั้นจึงมีการกำหนดเกณฑ์เพิ่มเติมเพื่อหยุดกระบวนการสร้าง

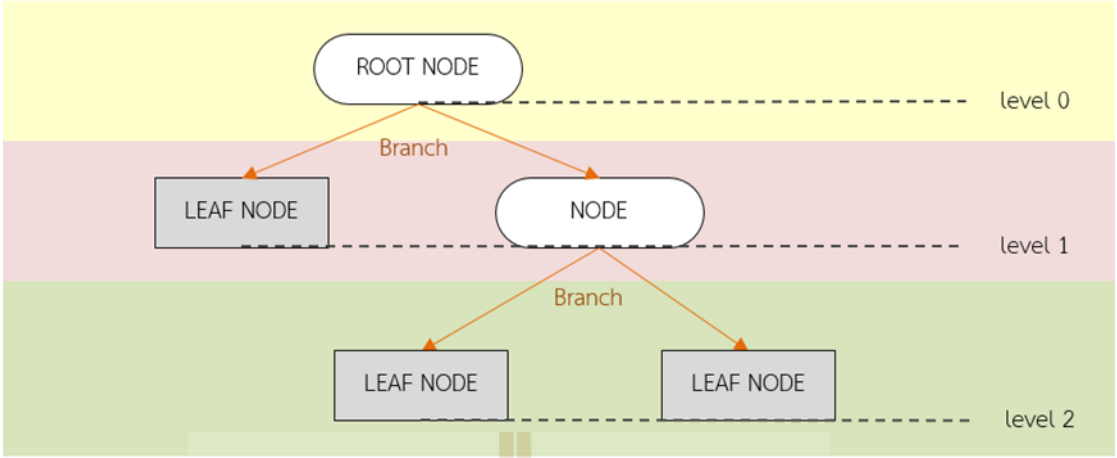
ต้นไม้ เช่น กำหนดความลึกของต้นไม้ตัดสินใจ กำหนดจำนวนสมาชิกในการแตกโหนดใหม่ว่าต้องไม่น้อยกว่าจำนวนเท่าใดจึงจะสามารถเกิดโหนดใหม่ได้เป็นต้น ซึ่งวิธีการเหล่านี้จะทำให้ได้ต้นไม้ตัดสินใจที่ดีมากยิ่งขึ้น นอกจากนี้แล้วต้นไม้ตัดสินใจยังมีเงื่อนไขในการสร้างได้อีกหลายรูปแบบ การที่ผู้ใช้สามารถเลือกใช้อัลกอริทึมที่เหมาะสมที่สุดกับข้อมูลที่มี ก็จะเป็นการเพิ่มประสิทธิภาพในการจำแนกข้อมูลอีกทางหนึ่งด้วย เงื่อนไขที่เป็นที่นิยมในการสร้างต้นไม้ตัดสินใจมีดังตารางที่ 2.2 (Quinlan, 1986; Althuwaynee et al., 2014)

ตารางที่ 2.2 ตัวอย่างชื่ออัลกอริทึมและเงื่อนในการสร้างต้นไม้ตัดสินใจ

เงื่อนไขในการสร้างต้นไม้ตัดสินใจ	**ชื่ออัลกอริทึม**
Information Gain	ID3, C4.5, C5.0
Gini Index	CART
Chi-Square	CHAID
Variance Reduction	CART

ต้นไม้ตัดสินใจนอกจากจะมีอัลกอริทึมให้เลือกใช้หลากหลายแล้ว ข้อดีที่ถือเป็นจุดแข็งก็คือ ความเข้าใจง่าย ไม่เฉพาะแต่โครงสร้างต้นไม้เท่านั้น ผู้ใช้ยังสามารถแปลงต้นไม้ตัดสินใจไปเป็นกฎในรูปแบบ *if…then…* เพื่อต่อยอดนำไปใช้ในโปรแกรมต่าง ๆ หรือฐานข้อมูลอื่น ๆ ก็ยังคงรูปแบบที่เข้าใจได้ง่ายไม่แพ้กัน ด้วยเหตุผลนี้เองที่ทำต้นไม้ตัดสินใจยังคงเป็นที่นิยมและสามารถนำไปพัฒนาร่วมกับแนวคิดอื่น ๆ ได้อย่างมีประสิทธิภาพ

CART (Classification and Regression Trees) เป็นหนึ่งในเทคนิคต้นไม้ตัดสินใจที่สามารถใช้ในการจำแนกกลุ่ม (Classification) และคาดการณ์ (Prediction) เทคนิคนี้ถูกเสนอโดย Breiman ในปี ค.ศ. 1984 (Loh, 2011; Song and Ying, 2015; Breiman et al., 2017) เป็นเทคนิคต้นไม้ตัดสินใจแบบ Binary Tree หรือต้นไม้ตัดสินใจที่แต่ละโหนดมีกิ่งออกไปได้มากที่สุดเพียงสองกิ่ง ประกอบด้วยโหนดรากและโหนดลูก 2 กลุ่ม ที่ไม่มีโหนดร่วมกัน แต่ละกลุ่มจะมีชื่อเรียกว่าต้นไม้ย่อยทางซ้าย (Left Subtree) และต้นไม้ย่อยทางขวา (Right Subtree) CART จะใช้ค่า Gini Index ในการเลือกคุณลักษณะ (Attribute) ในการจำแนกกลุ่มและใช้ผลรวมกำลังสองของความคลาดเคลื่อน (Sum of Squares Error) ในการคาดการณ์ค่าเพื่อสร้างต้นไม้ตัดสินใจ



รูปที่ 2.8 โครงสร้างต้นไม้ตัดสินใจ

รูปที่ 2.8 แสดงให้เห็นถึงโครงสร้างของต้นไม้ตัดสินใจซึ่งประกอบด้วยโหนด คือ คุณลักษณะของข้อมูล โดยโหนดที่เป็นจุดเริ่มต้นของต้นไม้และอยู่บนสุดของต้นไม้จะเรียกว่า โหนดราก (Root Node) ค่าของคุณลักษณะหรือแอททริบิวต์ของโหนดภายในที่แตกกิ่งออกมา เรียกว่า กิ่ง (Branch) และสิ่งที่อยู่ล่างสุดของต้นไม้เรียกว่าโหนดใบ (Leaf Node) มีหน้าที่เพื่อแสดงผลลัพธ์ของต้นไม้ตัดสินใจ

CART สำหรับการจำแนกกลุ่มมีเกณฑ์ในการแตกโหนด โดยจะพิจารณาค่าการแตกโหนดในทุกแอททริบิวต์และทุกกรณีที่เป็นไปได้ของการแตกโหนด จากนั้นเลือกการแตกโหนดในกรณีที่มีค่า การแตกโหนดมากที่สุดมาใช้ในการสร้างต้นไม้ตัดสินใจ โดยจะทำการแตกโหนดไปเรื่อย ๆ จะหยุดการแตกโหนดก็ต่อเมื่อโหนดนั้นมีคำตอบของคลาสเพียงค่าเดียว

เกณฑ์ในการแตกโหนดสามารถคำนวณได้ดังสมการต่อไปนี้ เมื่อ s แทนตัวเลือกที่เราสนใจจะทำการแตกโหนด (Split candidate) และ t แทนโหนดนั้น

$$\Phi(s|t) = 2P_L P_R \times \mathrm{Q}(s|t) \tag{2.4}$$

โดยที่ $\Phi(s|t)$ คือ หน่วยวัดของการแตกโหนด

$\mathrm{Q}(s|t) = \sum_{j=1}^{\#classes} |P(j|t_L) - P(j|t_r)|$

t_L คือ โหนดลูกทางซ้ายของโหนด t

t_R คือ โหนดลูกทางขวาของโหนด t

P_L คือ $\frac{\text{จำนวนเรคคอร์ดที่โหนด } t_L}{\text{จำนวนเรคคอร์ดในชุดข้อมูล}}$

P_R คือ $\frac{\text{จำนวนเรคคอร์ดที่โหนด } t_R}{\text{จำนวนเรคคอร์ดในชุดข้อมูล}}$

$P(j|t_L)$คือ $\frac{\text{จำนวนเรคคอร์ดของคลาส } j \text{ ที่ } t_L}{\text{จำนวนเรคคอร์ดที่ } t}$

$P(j|t_r)$คือ $\frac{\text{จำนวนเรคคอร์ดของคลาส } j \text{ ที่ } t_R}{\text{จำนวนเรคคอร์ดที่ } t}$

ตารางที่ 2.3 ตัวอย่างข้อมูลเพื่อสร้างต้นไม้ตัดสินใจ CART สำหรับการจำแนกกลุ่ม

ข้อมูลที่	Attribute A	Attribute B	Attribute C	Attribute Y
1	กลาง	สูง	75000	ดี
2	ต่ำ	ต่ำ	50000	แย่
3	สูง	กลาง	25000	แย่
4	กลาง	กลาง	50000	ดี
5	ต่ำ	กลาง	100000	ดี
6	สูง	สูง	25000	ดี
7	ต่ำ	ต่ำ	25000	แย่
8	กลาง	กลาง	75000	ดี

ตารางที่ 2.4 การแจกแจงกรณีของการแตกกิ่งในทุกแอททริบิวต์

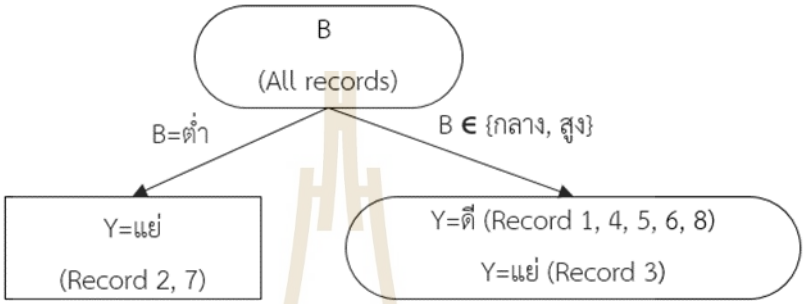
กรณี	โหนดทางซ้าย	โหนดทางขวา
1	A=ต่ำ	Aϵ { กลาง, สูง}
2	A=กลาง	Aϵ {ต่ำ, สูง}
3	A=สูง	Aϵ {ต่ำ, กลาง}
4	B=ต่ำ	Bϵ {กลาง, สูง}
5	B=กลาง	Bϵ {ต่ำ, สูง}
6	B=สูง	Bϵ {ต่ำ, กลาง}
7	C <= 25000	C > 25000
8	C<= 50000	C > 50000
9	C <= 75000	C > 75000

ตัวอย่างในตารางที่ 2.3 มีข้อมูลทั้งหมด 8 ข้อมูลซึ่งข้อมูลในแต่ละแถวแสดงค่าในแอททริบิวต์ A , B และ C ซึ่งใช้เป็นแอททริบิวต์ในการสร้าง CART เพื่อใช้ในการจำแนกหาค่าของแอททริบิวต์ Y ซึ่งเป็นคลาสเป้าหมาย และตารางที่ 2.4 แสดงการแจกแจงกรณีการแตกกิ่งที่เป็นไปได้ของแอททริบิวต์ A, B และ C เป็นจำนวนทั้งสิ้น 9 กรณี โดยแอททริบิวต์ A และ B จะมีค่าที่เป็นไปได้ 3 แบบ ได้แก่ สูง กลาง และต่ำ ในส่วนของแอททริบิวต์ C เนื่องจากข้อมูลเป็นตัวเลขจึงต้องทำการสร้างเงื่อนไขเพื่อแบ่งข้อมูลตัวเลขให้เป็นลักษณะของช่วงข้อมูลซึ่งในกรณีศึกษานี้จะมี 3 ช่วงข้อมูลด้วยกัน ได้แก่ C <=25000, 25000 < C <=50000 และ 50000 < C <=75,000

ตารางที่ 2.5 การคำนวณการแตกกิ่งของต้นไม้ CART รอบที่ 1

กรณี	P_L	P_R	$P(j\|t_L)$	$P(j\|t_R)$	$2P_LP_R$	$Q(s\|t)$	$\Phi(s\|t)$
1	$\frac{3}{8}=0.375$	$\frac{5}{8}=0.625$	ดี: $\frac{1}{3}=0.333$ แย่: $\frac{2}{3}=0.667$	ดี: $\frac{4}{5}=0.8$ แย่: $\frac{1}{5}=0.2$	0.46875	0.934	0.4378
2	$\frac{3}{8}=0.375$	$\frac{5}{8}=0.625$	ดี: $\frac{3}{3}=1$ แย่: $\frac{0}{3}=0$	ดี: $\frac{2}{5}=0.4$ แย่: $\frac{3}{5}=0.6$	0.46875	1.2	0.5625
3	$\frac{2}{8}=0.25$	$\frac{6}{8}=0.75$	ดี: $\frac{1}{2}=0.5$ แย่: $\frac{1}{2}=0.5$	ดี: $\frac{4}{6}=0.667$ แย่: $\frac{2}{6}=0.333$	0.375	0.334	0.1253
4*	$\frac{2}{8}=0.25$	$\frac{6}{8}=0.75$	ดี: $\frac{0}{2}=0$ แย่: $\frac{2}{2}=1$	ดี: $\frac{5}{6}=0.833$ แย่: $\frac{1}{6}=0.167$	0.375	1.667	0.6248
5	$\frac{4}{8}=0.5$	$\frac{4}{8}=0.5$	ดี: $\frac{3}{4}=0.75$ แย่: $\frac{1}{4}=0.25$	ดี: $\frac{2}{4}=0.5$ แย่: $\frac{2}{4}=0.5$	0.5	0.5	0.25
6	$\frac{2}{8}=0.25$	$\frac{6}{8}=0.75$	ดี: $\frac{2}{2}=1$ แย่: $\frac{0}{2}=0$	ดี: $\frac{3}{6}=0.5$ แย่: $\frac{3}{6}=0.5$	0.375	1	0.375
7	$\frac{3}{8}=0.375$	$\frac{5}{8}=0.625$	ดี: $\frac{1}{3}=0.333$ แย่: $\frac{2}{3}=0.667$	ดี: $\frac{4}{5}=0.8$ แย่: $\frac{1}{5}=0.2$	0.46875	0.934	0.4378
8	$\frac{5}{8}=0.625$	$\frac{3}{8}=0.375$	ดี: $\frac{2}{5}=0.4$ แย่: $\frac{3}{5}=0.6$	ดี: $\frac{3}{3}=1$ แย่: $\frac{0}{3}=0$	0.46875	1.2	0.5625
9	$\frac{7}{8}=0.875$	$\frac{1}{8}=0.125$	ดี: $\frac{4}{7}=0.571$ แย่: $\frac{3}{7}=0.429$	ดี: $\frac{1}{1}=1$ แย่: $\frac{0}{1}=0$	0.21875	0.858	0.1877

จากตารางที่ 2.4 ซึ่งแจกแจงกรณีความน่าจะเป็นของการคัดเลือกโหนดที่จะเป็นโหนดรากและเกณฑ์ของการแตกโหนด การคัดเลือกทำได้โดยการคำนวนค่าเกณฑ์ในการแตกโหนด Φ แสดงในตารางที่ 2.5 ซึ่งจากตารางจะเห็นว่าหน่วยวัดของการแตกโหนดมีค่าสูงสุดอยู่ที่ $\Phi(s|t)=0.6248$ ของกรณีที่ 4* ดังนั้น จึงกำหนดแอททริบิวต์ B เป็นโหนดราก จะได้รูปร่างของต้นไม้ตัดสินใจจากการแตกโหนดในครั้งที่ 1 ดังที่ได้แสดงในรูป 2.9



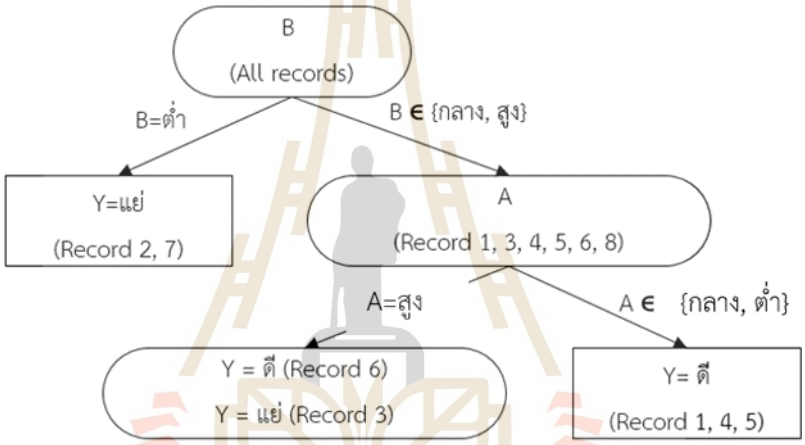
รูปที่ 2.9 ลักษณะโครงสร้างของ CART สำหรับการจำแนกกลุ่มของการคำนวณรอบที่ 1

ตารางที่ 2.6 แสดงการคำนวณการแตกกิ่งของต้นไม้ CART รอบที่ 2

กรณี	P_L	P_R	$P(j\|t_L)$	$P(j\|t_R)$	$2P_LP_R$	$Q(s\|t)$	$\Phi(s\|t)$
1	$\frac{1}{6}=0.167$	$\frac{5}{6}=0.833$	ดี: $\frac{1}{1}=1$ แย่: $\frac{0}{1}=0$	ดี: $\frac{4}{5}=0.8$ แย่: $\frac{1}{5}=0.2$	0.2782	0.4	0.1112
2	$\frac{3}{6}=0.5$	$\frac{3}{6}=0.5$	ดี: $\frac{3}{3}=1$ แย่: $\frac{0}{3}=0$	ดี: $\frac{2}{3}=0.667$ แย่: $\frac{1}{3}=0.333$	0.5	0.666	0.3333
3*	$\frac{2}{6}=0.333$	$\frac{4}{6}=0.667$	ดี: $\frac{1}{2}=0.5$ แย่: $\frac{1}{2}=0.5$	ดี: $\frac{4}{4}=1$ แย่: $\frac{0}{4}=0$	0.4444	1	0.4444
5	$\frac{4}{6}=0.667$	$\frac{2}{6}=0.333$	ดี: $\frac{3}{4}=0.75$ แย่: $\frac{1}{4}=0.25$	ดี: $\frac{2}{2}=1$ แย่: $\frac{0}{2}=0$	0.4444	0.5	0.2222
6	$\frac{2}{6}=0.333$	$\frac{4}{6}=0.667$	ดี: $\frac{2}{2}=1$ แย่: $\frac{0}{2}=0$	ดี: $\frac{3}{4}=0.75$ แย่: $\frac{1}{4}=0.25$	0.4444	0.5	0.2222
7*	$\frac{2}{6}=0.333$	$\frac{4}{6}=0.667$	ดี: $\frac{0}{2}=0$ แย่: $\frac{2}{2}=1$	ดี: $\frac{4}{4}=1$ แย่: $\frac{0}{4}=0$	0.4444	1	0.4444
8	$\frac{3}{6}=0.5$	$\frac{3}{6}=0.5$	ดี: $\frac{2}{3}=0.667$ แย่: $\frac{1}{3}=0.333$	ดี: $\frac{3}{3}=1$ แย่: $\frac{0}{3}=0$	0.5	0.6666	0.3333
9	$\frac{5}{6}=0.833$	$\frac{1}{8}=0.167$	ดี: $\frac{4}{5}=0.8$ แย่: $\frac{1}{5}=0.2$	ดี: $\frac{1}{1}=1$ แย่: $\frac{0}{1}=0$	0.2787	0.4	0.1112

จากรูปที่ 2.9 จะเห็นว่าโหนดทางด้านซ้ายที่ B = ต่ำ มีค่าของคลาส Y = แย่ เพียงค่าเดียว ดังนั้นโหนดนี้จึงไม่ต้องทำการแตกกิ่งต่อ แต่ในส่วนของโหนดด้านขวา จำเป็นต้องทำการแตกกิ่งออกไปจนข้อมูลในแต่ละโหนดมีค่าคลาสเหมือนกันทั้งหมด การแตกโหนดด้านขวานั้นจะทำการพิจารณาข้อมูลที่ 1, 3, 4, 5, 6, 8 เท่านั้น และจะพิจารณาการแจกแจงกรณีของการแตกโหนดดังที่แสดงในตารางที่ 2.6 โดยมีการยกเว้นกรณีที่ 4 เพราะได้ถูกใช้ไปในรอบที่ 1 แล้ว

จากตารางที่ 2.6 การสร้างต้นไม้รอบที่ 2 จะเห็นว่าหน่วยวัดของการแตกโหนดมีค่าสูงที่สุด $\Phi(s|t)$=0.444 ที่กรณี 3* และ 7* ในที่นี้จะสุ่มเลือกกรณีที่ 3 เพื่อมาแสดงตัวอย่างการแตกต้นไม้ตัดสินใจซึ่งจะได้ต้นไม้ดังรูปที่ 2.10



รูปที่ 2.10 ลักษณะโครงสร้างของ CART สำหรับการจำแนกกลุ่มของการคำนวณรอบที่ 2

จากรูปที่ 2.10 จะเห็นว่าโหนดทางด้านขวาของการแตกของโหนด A มีค่าของคลาส Y = ดี เพียงค่าเดียว ดังนั้น โหนดนี้จึงไม่จำเป็นต้องทำการแตกกิ่งต่อ แต่โหนดด้านซ้ายจะต้องทำการแตกกิ่งออกไปจนข้อมูลในแต่ละโหนดมีคลาสคำตอบเดียวกัน การแตกโหนดด้านนั้นจะทำการพิจารณาข้อมูลที่ 3 และ 6 เท่านั้นแต่จะพิจารณากรณีของการแตกโหนดแสดงในตารางที่ 2.6 ยกเว้นกรณีที่ 3, 4 ทำการคำนวณค่าเกณฑ์ในการแตกโหนดต่อไปเรื่อย ๆ จนไม่มีโหนดใดสามารถแตกออกได้อีกจึงจะทำการหยุดการสร้างแบบจำลองต้นไม้ตัดสินใจ

จากการสร้างต้นไม้ตัดสินใจตามตัวอย่างข้างต้น ในการแตกโหนดรอบที่ 2 จะเห็นว่าค่าการแตกโหนดมีค่าเท่ากันสองกรณี แสดงว่าข้อมูลนี้สามารถสร้างต้นไม้ได้หลายแบบ และการที่จะเลือกต้นไม้ที่ดีที่สุดนั้นสามารถทำด้วยการนำต้นไม้ที่เป็นไปได้ทั้งหมดมาทดสอบด้วยข้อมูลทดสอบเพื่อหาเกณฑ์มาวัดประสิทธิภาพของต้นไม้

CART สำหรับคาดการณ์ หรือทำนายข้อมูลมีเกณฑ์ในการแตกโหนดโดยใช้ผลรวมกำลังสองของความคลาดเคลื่อน (Sum of Squared Errors: SSE) ดังสมการต่อไปนี้

$$R(t) = \frac{1}{N}\sum_n (y_n - \hat{y}(t))^2 \quad (2.5)$$

โดยที่ $R(t)$ คือ ค่าผลรวมกำลังสองของความคลาดเคลื่อนของต้นไม้

$\hat{y}(t)$ $= \frac{1}{N(t)}\sum_{x_n \in t} y_n$

$\hat{y}(t)$ คือ ค่าการทำนายที่ปรากฏที่โหนดใบ t

y_n คือ ค่าตัวแปรตามซึ่งเป็นค่าที่แท้จริง

N คือ จำนวนข้อมูลทั้งหมดในต้นไม้

ขั้นตอนพื้นฐานสำหรับการสร้างต้นไม้ตัดสินใจ CART สำหรับการพยากรณ์ข้อมูลมีดังนี้

คำนวณค่า $R(t)$ และ $\hat{y}(t)$ ของข้อมูลทั้งหมด

ทำการหากรณีที่เป็นไปได้ในการแตกโหนดแล้วคำนวณค่า $R(t)$ และ $\hat{y}(t)$ ซึ่งการแตกโหนดที่ดีที่สุดคือ เลือกค่า $\Delta R(s,t)$ ที่มีค่ามากที่สุด

$$\Delta R(s,t) = R(t) - R(t_L) - R(t_R) \quad (2.6)$$

$R(t)$ คือ ค่า **SSE** ของโหนด t

$R(t_L)$ คือ ค่า **SSE** ของโหนดลูกทางซ้ายของโหนด t

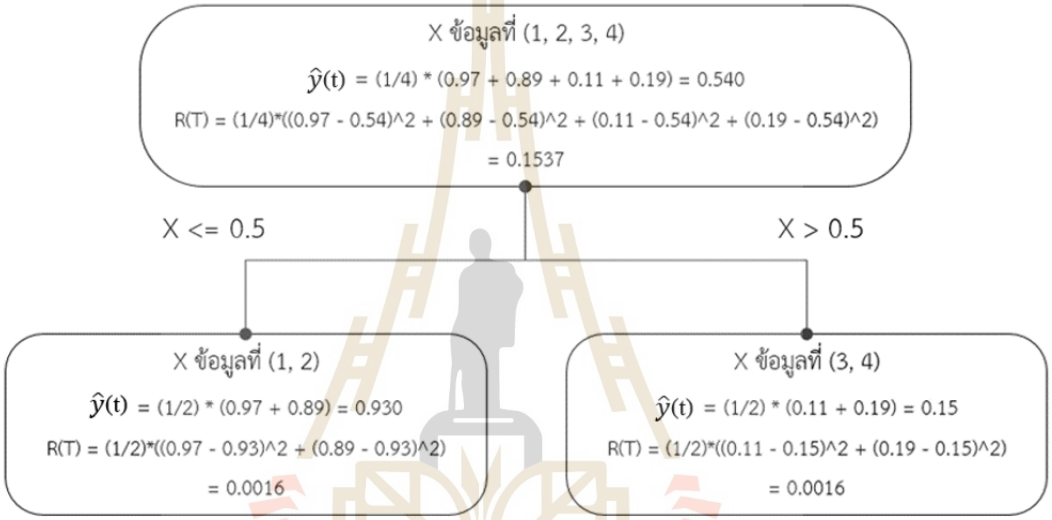
$R(t_R)$ คือ ค่า **SSE** ของโหนดลูกทางขวาของโหนด t

โดยจะแบ่งโหนดเมื่อ $R(t) \geq R(t_L) + R(t_R)$

ทำขั้นตอนที่ 1 และ 2 ซ้ำจนกว่าจำนวนของข้อมูลในโหนดนั้นมีค่าน้อยกว่าจำนวนข้อมูลที่ต้องการ $N(t) \leq N_{min}$

ตารางที่ 2.7 ข้อมูลที่ใช้ในการสร้างต้นไม้ตัดสินใจ CART สำหรับการคาดการณ์

ข้อมูลที่	X	Y
1	0.05	0.97
2	0.32	0.89
3	0.76	0.11
4	0.81	0.19

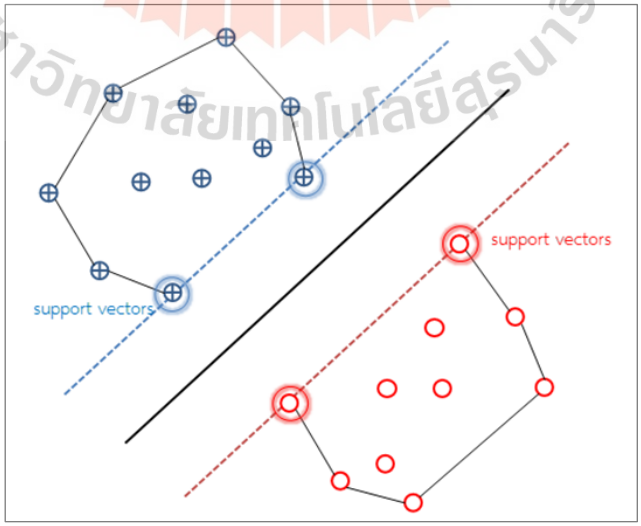


รูปที่ 2.11 ตัวอย่างการคำนวณค่าต่าง ๆ ในกระบวนการสร้างต้นไม้ตัดสินใจของ CART สำหรับการคาดการณ์ข้อมูลตัวเลข

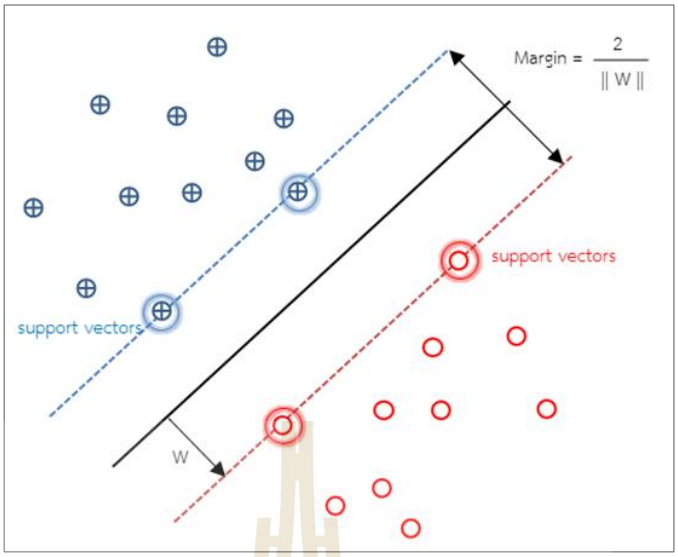
CART สำหรับการคาดการณ์ตัวเลขนั้นจะใช้ค่าเฉลี่ยของตัวแปรตามมาเป็นค่าของโหนดใบในต้นไม้ตัดสินใจ จากตารางที่ 2.7 และรูปที่ 2.11 เป็นการแสดงการคำนวณค่า R(t), $\hat{y}(t)$ และ $\Delta R(s,t)$ กรณีของการแบ่งค่าแอททริบิวต์ X ที่จุด 0.5 (โดยในที่นี้สมมติให้เป็นค่าเริ่มต้น) จากโครงสร้างต้นไม้ตัดสินใจนี้จะเห็นได้ว่าถ้า X น้อยกว่าหรือเท่ากับ 0.5 ตัวแบบจำลองคาดการณ์ค่า Y เป็น 0.93 และถ้า X มีค่ามากกว่า 0.5 ตัวแบบจำลองคาดการณ์ค่า Y เป็น 0.15 และค่า $\Delta R(s,t)$ จะเท่ากับ 0.1537 – 0.0016 - 0.0016 = 0.1505 ซึ่งอัลกอริทึม CART จะทำการปรับการแบ่งค่าแอททริบิวต์ X จนกว่าจะได้รับผลลัพธ์ตามต้องการ

2.8 Support Vector Machine (SVM)

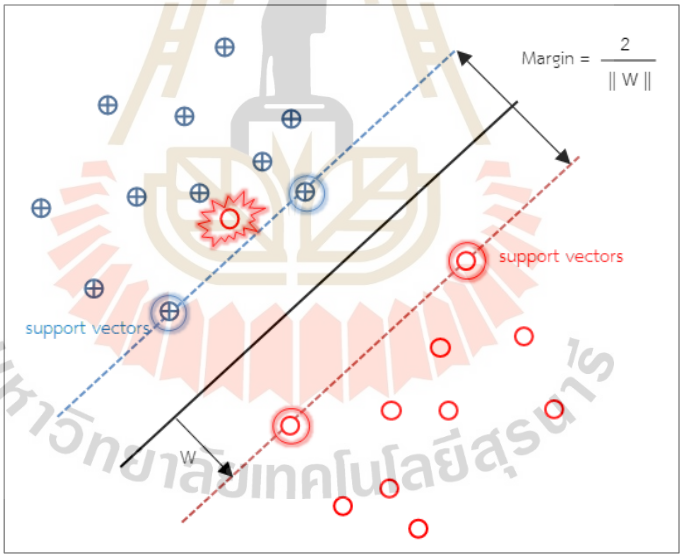
ซัพพอร์ตเวกเตอร์แมชชีน เป็นหนึ่งในอัลกอริทึมที่นิยมใช้ในการจำแนกคลาสของข้อมูล เนื่องจากมีความแม่นยำในการจำแนกสูง (Suykens and Vandewalle, 1999; Furey et al., 2000; Tong and Koller, 2001; Hsu et al., 2003; Meyer and Wien, 2015; Guenther and Schonlau, 2016) โดยมีพื้นฐานมาจากแบบจำลองเชิงเส้นและถูกพัฒนามาจากอัลกอริทึมเพอร์เซปตรอน ซึ่งได้มีการเพิ่มประสิทธิภาพในการจำแนกข้อมูลด้วยการพยายามปรับเส้นแบ่งให้เกิดระยะขอบ (Margin) ให้มากที่สุด หลักการที่สำคัญของซัพพอร์ตเวกเตอร์แมชชีน คือการนำค่าพารามิเตอร์ของชุดข้อมูลมาแปลงเป็นพิกัดเพื่อวางลงบนฟีเจอร์สเปซ (Feature Space) จากนั้นจึงทำการหาจุดขอบ (Convex Hull) ในแต่ละคลาสของข้อมูลมาลากเส้นขอบเชื่อมต่อกัน ดังรูปที่ 2.12 ซึ่งด้วยวิธีการนี้เองทำให้ซัพพอร์ตเวกเตอร์แมชชีนใช้ระยะเวลาในการสร้างแบบจำลองได้เร็วกว่าอัลกอริทึมเพอร์เซปตรอนที่นำข้อมูลทุกจุดมาใช้ในการคำนวณ โดยข้อมูลในแต่ละคลาสที่เป็นจุดของการสร้างเส้นขอบเพื่อแบ่งแยกคลาสจะเรียกว่าซัพพอร์ตเวกเตอร์ (Support Vectors) เมื่อแบบจำลองสร้างเส้นเชื่อมขอบได้เรียบร้อยแล้ว ก็จะทำการสร้างเส้นตรงที่อยู่ระหว่างจุดขอบทั้งสองกลุ่ม ซึ่งจะพยายามสร้างเส้นแบ่งให้มีระยะขอบที่มากที่สุด เพื่อแบ่งข้อมูลทั้งสองคลาสออกจากกัน ดังรูปที่ 2.13 ในบางกรณีเพื่อให้ได้ระยะขอบที่มากที่สุด จึงจำเป็นต้องใช้วิธีการยอมให้เกิดตัวแปรอนุโลม (Slack Variable) เข้ามาช่วย คือ การยอมให้เกิดการทำนายผิดได้บ้าง หรือยอมมองข้ามจุดขอบบางจุดไป เพื่อให้ได้มาซึ่งการจำแนกข้อมูลที่มีความแม่นยำมากที่สุด วิธีแบบนี้เป็นวิธีที่ถือได้ว่ามีความยืดหยุ่น ซึ่งจะพิจารณาได้ดังรูปที่ 2.14



รูปที่ 2.12 การลากเส้นเชื่อมจุดขอบของแต่ละกลุ่มตัวอย่าง



รูปที่ 2.13 การพยายามสร้างเส้นแบ่งระหว่างกลุ่มข้อมูล โดยให้มีระยะขอบที่มากที่สุด



รูปที่ 2.14 การยอมให้มีตัวแปรอนุโลมเพื่อให้ได้ระยะขอบที่มากที่สุด

การสร้างเส้นแบ่งของซัพพอร์ตเวกเตอร์แมชชีนถูกอธิบายด้วยการคำนวณดังสมการที่ 2.7

$$(x_i, y_i), \ldots, (x_n, y_n) \text{ เมื่อ } x \in R^m, y \in \{+1, -1\} \tag{2.7}$$

โดย $(x_i, y_i), \ldots, (x_n, y_n)$ คือ ข้อมูลตัวอย่างที่ใช้สำหรับการสอน

n คือ จำนวนข้อมูลตัวอย่าง

m คือ จำนวนมิติข้อมูลเข้า และ

y คือ ผลลัพธ์ที่แทนประเภทหรือกลุ่มข้อมูลมีค่า +1 หรือ -1

เมื่อ +1 แทน Positive Data และ –1 แทน Negative Data

สำหรับตัวอย่างปัญหาเชิงเส้น 2 มิติที่เป็นพื้นฐานของการจำแนกข้อมูล สามารถคำนวณได้ดังสมการที่ 2.8

$$(w^T * x) + b \tag{2.8}$$

โดย w คือ เวกเตอร์ถ่วงน้ำหนัก (Weight Vector) และ

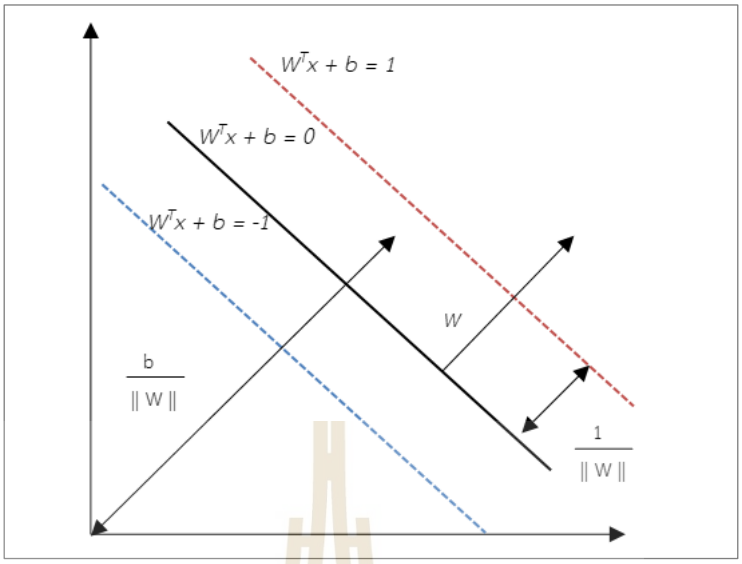
b คือ ค่าไบแอส (Bias)

สมการที่ใช้สำหรับจำแนกประเภทของข้อมูลด้วยอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนนั้นได้แสดงไว้ในสมการที่ 2.9 และ 2.10

$$(w^T * x) + b > 0, \ when\ y_i = +1 \tag{2.9}$$

$$(w^T * x) + b < 0, \ when\ y_i = -1 \tag{2.10}$$

ในส่วนของการหาค่าเวกเตอร์ถ่วงน้ำหนักนั้น สามารถหาได้จากความชันของเส้นแบ่ง กล่าวคือเวกเตอร์ถ่วงน้ำหนัก คือ เส้นที่ลากไปตั้งฉากกับเส้นแบ่ง แสดงดังรูปที่ 2.15



รูปที่ 2.15 เวกเตอร์ถ่วงน้ำหนัก และค่าไบแอส

นอกเหนือจากฟังก์ชันเชิงเส้นที่ใช้ในการแก้ปัญหาของข้อมูลใน 2 มิติแล้ว สำหรับข้อมูลที่มีมิติที่สูงขึ้น และสำหรับข้อมูลที่ไม่สามารถแบ่งแยกได้โดยง่าย ซัพพอร์ตเวกเตอร์แมชชีน ยังมีเคอร์เนลฟังก์ชัน (Kernel Function) ที่ถูกพัฒนาขึ้นมาให้ผู้ใช้สามารถประยุกต์ใช้ในการแก้ปัญหาได้หลายวิธี (Scholkopf et al., 2002; Hsu and Lin, 2002; Moghaddam and Hamidzadeh, 2016) เช่น ฟังก์ชันพหุนาม (Polynomial Function) ฟังก์ชันเรเดียลเบสิส (Radial Basis Function) และฟังก์ชันซิกมอยด์ (Sigmoid Function) ดังตารางที่ 2.8

ตารางที่ 2.8 เคอร์เนลฟังก์ชันสำหรับอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีน

Kernel	Inner Product Kernel
Linear	$x^T x_i$
Polynomial	$(x^T x_i + n)^d$
Radial-basis function	$\exp(-\gamma \|x - x_i\|)^2, \gamma > 0$
Sigmoidal	$\tanh(\gamma (x^T.x_i + n)), \gamma > 0$

อัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนนั้นจะมีพารามิเตอร์ต่าง ๆ เพื่อใช้ในการปรับปรุงประสิทธิภาพการจำแนกให้มากขึ้น โดยพารามิเตอร์ที่นิยมปรับใช้นั้น ได้แก่ พารามิเตอร์ C, พารามิเตอร์ Epsilon และพารามิเตอร์ Gamma (ในเคอร์เนลฟังก์ชันเรเดียลเบสิส)

พารามิเตอร์ C เป็นตัวควบคุมค่าใช้จ่าย (Cost) ที่ใช้ในการจำแนกข้อมูลที่ผิดคลาสในชุดข้อมูลฝึกสอน ดังที่ได้กล่าวมาตอนต้นของอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนนั้นมีโอกาสที่จะยอมให้เกิดการจำแนกข้อมูลที่ผิดพลาดบ้าง เพื่อแลกเปลี่ยนกับการได้ขอบที่มีระยะห่างมากขึ้น โดยค่าพารามิเตอร์ C ที่มีค่าน้อยหมายถึงการที่จะทำให้ความกว้างของขอบมีขนาดใหญ่ ส่งผลให้โมเดลมีความยืดหยุ่นต่อข้อมูลใหม่ที่ยังไม่ได้จำแนกสูง และหากค่าพารามิเตอร์ C มีค่ามากจะส่งผลให้ความกว้างของขอบมีขนาดแคบ อาจจะทำให้เกิดปัญหา Overfitting ได้

พารามิเตอร์ Epsilon เป็นตัวควบคุมการเพิ่มประสิทธิภาพในการจำแนกข้อมูลด้วยการช่วยในการหาค่าเส้นแบ่งหรือไฮเปอร์เพลนที่ดีที่สุดได้ดียิ่งขึ้น โดยไฮเปอร์เพลนจะมีการปรับปรุงตามข้อผิดพลาดที่เกิดขึ้นในรอบที่ผ่านมา จนกระทั่งได้ระยะห่างตามที่กำหนดไว้ หรือจนกว่าจะได้ค่าที่ดีที่สุด

พารามิเตอร์ Gamma เป็นพารามิเตอร์เฉพาะสำหรับเคอร์เนลฟังก์ชันเรเดียลเบสิส โดยส่วนมากจะใช้สำหรับการจำแนกในชุดข้อมูล 2 มิติ ที่ไม่สามารถแบ่งแยกข้อมูลออกจากกันได้โดยง่าย โดยจะใช้การแก้ปัญหาที่ไม่เป็นเชิงเส้น (Non-Linear) โดยพารามิเตอร์ Gamma นั้นจะทำหน้าที่ควบคุมความโค้งของไฮเปอร์เพลน ถ้าหากพารามิเตอร์ Gamma มีค่าน้อยจะทำให้ความกว้างของขอบมีขนาดกว้าง ซึ่งส่งผลให้การจำแนกข้อมูลที่เข้ามาใหม่นั้นมีความยืดหยุ่น แต่หากค่าพารามิเตอร์ Gamma มีค่ามากจะทำให้ความกว้างของขอบมีขนาดแคบ อาจจะทำให้เกิดปัญหา Overfitting ได้

2.9 การวิเคราะห์การถดถอยเชิงเส้น (Linear Regression Analysis: LR)

การวิเคราะห์การถดถอยเชิงเส้น (Montgomery et al., 2012; Seber and Lee, 2012; Kutner et al., 2005) เป็นวิธีการทางสถิติเพื่อใช้ในการศึกษาเกี่ยวกับการหาความสัมพันธ์ของข้อมูลเชิงปริมาณตั้งแต่สองตัวแปรขึ้นไป ที่มีระดับมาตรวัดของข้อมูลอยู่ในระดับมาตรวัดแบบช่วง (Interval Scale) หรือมาตรวัดแบบอัตราส่วน (Ratio Scale) ซึ่งตัวแปรทั้งสองจะต้องมีค่าความสหสัมพันธ์ (Correlation) อยู่ในระดับที่ยอมรับได้ จากนั้นจึงกำหนดว่าจะให้ตัวแปรใดเป็นตัวแปรต้น (X) ซึ่งจะนำมาใช้ทำนายตัวแปรตาม (Y) โดยการกำหนดตัวแปรต้นหรือตัวแปรตามนั้นต้องอาศัยความเข้าใจในข้อมูล การที่ให้ X เป็นตัวแปรต้นเพราะอยู่บนสมมุติฐานที่ว่า X คือ สาเหตุ และ Y คือ ผลกระทบและอยากทราบว่าตัวแปรต้นส่งผลอย่างไรกับตัวแปรตาม ซึ่งผลลัพธ์ที่ได้นั้นจะอยู่ในรูปแบบของสมการทางคณิตศาสตร์ ซึ่งผ่านกระบวนการคำนวณจากข้อมูลที่มีอยู่ เพื่อหาสมการที่ดีที่สุดที่สามารถทำนาย Y ด้วย X ผ่านรูปแบบสมการเส้นตรง ข้อดีอย่างหนึ่งของการวิเคราะห์การถดถอยเชิงเส้น คือ ตัวแปรต้นที่นำมาใช้นั้นไม่จำเป็นต้องอยู่ในรูปแบบที่มีการกระจายตัวแบบปกติก็ได้

เพราะถือว่าเป็นตัวแปรควบคุมในการทดลอง ในกรณีที่ใช้ตัวแปรต้นหนึ่งตัวเพื่อทำนายตัวแปรตามหนึ่งตัวผ่านรูปแบบสมการเส้นตรงจะเรียกวิธีการนี้ว่า การวิเคราะห์การถดถอยเชิงเส้นอย่างง่าย (Simple Linear Regression Analysis: SLR) และในกรณีที่ใช้ตัวแปรต้นหลายตัวเพื่อทำนายตัวแปรตามหนึ่งตัวผ่านรูปแบบสมการเส้นตรงจะเรียกว่า การวิเคราะห์การถดถอยเชิงเส้นพหุคูณ (Multiple Linear Regression Analysis: MLR) ซึ่งมีรายละเอียดดังต่อไปนี้

2.9.1 การวิเคราะห์การถดถอยเชิงเส้นอย่างง่าย

การวิเคราะห์การถดถอยเชิงเส้นอย่างง่าย (มนต์ชัย เทียนทอง, 2548; รติพร จันทร์กลั่น, 2560; Berger et al., 2018; Chatterjee and Hadi, 2015; Harrell, 2015) ประกอบไปด้วยตัวแปรที่ทราบค่า หรือตัวแปรต้น (X) และตัวแปรที่ไม่ทราบค่า หรือตัวแปรตาม (Y) อย่างละ 1 ตัวเท่านั้น และทั้งสองตัวแปรนี้ต้องมีความสัมพันธ์กันในลักษณะเส้นตรง โดยจุดประสงค์ของการวิเคราะห์การถดถอยเชิงเส้นอย่างง่ายเพื่อต้องการหาสมการที่จะนำไปสู่การคาดการณ์ หรือการประมาณค่าของตัวแปรที่ไม่ทราบค่าได้ โดยแสดงในสมการที่ 2.11

$$\hat{Y} = a + bX \tag{2.11}$$

เมื่อ $\hat{Y}$ คือ ตัวแปรตาม (ตัวแปรที่เราไม่ทราบค่าที่ต้องการจะพยากรณ์)

X คือ ตัวแปรต้น

a คือ ค่าคงที่ของสมการถดถอยหรือจุดตัดบนแกน Y

b คือ ความชันของเส้นถดถอย หรือสัมประสิทธิ์การถดถอยของตัวแปรต้น

สัมประสิทธิ์การถดถอย (Regression Coefficient) หรือค่าของ b ซึ่งเป็นความชันของกราฟเส้นตรงนั้นมีที่มาจากสมการเชิงเส้น ถ้าหากว่าทราบค่าของ b และค่าของ a จะทำให้สามารถพยากรณ์ค่าของตัวแปร Y ได้ ส่วน a เรียกว่า จุดตัดบนแกน y โดยจากที่กล่าวมาสามารถสรุปความสัมพันธ์ของ X และ Y จากค่าของ b ได้ดังนี้

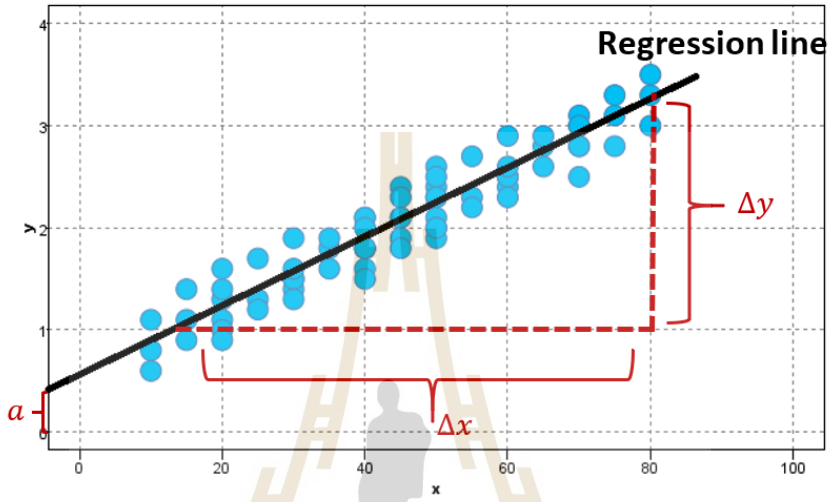
1. ในกรณีที่ b > 0 กล่าวได้ว่า X และ Y มีความสัมพันธ์กันไปในทิศทางเดียวกัน ถ้า X มีค่ามากขึ้น Y ก็มีค่ามากขึ้นตาม และถ้า X มีค่าน้อยลง Y ก็มีค่าน้อยลงตามด้วย

2. ในกรณีที่ b < 0 กล่าวได้ว่า X และ Y มีความสัมพันธ์กันในทิศทางที่ตรงกันข้าม ถ้า X มีค่ามากขึ้นค่าของ Y จะน้อยลง และถ้าหาก X มีค่าน้อยลงค่าของ Y จะมากขึ้น

3. ในกรณีที่ b มีค่าเข้าใกล้ 0 กล่าวได้ว่า X และ Y มีความสัมพันธ์กันน้อย

4. ในกรณีที่ b มีค่าเป็น 0 กล่าวได้ว่า X และ Y ไม่มีความสัมพันธ์กันเลย เส้นกราฟที่ได้จะเป็นเส้นตรงที่ค่าของ Y จะมีค่าเป็นค่าเดียว คือ ค่าคงที่ (a)

5. ในกรณีที่ b มีค่าเป็น 1 ความชันของเส้นกราฟทำมุม 45 องศา กล่าวได้ว่าค่า X และ Y จะมีค่าเท่ากัน คือ มีค่าคงที่ a เท่ากับศูนย์



รูปที่ 2.16 ตัวอย่างการ Plot กราฟของข้อมูล และเส้นกราฟถดถอย (Regression line)

ตารางที่ 2.9 ตัวอย่างข้อมูลแสดงการคำนวณการวิเคราะห์การถดถอยเชิงเส้นอย่างง่าย

ตัวที่	1	2	3	4	5	6	7	8	9	10	11	12	13
X	40	50	55	60	40	45	50	55	60	65	70	75	80
Y	42	46	47	48	40	41	43	43	45	48	60	61	63

จากตารางที่ 2.9 แสดงตัวอย่างข้อมูลเพื่อใช้ในการคำนวณการวิเคราะห์การถดถอยเชิงเส้นอย่างง่าย โดยกำหนดให้ตัวแปรต้น คือ X แทนอุณหภูมิภายนอกของฮาร์ดดิสก์ไดรฟ์ขณะทำงานในหน่วยของศาเซลเซียส และ Y เป็นตัวแปรที่ต้องการพยากรณ์โดยในที่นี้คือ อุณหภูมิภายในของฮาร์ดดิสก์ไดรฟ์ขณะทำงานในหน่วยของศาเซลเซียส ข้อมูลดังกล่าวถูกจัดเตรียมให้อยู่ในตารางเพื่อให้ง่ายในการคำนวณเส้นถดถอย ดังตารางที่ 2.10 โดยตารางนี้จะทำการหาผลรวม ค่าเฉลี่ย ผลคูณ ผลรวมของผลคูณของตัวแปรต้นกับตัวแปรตาม และค่ายกกำลังสองของตัวแปรต้น เพื่อหาสมการถดถอยที่สามารถพยากรณ์ได้ว่าที่อุณหภูมิแตกต่างกันจะทำให้อุณหภูมิภายในของฮาร์ดดิสก์ไดรฟ์เป็นเท่าใด

ตารางที่ 2.10 แสดงผลรวม ค่าเฉลี่ย ผลคูณ ผลรวมของผลคูณของตัวแปรต้นกับตัวแปรตาม และค่ายกกำลังสองของตัวแปรต้น

ตัวที่	X	Y	XY	X^2
1	40	42	1680	1600
2	50	46	2300	2500
3	55	47	2585	3025
4	60	48	2880	3600
5	40	40	1600	1600
6	45	41	1845	2025
7	50	43	2150	2500
8	55	43	2365	3025
9	60	45	2700	3600
10	65	48	3120	4225
11	70	60	4200	4900
12	75	61	4575	5625
13	80	63	5040	6400
ผลรวม	**745.0**	**627.00**	**37040.0**	**44625.0**
ค่าเฉลี่ย	**57.3**	**48.23**	**2849.2**	**3432.7**

คำนวณหาสัมประสิทธิ์ของตัวแปรต้นและค่าคงที่ โดยนำค่าที่คำนวณได้จากตารางที่ 2.10 มาคำนวณหาดังนี้

$$b = \frac{n\sum XY-(\sum X)(\sum Y)}{n\sum X^2-(\sum X)^2}$$

$$= \frac{13(37040)-(745)(627)}{13(44625)-(745)^2}$$

$$= \frac{481520-467115}{580125-555025}$$

$$= \frac{14405}{25100}$$

$$= 0.5739$$

$$a = \overline{Y} - b\overline{X}$$

$$= 48.23 - (0.5739)57.31$$

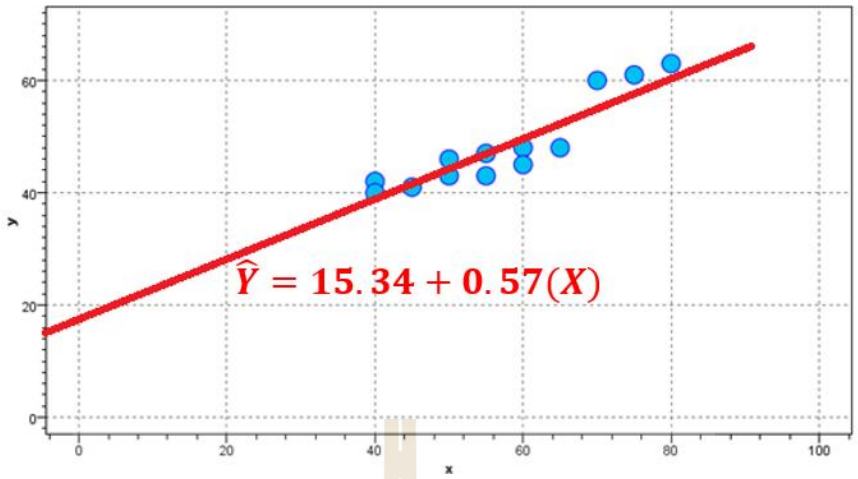
$$= 48.23 - 32.889$$

$$= 15.3416$$

สมการถดถอยอย่างง่ายจากข้อมูลข้างต้น สามารถเขียนได้ดังสมการต่อไปนี้

$$\hat{Y} = a + bX$$

$$\hat{Y} = 15.34 + 0.57(X)$$



รูปที่ 2.17 การกระจายตัวของข้อมูลอุณหภูมิภายนอก อุณหภูมิภายในฮาร์ดดิสก์ไดรฟ์ขณะฮาร์ดดิสก์ไดรฟ์กำลังทำงาน และเส้นกราฟถดถอย

จากรูปที่ 2.17 แสดงการกระจายตัวของข้อมูลอุณหภูมิภายนอกฮาร์ดดิสก์ไดรฟ์ (แกน X) อุณหภูมิภายในฮาร์ดดิสก์ไดรฟ์ (แกน Y) และเส้นกราฟถดถอย เมื่อได้เส้นกราฟถดถอยที่ดีที่สุดมาแล้ว จะสามารถพยากรณ์สิ่งที่จะเกิดในอนาคตได้ ตัวอย่างเช่น อยากทราบว่าที่อุณหภูมิภายนอก 90 องศาเซลเซียส ส่งผลให้อุณหภูมิภายในฮาร์ดดิสก์ไดรฟ์จะเป็นเท่าใด วิธีการ คือ สามารถแทนค่า X ลงไปในสมการถดถอยที่ได้มา ก็จะทำให้ทราบค่า Y หรือในที่นี้ คือ อุณหภูมิภายในฮาร์ดดิสก์ไดรฟ์ขณะทำงานนั่นเอง และคำตอบที่ได้ ณ อุณหภูมิภายนอก 90 องศาเซลเซียส มีอุณหภูมิภายในฮาร์ดดิสก์ไดรฟ์อยู่ที่ 66.64

2.9.2 การวิเคราะห์การถดถอยเชิงเส้นพหุคูณ

การวิเคราะห์การถดถอยเชิงเส้นพหุคูณ หรือ Multiple Linear Regression (Fox, 2015; Aiken et al., 2012) เป็นการศึกษาความสัมพันธ์ระหว่างตัวแปรตาม (Y) หนึ่งตัวและตัวแปรต้นหลายตัว ($X_1, X_2, X_3, \ldots X_n$) โดยที่ตัวแปรต้นทั้งหลายเหล่านี้ต้องมีความสัมพันธ์กับตัวแปรตามในลักษณะเส้นตรง และตัวแปรต้นทุกตัวต้องไม่มีความสัมพันธ์กันเอง (Collinearity) การวิเคราะห์การถดถอยเชิงเส้นพหุคูณ สามารถตอบโจทย์ปัญหาที่เกิดขึ้นในชีวิตจริงซึ่งมีหลายปัจจัยและความซับซ้อนมากกว่า ทำให้ไม่สามารถใช้การวิเคราะห์การถดถอยเชิงเส้นอย่างง่ายที่มีตัวแปรต้นเพียงตัวเดียวมาอธิบายความสัมพันธ์ได้อย่างแม่นยำ ดังนั้น จึงจำเป็นที่จะต้องใช้การวิเคราะห์การถดถอยเชิงเส้นพหุคูณซึ่งแสดงในสมการที่ 2.12

$$\hat{Y} = b_0 + b_1X_1 + b_2X_2 + \cdots + b_kX_k \qquad (2.12)$$

เมื่อ $\hat{Y}$ คือ ตัวแปรตาม (ตัวแปรที่เราไม่ทราบค่าและต้องการจะพยากรณ์)

$X_1, X_1, X_1, \ldots, X_k$ คือ ตัวแปรต้นตัวที่ 1, 2, 3 ไปจนถึง ตัวที่ k

b_0 คือ ค่าคงที่ของสมการถดถอยหรือจุดตัดบนแกน Y

$b_1,\ b_2,\ b_3,\ \ldots, b_k$ คือ ความชันของเส้นถดถอย หรือสัมประสิทธิ์การถดถอยของตัวแปรต้นตัวที่ 1, 2, 3 ไปจนถึง ตัวที่ k

ใน Multiple Linear Regression นั้นจะมีวิธีการคัดเลือกตัวแปรต้นหรือตัวแปรทำนายเพื่อให้สมการสามารถทำนายได้มีประสิทธิภาพที่ดีสุด โดยมีวิธีการคัดเลือกตัวแปรดังนี้

1. วิธีการคัดเลือกตัวแปรแบบ Forward Selection เป็นการเลือกตัวแปรทำนายที่มีสหสัมพันธ์กับตัวแปรตามสูงที่สุดเข้าสมการก่อน ส่วนตัวแปรที่เหลือจะมีการคำนวณค่าสัมประสิทธิ์การถดถอยในรูปคะแนนมาตรฐาน (t-test) ทดสอบนัยสำคัญของค่าสัมประสิทธิ์การถดถอยในรูปคะแนนมาตรฐาน และค่าสหสัมพันธ์แบบแยกส่วน (Partial Correlation) โดยเป็นความสัมพันธ์เฉพาะตัวแปรที่เหลือตัวนั้นกับตัวแปรตามซึ่งนั่นตีความได้ว่าตรงจุดนี้ได้กำจัดอิทธิพลของตัวแปรอื่น ๆ ออกไปแล้ว โดยถ้าตัวแปรใดมีค่าสัมประสิทธิ์การถดถอยมีนัยสำคัญทางสถิติก็จะนำเข้ากระบวนการนี้และจะกระทำการวนซ้ำจนกระทั่ง ตัวแปรที่เหลืออยู่ไม่มีนัยสำคัญทางสถิติ จึงจะทำการหยุดและได้สมการที่มีสัมประสิทธิ์การทำนายสูงสุด

2. วิธีการคัดเลือกตัวแปรแบบ Backward Selection เป็นการนำตัวแปรทำนายทั้งหมดเข้าสมการ จากนั้นจึงทำการตัดตัวแปรทำนายออกทีละตัว โดยพิจารณาจากค่าสัมประสิทธิ์การถดถอยของตัวแปรทำนายที่อยู่ในสมการ หากพบว่าไม่มีนัยสำคัญทางสถิติก็จะทำการตัดตัวแปรนั้นออกจากสมการ จนกระทั่งตัวแปรทำนายที่เหลืออยู่ในสมการมีนัยสำคัญทางสถิติทั้งหมดจึงจะหยุดการคัดเลือก และได้สมการการทดสอบที่มีสัมประสิทธิ์การทำนายสูงสุด

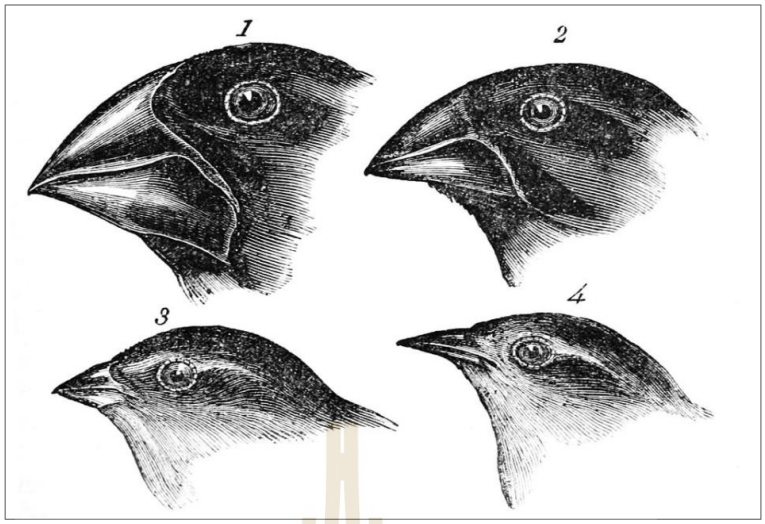
3. การคัดเลือกตัวแปรแบบ Stepwise Selection การคัดเลือกชนิดนี้เป็นการผสมผสานระหว่างวิธีการคัดเลือกตัวแปรทั้งสองวิธีที่ได้กล่าวมาในข้างต้น ในขั้นแรกจะเลือกตัวแปรที่มีสหสัมพันธ์กับตัวแปรตามสูงที่สุดเข้าสมการก่อน จากนั้นจึงทดสอบตัวแปรที่ไม่ได้อยู่ในสมการว่าจะมีตัวแปรใดบ้างมีสิทธิ์เข้ามาอยู่ในสมการด้วยวิธีการคัดเลือกแบบ Forward Selection และขณะเดียวกันก็จะทดสอบตัวแปรที่อยู่ในสมการด้วยว่าตัวแปรใดมีโอกาสที่จะถูกตัดออกจากสมการด้วยวิธีการคัดเลือกแบบ Backward Selection โดยจะกระทำการคัดเลือกด้วยสองวิธีนี้ในทุกครั้งจนกระทั่งไม่มีตัวแปรใดที่ถูกคัดออกและไม่มีตัวแปรใดที่จะถูกนำเข้าสมการอีก จากนั้นกระบวนการจึงยุติการทำงานและได้สมการถดถอยที่มีสัมประสิทธิ์การทำนายสูงสุด

โดยสรุปแล้ว Linear Regression นั้นนับได้ว่าเป็นอัลกอริทึมสารพัดประโยชน์ ซึ่งงานวิจัยชิ้นนี้ได้นำอัลกอริทึม MLR มาใช้ในการสร้างโมเดลการเรียนรู้ในการคาดการณ์ผลผลิต นอกจากนั้นแล้วยังนำ Stepwise Regression มาประยุกต์ใช้ในขั้นตอนประบวนการคัดเลือกคุณลักษณะอีกด้วย

2.10 ขั้นตอนวิธีเชิงพันธุกรรม (Genetic Algorithm)

Genetic Algorithm หรือนิยมเรียกชื่อย่อว่า GA เป็นวิธีการทางปัญญาประดิษฐ์อย่างหนึ่งที่ใช้ในการค้นหาคำตอบ รวมถึงสามารถนำมาใช้เพื่อเพิ่มประสิทธิภาพการเรียนรู้ของเครื่อง GA ได้รับพื้นฐานแนวคิดมาจากทฤษฎีวิวัฒนาการทางธรรมชาติของ ชาลส์ โรเบิร์ต ดาวินส์ (Charles Robert Darwin) ซึ่งถูกตีพิมพ์ในหนังสือ “The Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life” เมื่อปี ค.ศ. 1859 (Darwin and Bynum, 2009) ทฤษฎีนี้กล่าวถึงความหลากหลายตามธรรมชาติและความแตกต่างของแต่ละสายพันธุ์ของสิ่งมีชีวิตนั้นมีต้นกำเนิดร่วมกัน แต่ปัจจัยของสภาพแวดล้อมที่อาศัยอยู่และการกลายพันธุ์จากรุ่นสู่รุ่นอย่างต่อเนื่อง มีอิทธิพลทำให้เกิดความแตกต่างของสายพันธุ์ขึ้นดังที่เป็นอยู่ในปัจจุบันนี้ โดยสายพันธุ์ที่มีความเหมาะสมกับสภาพแวดล้อมมากกว่า จะมีโอกาสอยู่รอดและถ่ายทอดลักษณะทางพันธุกรรมไปสู่รุ่นต่อไปได้มากกว่า ในขณะที่สายพันธุ์ที่ไม่เหมาะสมกับสภาพแวดล้อมจะมีโอกาสอยู่รอดได้น้อยและอาจจะสูญพันธุ์ไปในที่สุด โดยกระบวนการนี้เป็นที่รู้จักดีภายใต้ชื่อว่ากระบวนการคัดเลือกทางธรรมชาติ (Natural Selection)

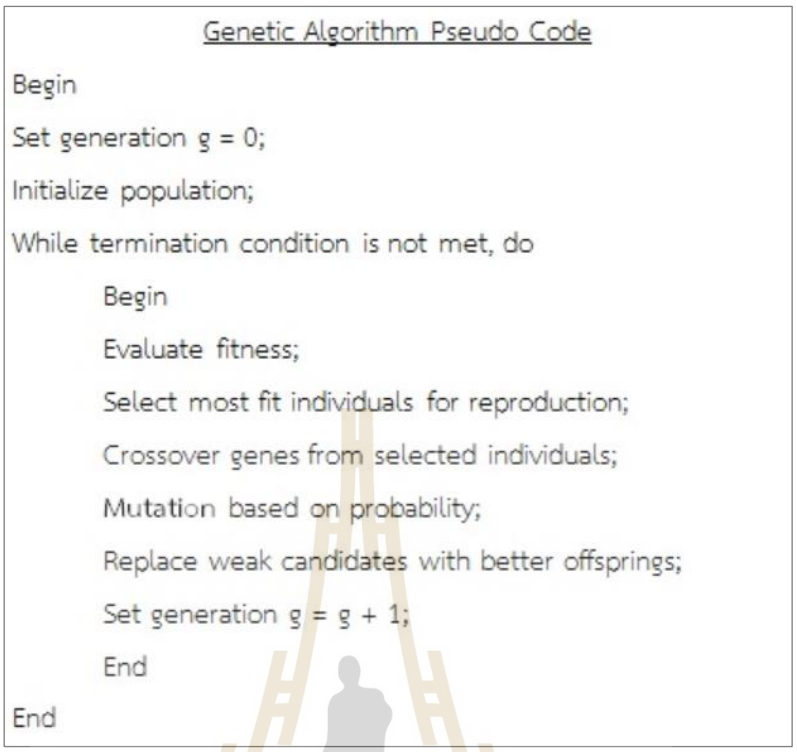
รูปที่ 2.18 เป็นภาพที่แสดงถึงนกกระจอก (Finches) ที่มีลักษณะและขนาดของจะงอยปากแตกต่างกันตามสภาพแวดล้อมและแหล่งอาหารตามสภาพแวดล้อมเหล่านั้น (Lamichhaney et al., 2015; Marsh, 2015; Soons et al., 2010; Podos and Nowicki, 2004) ยกตัวอย่างเช่นนกกระจอก (1) มีขนาดปากที่หนาและใหญ่ซึ่งคาดว่าเป็นสายพันธุ์ดั้งเดิมที่หาอาหารจากเมล็ดพืช ซึ่งส่วนใหญ่มักจะอยู่ตามพื้นดิน นกกระจอก (2) เป็นสายพันธุ์ที่ปรับตัวเข้ากับสภาพแวดล้อมที่มีอาหารชนิดอื่นนอกจากเมล็ดพืช ได้แก่ หน่วยอ่อนของดอกไม้ และผลไม้ เป็นต้น ซึ่งจะต้องมีขนาดจะงอยปากที่เล็กลงเพื่อให้เหมาะสมกับการบินขึ้นไปเพื่อหาอาหารเหล่านั้น นกกระจอก (3) มีขนาดปากที่เล็กลงไปอีก ทั้งยังมีลักษณะของหัวที่เรียวและกระทัดรัดมากขึ้นจึงเหมาะสมกับการหาอาหารจำพวกใบไม้ใบหญ้า ในส่วนของนกกระจอก (4) มีจะงอยปากที่ยาวและเรียวแหลมเหมาะกับสภาพแวดล้อมที่มีอาหารจำพวกแมลง ซึ่งการคัดเลือกทางธรรมชาตินี้เองทำให้นกกระจอกสายพันธุ์นี้สามารถจับแมลงได้คล่องแคล่วและแม่นยำมากยิ่งขึ้น



รูปที่ 2.18 นกกระจอกที่มีจะงอยปากและขนาดที่แตกต่างกันตามสภาพแวดล้อมที่อยู่อาศัย
(Reference: https://en.wikipedia.org/wiki/Darwin%27s_finches)

ขั้นตอนวิธีเชิงพันธุกรรมได้ถูกนำเสนอในงานวิจัยของ John Holland (Holland, 1975) โดยเป็นการประยุกต์ใช้แนวคิดจากกระบวนการวิวัฒนาการทางธรรมชาติในการค้นหาคำตอบของปัญหา ซึ่งจุดเด่นในด้านความทนทานต่อความผิดพลาดเพื่อการค้นหาคำตอบจากแหล่งข้อมูลที่มีความซับซ้อนและมีความแตกต่างกันนั้น (Russell and Norvig, 2002) ทำให้อัลกอริทึมนี้เริ่มได้รับความนิยมเป็นที่แพร่หลายในการนำไปประยุกต์ใช้ในการแก้ปัญหาต่าง ๆ มากขึ้น การทำงานของขั้นตอนวิธีเชิงพันธุกรรมนั้นสามารถสรุปออกมาได้เป็นหัวข้อของขั้นตอนการทำงานดังต่อไปนี้

1. การเริ่มต้นสร้างประชากร (Initialize Population)
2. การประเมินค่าด้วยฟังก์ชันความเหมาะสม (Evaluate by Fitness Function)
3. การดำเนินการทางขั้นตอนวิธีเชิงพันธุกรรม (Genetic Operations)
4. การแทนที่ด้วยประชากรรุ่นลูกที่ดีกว่า (Replacement by Better Offspring)
5. การตรวจสอบเงื่อนไขในการจบการทำงาน (Termination Condition)



Genetic Algorithm Pseudo Code

```
Begin
Set generation g = 0;
Initialize population;
While termination condition is not met, do
        Begin
        Evaluate fitness;
        Select most fit individuals for reproduction;
        Crossover genes from selected individuals;
        Mutation based on probability;
        Replace weak candidates with better offsprings;
        Set generation g = g + 1;
        End
End
```

รูปที่ 2.19 Pseudo Code สำหรับขั้นตอนวิธีเชิงพันธุกรรม

(Reference: K.Leelawong, ICCS 451 Lecture Note, March 2009)

จากรูปที่ 2.19 เป็นแผนภาพที่แสดงให้เห็นถึงรหัสเทียม (Pseudo Code) ของขั้นตอนการทำงานหลักของของ Genetic Algorithm เริ่มตั้งแต่ขั้นตอน Initialize population ไปจนจบสิ้นกระบวนการที่ Terminate Condition

การเริ่มต้นสร้างประชากร (Initialize population) นั้นจะเริ่มด้วยการสุ่มจากประชากรที่มีอยู่ทั้งหมดเพื่อนำเข้าสู่กระบวนการของ Genetic Algorithm ตามจำนวนของประชากรที่กำหนดไว้ หลังจากที่ได้ประชากรเริ่มต้นแล้วระบบจะทำการประเมินค่าความเหมาะสม (Evaluate by Fitness Function) ของประชากรเหล่านั้นและคัดเลือกประชากรที่มีความเหมาะสมตามที่กำหนด (Most fit individuals for reproduction) เพื่อนำโครโมโซมของประชากรเหล่านั้นไปใช้ในการเริ่มต้นสืบทอดพันธุกรรมในรุ่นถัดไป และเมื่อได้โครโมโซมมาแล้วจะมีการดำเนินการทางพันธุกรรม (Genetic Operations) ได้แก่ การคัดเลือกเพื่อการสืบพันธุ์ (Selection for reproduction) การแลกเปลี่ยนพันธุกรรม (Crossover) และการกลายพันธุ์ (Mutation) เพื่อให้เกิดความหลากหลายทางพันธุกรรม ซึ่งประชากรใหม่นั้นจะได้รับข้อมูลโครโมโซมจากประชากรเริ่มต้นโดยผ่านทั้งการสลับสายพันธุ์และการกลายพันธุ์ จึงทำให้ประชากรใหม่นั้นจะมีความแตกต่างไปจากประชากรดั้งเดิมเล็กน้อย

หลังจากที่ได้ประชากรใหม่แล้วระบบจะทำการแทนที่ประชากรรุ่นก่อนหน้าด้วยประชากรใหม่ที่มีความเหมาะสมมากกว่า จนกระทั่งได้ประชากรรุ่นสุดท้ายที่บรรลุตามเงื่อนไขที่กำหนดไว้

การกำหนดและการเข้ารหัสโครโมโซม (Chromosome Encoding) เป็นการเตรียมพร้อมข้อมูลก่อนนำชุดข้อมูลเข้าสู่ขั้นตอนแรกของกระบวนการ มีหน้าที่ในการออกแบบให้โครโมโซมเป็นตัวแทนของคำตอบของปัญหา โดยทั่วไปแล้วมักจะปรากฏในรูปแบบของชุดสายข้อมูลของเลขฐานสอง (Bit string) หรือ ชุดสายข้อมูลของอักขระ (Alphabet String) วิธีการเข้ารหัสโครโมโซมที่เป็นที่นิยมในปัจจุบันนั้นมี 3 วิธี ได้แก่

1. การเข้ารหัสแบบเลขฐานสอง (Binary Encoding) เป็นรูปแบบการเข้ารหัสแบบพื้นฐานที่เป็นที่นิยมและแพร่หลายมากที่สุด วิธีการเข้ารหัสแบบนี้เป็นการแปลงค่าให้อยู่ในรูปแบบเลขฐานสอง โดยจะมีรูปแบบของโครโมโซมอยู่ในรูป Bit String (แสดงค่าด้วยตัวเลข 0 และ 1 เท่านั้น) ดังที่ได้แสดงตัวอย่างไว้ในรูปที่ 2.20

Chromosome #1 :	0	1	1	0	1
Chromosome #2 :	1	0	0	1	1

รูปที่ 2.20 ตัวอย่างโครโมโซมจากการเข้ารหัสแบบเลขฐานสอง

2. การเข้ารหัสแบบค่าจริง (Value Encoding) ถูกนำเสนอโดย Alden H. Wright (Wright, 1991) เป็นรูปแบบการเข้ารหัสโดยใช้ค่าของข้อมูลที่เชื่อมโยงกับคำตอบของปัญหาโดยตรง เช่น “% ของผลิตภัณฑ์ที่ผ่านการทดสอบ” สภาพอากาศของแต่ละวัน ดังที่ได้แสดงตัวอย่างไว้ในรูปที่ 2.21

Chromosome #4 :	95.0	95.5	99.0	100.0	80.0
Chromosome #5 :	94.5	88.8	97.5	99.0	88.0
Chromosome #6 :	99.1	93.2	90.2	98.5	85.0

Chromosome #7 :	Rain	Rain	Cloud	Sun	Sun
Chromosome #8 :	Snow	Cloud	Cloud	Cloud	Rain
Chromosome #9 :	Cloud	Cloud	Snow	Rain	Sun

รูปที่ 2.21 ตัวอย่างโครโมโซมจากการเข้ารหัสแบบค่าจริง

3. การเข้ารหัสแบบเพอร์มิวเตชัน (Permutation Encoding) ถูกนำเสนอโดย Malhotra และคณะ (Malhotra et al., 2011) เป็นการเข้ารหัสโดยใช้การแทนค่าลำดับลงไปที่ทุกตำแหน่งของยีนในโครโมโซม ดังที่ได้แสดงไว้ในรูปที่ 2.22 ซึ่งหมายความว่าค่าในแต่ละตำแหน่งในโครโมโซมเดียวกันจะไม่ซ้ำกันเลย

Chromosome #10 :	1	2	3	4	5
Chromosome #11 :	4	5	2	3	1
Chromosome #12 :	3	1	2	5	4

รูปที่ 2.22 ตัวอย่างโครโมโซมจากการเข้ารหัสแบบเพอร์มิวเตชัน

2.10.1 การเริ่มต้นสร้างประชากร (Initialize Population)

ขั้นตอนที่ 1 การเริ่มต้นสร้างประชากรจะเป็นการสร้างประชากรชุดแรกโดยจะเริ่มจากการสุ่มประชากรทั้งหมด เพื่อให้ได้จำนวนประชากรตามที่กำหนดไว้สำหรับค้นหาคำตอบของปัญหา ซึ่งในขั้นตอนการเริ่มต้นสร้างประชากรนี้จะไม่นำการประเมินค่าความเหมาะสมเข้ามาพิจารณาด้วย โดยรูปที่ 2.23 และ 2.24 แสดงถึงตัวอย่างการสุ่มเลือกประชากรในขั้นตอนการเริ่มต้นสร้างประชากรชุดแรก ในตัวอย่างนี้จะมีประชากรทั้งหมด 12 แถวข้อมูล และกำหนดให้จำนวนประชากรที่ต้องการสำหรับค้นหาคำตอบได้แก่ 6 ข้อมูล ตัวอย่างที่แสดงในรูปที่ 2.24 นั้นได้สุ่มเลือกมาจากประชากรตัวที่ 1, 6, 3, 2, 11 และ 9 ตามลำดับ

	Gene#1	Gene#2	Gene#3	Gene#4	Gene#5
Chromosome #1 :	0	1	1	0	1
Chromosome #2 :	1	0	0	1	1
Chromosome #3 :	0	1	0	0	0
Chromosome #4 :	0	0	1	0	0
Chromosome #5 :	0	1	0	0	1
Chromosome #6 :	0	0	0	0	1
Chromosome #7 :	0	0	0	1	1
Chromosome #8 :	1	1	1	1	1
Chromosome #9 :	0	0	1	0	0
Chromosome #10 :	1	0	0	0	1
Chromosome #11 :	1	1	0	1	1
Chromosome #12 :	1	1	1	1	1

รูปที่ 2.23 ตัวอย่างประชากรทั้งหมด

	Gene#1	Gene#2	Gene#3	Gene#4	Gene#5
Chromosome #1 :	0	1	1	0	1
Chromosome #6 :	0	0	0	0	1
Chromosome #3 :	0	1	0	0	0
Chromosome #2 :	1	0	0	1	1
Chromosome #11 :	1	1	0	1	1
Chromosome #9 :	0	0	1	0	0

รูปที่ 2.24 ตัวอย่างการสุ่มประชากรเพื่อนำมาใช้เป็นประชากรเริ่มต้น

2.10.2 การประเมินค่าด้วยฟังก์ชันความเหมาะสม (Evaluation by Fitness Function)

ขั้นตอนนี้จะเริ่มจากการกำหนดฟังก์ชันค่าความเหมาะสม (Fitness Function) ขึ้นมาก่อนเป็นอันดับแรก แล้วจึงนำฟังก์ชันค่าความเหมาะสมไปประเมินความเหมาะสมของแต่ละโครโมโซมเพื่อใช้เป็นหลักเกณฑ์ในการพิจารณาว่าโครโมโซมเหล่านั้นสมควรได้รับการสืบทอดพันธุกรรมไปยังรุ่นต่อไปหรือไม่ โดยวิธีการคำนวณค่าความเหมาะสมนั้นจะแตกต่างกันออกไปเพื่อให้สอดคล้องกับแต่ละวิธีการแก้ไขปัญหา

ตัวอย่าง กำหนดให้ Fitness Function สำหรับการประเมินค่าความเหมาะสม คือ “การแปลงเลขฐานสองจำนวน 5 บิตให้อยู่ในรูปของเลขฐานสิบและนำค่าที่ได้ไปยกกำลังสอง โดยค่าที่น้อยที่สุดจะเป็นค่าที่เหมาะสมในการสืบทอดพันธุกรรม” จากตัวอย่างที่กำหนดมาให้นี้แสดงเป็นขั้นตอนย่อยในการดำเนินงาน ได้แก่

1. การจัดเรียงแต่ละโครโมโซมให้อยู่ในรูปแบบเลขฐานสอง เช่น Chromosome#1 = 01101, Chromosome#6 = 00001 และ Chromosome#3 = 01000 เป็นต้น

2. จากนั้นจึงนำเลขฐานสองที่ได้แปลงเป็นเลขฐานสิบได้ดังนี้ Chromosome#1 = 13, Chromosome#6 = 1, Chromosome#3 = 8, Chromosome#2 = 19, Chromosome#11 = 27 และ Chromosome#9 = 4

3. หลังจากที่ได้เลขฐานสิบมาเรียบร้อยแล้ว จึงนำมาหาค่ายกกำลังสองดังสมการที่ได้แสดงในสมการที่ 2.13 ซึ่งผลลัพธ์ของการคำนวณค่าความเหมาะสมที่ได้จะถูกแสดงไว้ในรูปที่ 2.25

$$F(x) = x^2 \tag{2.13}$$

	Gene#1	Gene#2	Gene#3	Gene#4	Gene#5	Fitness value
Chromosome #1 :	0	1	1	0	1	169
Chromosome #6 :	0	0	0	0	1	1
Chromosome #3 :	0	1	0	0	0	64
Chromosome #2 :	1	0	0	1	1	361
Chromosome #11 :	1	1	0	1	1	729
Chromosome #9 :	0	0	1	0	0	16

รูปที่ 2.25 การคำนวณค่าความเหมาะสมตาม Fitness Function ที่ได้กำหนดไว้

2.10.3 การดำเนินการทางขั้นตอนวิธีเชิงพันธุกรรม (Genetic Operations)

ตัวดำเนินการ (Operator) ที่ใช้ในขั้นตอนวิธีเชิงพันธุกรรมนั้น จะประกอบด้วยตัวดำเนินการ 3 ชนิด ได้แก่ การสืบพันธุ์ (Inheritance หรือ Reproduction) การแลกเปลี่ยนพันธุกรรม (Crossover หรือ Recombination) และการกลายพันธุ์ (Mutation) โดยมีลำดับการนำไปใช้ ตามรหัสเทียมที่ได้แสดงไว้ข้างต้นในรูปที่ 2.19

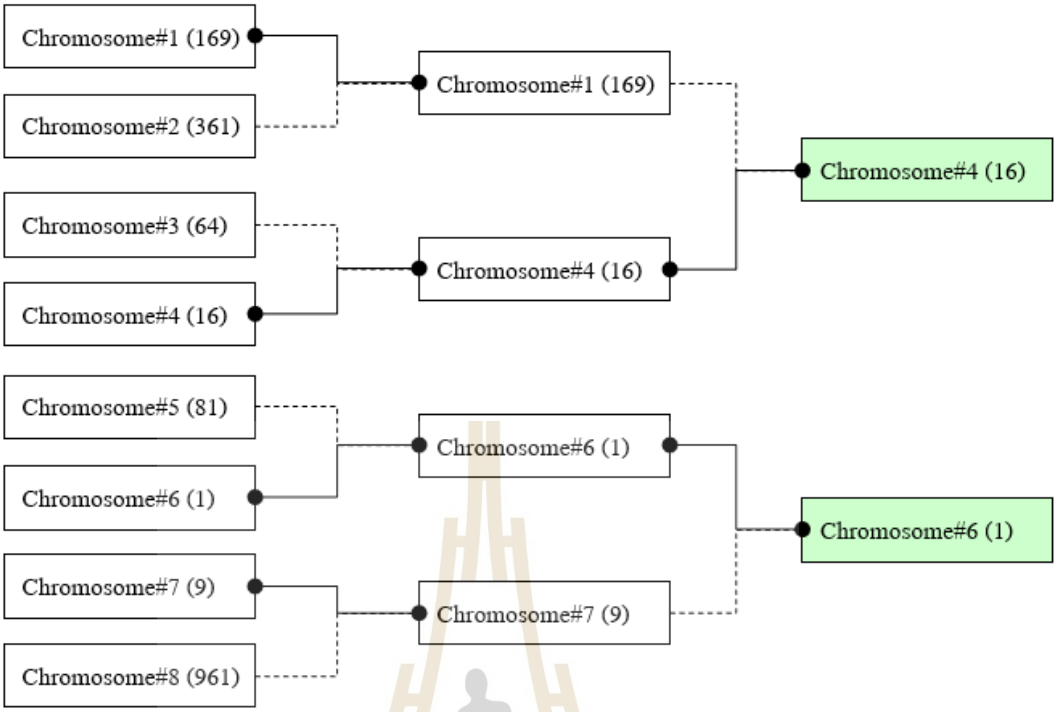
1. การสืบพันธุ์ เป็นตัวดำเนินการทางขั้นตอนวิธีเชิงพันธุกรรมในลำดับแรก โดยเป็นการสร้างประชากรรุ่นใหม่จากการคัดเลือกประชากรชุดเดิมในรุ่นก่อนหน้า ซึ่งการคัดเลือกจะเป็นตามคะแนนความเหมาะสมที่ได้รับจากการประเมินค่าฟังก์ชันความเหมาะสม (Fitness Function) ซึ่งเป็นการเลียนแบบกระบวนการคัดเลือกตามธรรมชาติ โดยสายพันธุ์ที่มีความเหมาะสมกับสภาพแวดล้อมมากกว่าจะมีโอกาสในการอยู่รอดและสืบทอดสายพันธุ์ได้มากกว่า วิธีการที่ใช้สำหรับการคัดเลือกประชากรในกระบวนการสืบพันธุ์ (กีระชาติ สุขสุทธิ์, 2559; Myreaders, 2010; Yaeger, 2008; Bies et al., 2006; Banzhaf et al., 1998) ได้แก่

การคัดเลือกแบบการสุ่มเลือก (Roulette Wheel) โดยการสุ่มนี้จะอ้างอิงหลักการของความน่าจะเป็นในการถูกคัดเลือก โดยจะเป็นไปตามอัตราส่วนของคะแนนความเหมาะสมของประชากรจากผลรวมคะแนนทั้งหมด

การคัดเลือกแบบแข่งขันแบบเปรียบเทียบ (Tournament) คือการสุ่มจับคู่เปรียบเทียบจากกลุ่มประชากรและคัดเลือกผู้ชนะจากการเปรียบเทียบนั้น ตัวอย่างเช่นมีประชากรในรุ่นนี้ทั้งหมด 8 ประชากรดังที่ได้แสดงไว้ในรูปที่ 2.26 และต้องการเลือกโครโมโซมที่ดีที่สุดสองโครโมโซมเพื่อใช้ในการสืบทอดสายพันธุ์ โดยรูปที่ 2.27 นั้นจะแสดงให้เห็นถึงการจับคู่เปรียบเทียบในแต่ละรอบโดยการพิจารณาว่า ค่าฟังก์ชันความเหมาะสมของประชากรตัวใดดีกว่ากัน (มีค่าน้อยกว่าจะถือว่าดีกว่า ตามตัวอย่างที่ได้กล่าวไว้ข้างต้น) จึงจะสามารถเข้าสู่การแข่งขันในรอบถัดไปและจะมีการตรวจสอบเปรียบเทียบค่าฟังก์ชันความเหมาะสมในแต่ละรอบจนกว่าจะได้ประชากรที่ชนะและเหลือรอด 2 ประชากรตามที่กำหนดไว้

	Gene#1	Gene#2	Gene#3	Gene#4	Gene#5	Fitness value
Chromosome #1 :	0	1	1	0	1	169
Chromosome #2 :	1	0	0	1	1	361
Chromosome #3 :	0	1	0	0	0	64
Chromosome #4 :	0	0	1	0	0	16
Chromosome #5 :	0	1	0	0	1	81
Chromosome #6 :	0	0	0	0	1	1
Chromosome #7 :	0	0	0	1	1	9
Chromosome #8 :	1	1	1	1	1	961

รูปที่ 2.26 แสดงตัวอย่างกลุ่มประชากรที่ถูกสุ่มมาเพื่อเข้าร่วมการคัดเลือกแบบทัวร์นาเมนต์



รูปที่ 2.27 การแข่งขันแบบ Tournament เพื่อให้ได้ประชากรที่เหมาะสมที่สุด 2 ประชากร

การคัดเลือกแบบ Linear Ranking เป็นการคัดเลือกโดยการจัดอันดับคะแนนความเหมาะสมของประชากรและกำหนดความน่าจะเป็นในการถูกคัดเลือกตามการจัดอันดับนั้น ตัวอย่างประกอบคำอธิบายจะเป็นชุดข้อมูลประชากรชุดเดิม 8 ประชากรดังที่ได้แสดงไว้แล้วในรูปที่ 2.26 ส่วนเงื่อนไขคือค่า Fitness Value ที่น้อยที่สุดสองอันดับแรกโดยหลังจากเรียงลำดับตามค่าความเหมาะสมแล้วจะได้ลำดับของประชากรใหม่ตามรูปที่ 2.28 โดยประชากรที่ดีที่สุดสองอันดับแรกได้แก่ Chromosome#6 และ Chromosome#7 ซึ่งมีค่าความเหมาะสมเท่ากับ 1 และ 9 ตามลำดับ

	Gene#1	Gene#2	Gene#3	Gene#4	Gene#5	Fitness value
Chromosome #6 :	0	0	0	0	1	1
Chromosome #7 :	0	0	0	1	1	9
Chromosome #4 :	0	0	1	0	0	16
Chromosome #3 :	0	1	0	0	0	64
Chromosome #5 :	0	1	0	0	1	81
Chromosome #1 :	0	1	1	0	1	169
Chromosome #2 :	1	0	0	1	1	361
Chromosome #8 :	1	1	1	1	1	961

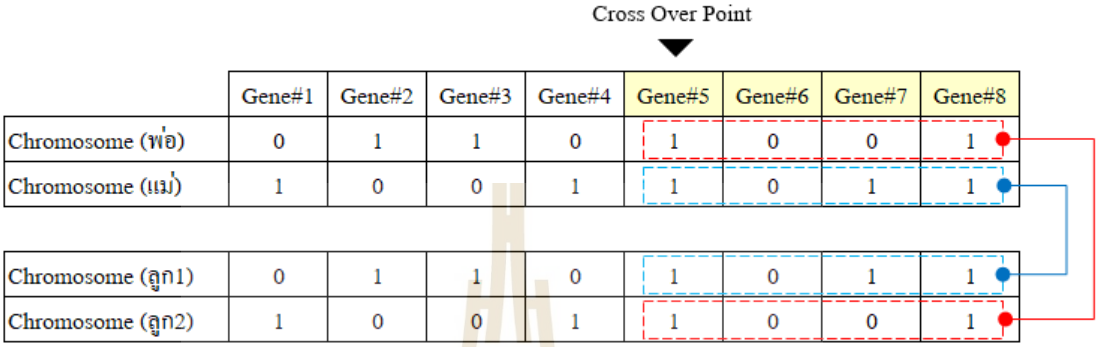
รูปที่ 2.28 แสดงการเรียงลำดับตามค่าความเหมาะสมของประชากรทั้ง 8

จะเห็นได้ว่าจากตัวอย่างประชากรกลุ่มเดียวกันแต่ใช้วิธีการในการคัดเลือกที่แตกต่างกันจะให้ผลลัพธ์ที่แตกต่างกันออกไปซึ่งความแตกต่างจะมากหรือน้อยก็ขึ้นอยู่กับลักษณะข้อมูล

2. การแลกเปลี่ยนพันธุกรรม เป็นหนึ่งในตัวดำเนินการทางพันธุกรรมที่สำคัญที่ทำให้เกิดการเปลี่ยนแปลงลักษณะของประชากรให้มีความหลากหลายมากขึ้น ในการแลกเปลี่ยนพันธุกรรมจะเป็นการนำประชากรที่ผ่านการคัดเลือกสำหรับการสืบทอดสายพันธุ์มาจับคู่ โดยกำหนดให้สมาชิกที่ถูกคัดเลือกมานั้นเป็นรุ่นพ่อ-แม่ (Parent Individual) แล้วจึงนำโครโมโซมจากรุ่นพ่อ-แม่มาผสมกันเพื่อให้กำเนิดโครโมโซมใหม่สำหรับรุ่นถัดไปเพื่อหาลักษณะทางพันธุกรรมที่มีความเหมาะสมกว่า วิธีการการแลกเปลี่ยนพันธุกรรมนั้นโดยทั่วไปแล้วจะแบ่งออกเป็น 2 ชนิด ได้แก่ การแลกเปลี่ยนพันธุกรรมแบบจุดเดียว (Single-Point Crossover) และการแลกเปลี่ยนพันธุกรรมแบบหลายจุด (Multiple-Point Crossover)

การแลกเปลี่ยนพันธุกรรมแบบจุดเดียว จะเริ่มจากการกำหนดจุดที่ต้องการจะเริ่มสลับโครโมโซมแล้วจากนั้นจึงทำการสลับข้อมูลในโครโมโซมตั้งแต่จุดนั้นเป็นต้นไป ตัวอย่างการแลกเปลี่ยนพันธุกรรมแบบจุดเดียวได้ถูกแสดงไว้ในรูปแบบแผนภาพรูปที่ 2.29 โดยตัวอย่างนี้จะมีจุดแลกเปลี่ยนพันธุกรรม (Cross Over Point) ที่ยีนลำดับที่ 5 (Gene#5) โครโมโซมของลูก1 จะรับข้อมูลพันธุกรรมที่ 1 ถึง 4 มาจากข้อมูลของโครโมโซมพ่อ แล้วจึงทำการสลับข้อมูลพันธุกรรมไปใช้โครโมโซมของแม่ตั้งแต่ตำแหน่งที่ 5 เป็นต้นไป เช่นเดียวกันกับโครโมโซมของลูก 2 จะได้รับข้อมูลพันธุกรรมที่ 1 ถึง 4 จากโครโมโซมแม่ และ ในตำแหน่งที่ 5 ถึง 8 จะใช้โครโมโซมของพ่อ

การแลกเปลี่ยนพันธุกรรมแบบหลายจุดนั้นจะมีหลักการเช่นเดียวกันกับการแลกเปลี่ยนพันธุกรรมแบบจุดเดียว เพียงแต่มีจุดในการทำการสลับข้อมูลพันธุกรรมมากกว่าหนึ่งจุด โดยการเลือกจุดนิยมใช้วิธีการแบบสุ่มเพื่อให้เกิดความหลากหลายทางพันธุกรรมมากยิ่งขึ้น



รูปที่ 2.29 ตัวอย่างแสดงการแลกเปลี่ยนพันธุกรรม

3. การกลายพันธุ์ เป็นอีกหนึ่งวิธีที่เพิ่มความหลากหลายของพันธุกรรมในรุ่นลูกรุ่นหลาน ซึ่งจะเป็นการเพิ่มโอกาสในการค้นหาคำตอบของปัญหาได้มากขึ้น เนื่องจากการคัดเลือกโครโมโซมในกระบวนการสืบพันธุ์และแลกเปลี่ยนพันธุกรรมเป็นการอาศัยชุดข้อมูลจากโครโมโซมเดิมที่มีอยู่แล้ว ซึ่งการกลายพันธุ์อาจจะทำให้พบโครโมโซมใหม่ที่ดีกว่านอกเหนือจากโครโมโซมของประชากรดั้งเดิม โดยการกลายพันธุ์ที่นิยมใช้อย่างมากในข้อมูลโครโมโซมที่เป็นเลขฐานสอง ได้แก่ การกลับบิตโดยจะเริ่มจากการกำหนดจุดที่ต้องการกลายพันธุ์ แล้วจึงทำการแทนที่ด้วยค่าตรงกันข้าม (Complement)

2.10.4 การแทนที่ด้วยประชากรรุ่นลูกที่ดีกว่า (Replacement by Better Offspring)

ขั้นตอนนี้จะเป็นขั้นตอนที่ได้รับโครโมโซมของรุ่นลูกที่ผ่านการดำเนินการทางพันธุกรรมเรียบร้อยแล้ว โครโมโซมของรุ่นลูกที่ได้มาใหม่นั้นจะถูกนำไปเพื่อพิจารณาแทนที่โครโมโซมของรุ่นพ่อ-แม่ ด้วยวัตถุประสงค์ในการให้กำเนิดสายพันธุ์ใหม่ที่มีความเหมาะสมกับคำตอบของปัญหามากขึ้น การแทนที่ด้วยประชากรรุ่นลูกนั้นอาจจะเป็นการแทนที่โดยใช้ประชากรรุ่นลูกทั้งหมดแทนที่ประชากรรุ่นพ่อ-แม่ทั้งหมด ที่ถูกเรียกว่าการแทนที่ด้วยประชากรทั้งรุ่น (Generation Genetic Algorithm) นอกเหนือจากนั้นแล้วยังมีการแทนที่ด้วยประชากรบางส่วน (Partial Genetic Algorithm) ซึ่งเป็นวิธีการที่จะนำค่าความเหมาะสมของประชากรรุ่นพ่อ-แม่มาทำการพิจารณาเปรียบเทียบกับค่าความเหมาะสมของประชากรรุ่นลูกด้วย โดยอาจจะเป็นการเก็บประชากรรุ่นพ่อ-แม่เฉพาะตัวที่ดีที่สุดไว้ หรืออาจจะเป็นการแทนที่ประชากรรุ่นพ่อ-แม่เฉพาะตัวที่

มีค่าความเหมาะสมที่แย่สุดก็ได้ และวิธีการสุดท้ายในการเลือกการแทนที่ ได้แก่วิธีการสุ่มเลือกซึ่งจะไม่เป็นที่นิยมมากนักเนื่องจากอาจจะทำให้เกิดการแทนที่ประชากรรุ่นพ่อแม่ด้วยประชากรรุ่นลูกที่มีค่าความเหมาะสมที่แย่กว่า

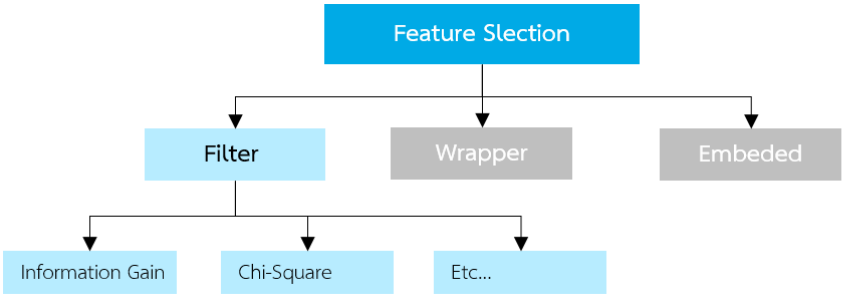
2.10.5 การตรวจสอบเงื่อนไขในการจบการทำงาน (Termination Condition)

กระบวนการทั้งหลายที่ได้กล่าวมาในข้างต้นนี้ ได้แก่ การประเมินค่าด้วยฟังก์ชันความเหมาะสม, การดำเนินการทางขั้นตอนวิธีเชิงพันธุกรรม และการแทนที่ด้วยประชากรรุ่นลูกหลาน จะทำงานแบบวนซ้ำไปเรื่อย ๆ (Lamichhaney et al., 2015; Marsh, 2015; Soons et al., 2010; Podos and Nowicki, 2004) จนกว่าจะบรรลุตามเงื่อนไขในการจบการทำงาน โดยเงื่อนไขการจบการทำงานนั้นจะมีหลายรูปแบบให้เลือกใช้ตามความเหมาะสมของลักษณะงานดังนี้

1. ได้รับผลลัพธ์ที่ประชากรทั้งหมดอยู่ในเกณฑ์ที่กำหนดไว้แล้ว
2. การทำซ้ำดำเนินการมาจนถึงประชากรรุ่นสุดท้ายที่ได้กำหนดไว้แล้ว
3. ทรัพยากรที่ใช้ในการคำนวณหมดแล้ว (ยกตัวอย่างเช่น การกำหนดเวลาที่ใช้ในการเรียนรู้)
4. ได้รับผลลัพธ์ที่ประชากรทั้งหมดมีความเหมาะสมอยู่ในระดับสูงสุดแล้ว
5. ตรวจสอบเงื่อนไขด้วยผู้ทำการทดลองสั่งหยุดเอง
6. ประยุกต์การตรวจสอบด้วยเงื่อนไขต่าง ๆ ด้านบนเข้าด้วยกัน

2.11 วิธีการทางสถิติสำหรับการคัดเลือกคุณลักษณะ

การหาความสัมพันธ์ระหว่างตัวแปรต้นและตัวแปรตามโดยใช้วิธีการทางสถิติเป็นอีกหนึ่งทางเลือกที่นิยมนำมาประยุกต์ใช้งานกับการแก้ปัญหาด้านต่าง ๆ และยังพบอีกว่ามีการนำไปใช้ในการคัดเลือกคุณลักษณะอีกด้วย โดยในงานวิจัยนี้ได้ทำการศึกษาวิธีการทางสถิติ 2 วิธีการ ได้แก่ Information Gain และ Chi-Squre ซึ่งจัดอยู่ในหมวดหมู่ของการคัดเลือกคุณลักษณะแบบ Filter Method ตามที่ได้แสดงไว้ในรูปแบบของแผนผัง ตามรูปที่ 2.30



รูปที่ 2.30 แผนผังแสดงตำแหน่งที่อยู่ของ Informatation และ Chi-Square ในการแบ่งประเภทเทคนิคในการคัดเลือกคุณลักษณะ

2.11.1 Information Gain

การคัดเลือกคุณลักษณะโดยใช้ Information Gain เป็นหนึ่งในการประยุกต์ความรู้ทางสถิติเข้ามาใช้งานร่วมกับการเรียนรู้ของเครื่อง โดย Information Gain เป็นวิธีการที่ได้รับอิทธิพลมาจากทฤษฎีการสื่อสาร (Communication Theory) ที่ตีพิมพ์ในหนังสือ A Mathematical Theory of Communication (Shannon, 2001) ของ Claude E. Shannon วิศวกรและนักวิจัยที่มีบทบาทอย่างมากในการปฏิวัติการสื่อสารโลกในช่วงศตวรรษที่ 19

โดยในส่วนของทฤษฎีการสื่อสาร Shannon ได้กล่าวไว้ว่า สัญญาณข้อมูล (Information Signal) และความคลุมเครือ หรือ Information Entropy มีความสัมพันธ์กันตามสมการดังต่อไปนี้

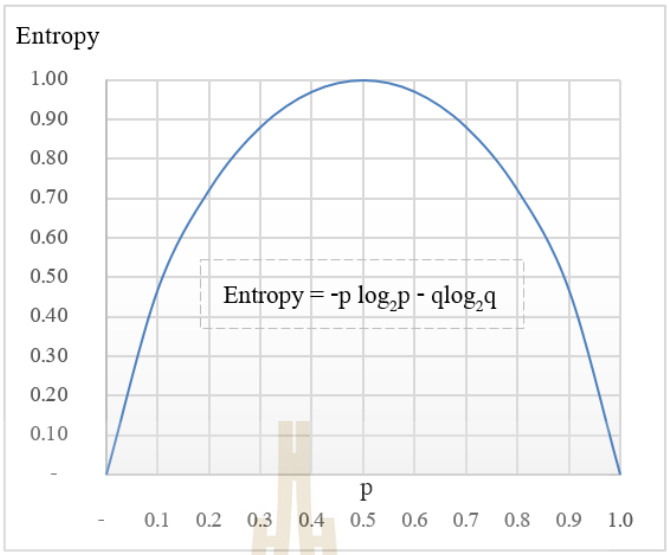
$$R = H(x) - Hy(x) \tag{2.14}$$

โดย $H(x)$ คือ ข้อมูลทั้งหมดที่มี

$Hy(x)$ คือ ฟังก์ชันผลเฉลี่ยของค่าความคลุมเครือของข้อมูล

R คือ สัญญาณข้อมูลที่ได้รับ

การนำวิธีการ Information Gain มาประยุกต์ใช้ในการคัดเลือกคุณลักษณะนั้น (Jadhav et al., 2018; Nowozin, 2012; Lee and Lee, 2006; Roobaert et al., 2006; Kent, 1983) จะเป็นการคำนวณค่า Information Entropy ซึ่งเป็นการวัดความแตกต่างของข้อมูล หากข้อมูลมีความแตกต่างกันมากค่า Entropy จะมีค่าสูง ในทางตรงข้ามหากข้อมูลมีความเป็นเนื้อเดียวกัน (Homogenous) ที่มาก ค่า Entropy ก็จะมีค่าต่ำ ลักษณะของ Entropy แสดงดังในรูปที่ 2.31



รูปที่ 2.31 กราฟแสดงการเปลี่ยนแปลงของค่า Entropy ที่แปรผันตามความเป็นเนื้อเดียวกันของข้อมูล (Homogenous)

การคำนวณหา Information Gain สามารถหาได้จากการคำนวณโดยสมการดังต่อไปนี้

$$H(X) = -\sum_{i=1}^{n} P(x_i) \log P(x_i) \qquad (2.15)$$

โดยที่ $H(X)$ คือ Information Gain

Entropy (x_i) คือ $-P(x_i) \log P(x_i)$ และ

$P(x_i)$ คือ ค่าความน่าจะเป็น (Probability) ของเหตุการณ์ (x_i)

ตัวอย่างแสดงดังรูปที่ 2.32 จะนำมาใช้อธิบายในส่วนของการคำนวณ เพื่อประยุกต์หลักการของ Information Gain ในการคัดเลือกคุณลักษณะจากข้อมูลการผลิตฮาร์ดดิสก์ไดรฟ์โดยชุดข้อมูลตั้งต้นจะประกอบด้วย 6 คุณลักษณะดังนี้

1) Drive SN (Drive Serial Number) คือ หมายเลขแสดงถึงฮาร์ดดิสก์ไดรฟ์แต่ละตัว โดยหมายเลขเหล่านี้จะไม่ซ้ำกัน

2) STATUS คือ สถานะการทดสอบที่แสดงว่าฮาร์ดดิสก์ไดรฟ์ตัวนั้น ๆ ผ่าน (Pass) หรือไม่ผ่าน (Fail) การทดสอบผลิตภัณฑ์

3) HSA PR แสดงถึงสถานะภาพของ HSA (Head Stack Assembly) ว่าเป็นของใหม่ (Prime) หรือของที่เคยใช้งานมาแล้ว (RCY, Recycled)

4) Media PR แสดงถึงสถานะภาพของ Media ว่าเป็นของใหม่ หรือของที่เคยใช้งานมาแล้ว

5) MBA PR แสดงถึงสถานะภาพของ MBA (Motor Base Assembly) ว่าเป็นของใหม่ หรือของที่เคยใช้งานมาแล้ว

6) PCBA PR แสดงถึงสถานะภาพของ PCBA (Printed Circuit Board Assembly) ว่าเป็นของใหม่ หรือของที่เคยใช้งานมาแล้ว

(ก) ตัวอย่างข้อมูลตั้งต้น

Drive SN	STATUS	HSA PR	MEDIA PR	MBA PR	PCBA PR
SN-01	Pass	Prime	Prime	Prime	RCY
SN-02	Pass	Prime	Prime	Prime	RCY
SN-03	Fail	Prime	RCY	Prime	Prime
SN-04	Fail	RCY	RCY	Prime	Prime
SN-05	Pass	Prime	Prime	Prime	Prime
SN-06	Pass	Prime	Prime	Prime	Prime
SN-07	Pass	Prime	Prime	Prime	Prime
SN-08	Pass	Prime	Prime	Prime	Prime
SN-09	Pass	Prime	Prime	Prime	Prime
SN-10	Pass	Prime	Prime	Prime	Prime
SN-11	Fail	Prime	RCY	RCY	RCY
SN-12	Pass	Prime	Prime	Prime	RCY
SN-13	Pass	Prime	Prime	Prime	Prime
SN-14	Pass	Prime	Prime	Prime	Prime
SN-15	Pass	Prime	Prime	Prime	Prime
SN-16	Pass	Prime	Prime	Prime	Prime
SN-17	Pass	Prime	Prime	Prime	RCY
SN-18	Pass	Prime	Prime	Prime	Prime
SN-19	Pass	Prime	Prime	RCY	Prime
SN-20	Fail	RCY	Prime	Prime	Prime

(ข) ความน่าจะเป็นของ STATUS

Pass Probalility =	0.80
Fail Probability =	0.20

รูปที่ 2.32 (ก) ตัวอย่างข้อมูลตั้งต้นที่จะมาใช้แสดงการคำนวณ Information Gain และ (ข) ค่าความน่าจะเป็นของ Staus

จากตัวอย่างข้างต้น จะมีขั้นตอนการคำนวณหา Information Gain ได้ดังต่อไปนี้

1) คำนวณหา Entropy เริ่มต้น

2) คำนวณหา Information Gain สำหรับ HSA PR (รูปที่ 2.33)

3) คำนวณหา Information Gain สำหรับ Media PR (รูปที่ 2.34)

4) คำนวณหา Information Gain สำหรับ MBA PR (รูปที่ 2.35)

5) คำนวณหา Information Gain สำหรับ PCBA PR (รูปที่ 2.36)

1) การคำนวณหา Entropy เริ่มต้น

$$\text{Entropy}_{(\text{Initial})} = -P_{(\text{Status = Pass})} * \text{Log}_2 P_{(\text{Status = Pass})} - P_{(\text{Status = Fail})} * \text{Log}_2 P_{(\text{Status = Fail})}$$

$$= -(0.8 * \text{Log}_2(0.8) + 0.2 * \text{Log}_2(0.2))$$

$$= 0.72$$

(ก) ตัวอย่างข้อมูลเมื่อพิจารณาเฉพาะ HSA PR

Drive SN	STATUS	HSA PR
SN-01	Pass	Prime
SN-02	Pass	Prime
SN-05	Pass	Prime
SN-06	Pass	Prime
SN-07	Pass	Prime
SN-08	Pass	Prime
SN-09	Pass	Prime
SN-10	Pass	Prime
SN-12	Pass	Prime
SN-13	Pass	Prime
SN-14	Pass	Prime
SN-15	Pass	Prime
SN-16	Pass	Prime
SN-17	Pass	Prime
SN-18	Pass	Prime
SN-19	Pass	Prime
SN-03	Fail	Prime
SN-11	Fail	Prime
SN-04	Fail	RCY
SN-20	Fail	RCY

(ข) ค่าต่าง ๆ ที่ได้จากการคำนวณเมื่อพิจารณาเฉพาะ HSA PR

P (HSA = Prime)	0.90
P (HSA = RCY)	0.10
P (HSA = Prime) & Pass	0.89
P (HSA = Prime) & Fail	0.11
P (HSA = RCY) & Pass	0.00
P (HSA =RCY) & Fail	1.00
Entropy HSA Prime	0.50
Entropy HSA RCY	0.00
Information Gain	0.27

รูป 2.33 (ก) ตัวอย่างข้อมูลเพื่อการคำนวณค่า Information Gain ของ HSA PR และ (ข) ค่าที่ได้จากการคำนวณเมื่อพิจารณาเฉพาะคุณลักษณะ HSA PR

2) คำนวณหา Information Gain สำหรับ HSA PR

$$\text{Information Gain}_{(\text{HSA PR})} = \text{Entropy}_{(\text{Initial})} - [P_{(\text{HSA = Prime})} * \text{Entropy}_{(\text{HSA = Prime})} + P_{(\text{HSA = RCY})} * \text{Entropy}_{(\text{HSA = RCY})}]$$

$$= 0.72 - [0.90 * 0.50 + 0.10 * 0.00]$$

$$= 0.72 - 0.45$$

$$= 0.27$$

(ก) ตัวอย่างข้อมูลเมื่อพิจารณาเฉพาะ MEDIA PR

Drive SN	STATUS	MEDIA PR
SN-01	Pass	Prime
SN-02	Pass	Prime
SN-05	Pass	Prime
SN-06	Pass	Prime
SN-07	Pass	Prime
SN-08	Pass	Prime
SN-09	Pass	Prime
SN-10	Pass	Prime
SN-12	Pass	Prime
SN-13	Pass	Prime
SN-14	Pass	Prime
SN-15	Pass	Prime
SN-16	Pass	Prime
SN-17	Pass	Prime
SN-18	Pass	Prime
SN-19	Pass	Prime
SN-20	Fail	Prime
SN-03	Fail	RCY
SN-04	Fail	RCY
SN-11	Fail	RCY

(ข) ค่าต่าง ๆ ที่ได้จากการคำนวณเมื่อพิจารณาเฉพาะ MEDIA PR

P (MEDIA = Prime)	0.85
P (MEDIA = RCY)	0.15
P (MEDIA = Prime) & Pass	0.94
P (MEDIA = Prime) & Fail	0.06
P (MEDIA = RCY) & Pass	0.00
P (MEDIA =RCY) & Fail	1.00
Entropy MEDIA Prime	0.32
Entropy MEDIA RCY	0.00
Information Gain	0.45

รูป 2.34 (ก) ตัวอย่างข้อมูลเพื่อการคำนวณค่า Information Gain ของ MEDIA PR และ (ข) ค่าที่ได้จากการคำนวณเมื่อพิจารณาเฉพาะคุณลักษณะ MEDIA PR

3) คำนวณหา Information Gain สำหรับ Media PR

$$\text{Information Gain}_{(\text{Media PR})} = \text{Entropy}_{(\text{Initial})} - [P_{(\text{Media = Prime})} * \text{Entropy}_{(\text{Media = Prime})} + P_{(\text{Media = RCY})} * \text{Entropy}_{(\text{Media = RCY})}]$$

$$= 0.72 - [0.85 * 0.32 + 0.15 * 0.00]$$

$$= 0.72 - 0.27$$

$$= 0.45$$

(ก) ตัวอย่างข้อมูลเมื่อพิจารณาเฉพาะ MBA PR

Drive SN	STATUS	MBA PR
SN-01	Pass	Prime
SN-02	Pass	Prime
SN-05	Pass	Prime
SN-06	Pass	Prime
SN-07	Pass	Prime
SN-08	Pass	Prime
SN-09	Pass	Prime
SN-10	Pass	Prime
SN-12	Pass	Prime
SN-13	Pass	Prime
SN-14	Pass	Prime
SN-15	Pass	Prime
SN-16	Pass	Prime
SN-17	Pass	Prime
SN-18	Pass	Prime
SN-03	Fail	Prime
SN-04	Fail	Prime
SN-20	Fail	Prime
SN-19	Pass	RCY
SN-11	Fail	RCY

(ข) ค่าต่าง ๆ ที่ได้จากการคำนวณเมื่อพิจารณาเฉพาะ MBA PR

P (MBA = Prime)	0.90
P (MBA = RCY)	0.10
P (MBA = Prime) & Pass	0.83
P (MBA = Prime) & Fail	0.17
P (MBA = RCY) & Pass	0.50
P (MBA =RCY) & Fail	0.50
Entropy MBA Prime	0.65
Entropy MBA RCY	1.00
Information Gain	0.04

รูป 2.35 (ก) ตัวอย่างข้อมูลเพื่อการคำนวณค่า Information Gain ของ MBA PR และ (ข) ค่าที่ได้จากการคำนวณเมื่อพิจารณาเฉพาะคุณลักษณะ MBA PR

4) คำนวณหา Information Gain สำหรับ MBA PR

$$\text{Information Gain}_{(\text{MBA PR})} = \text{Entropy}_{(\text{Initial})} - [P_{(\text{MBA = Prime})} * \text{Entropy}_{(\text{MBA = Prime})} + P_{(\text{MBA = RCY})} * \text{Entropy}_{(\text{MBA = RCY})}]$$

$$= 0.72 - [0.90 * 0.65 + 0.10 * 1.00]$$

$$= 0.72 - 0.69$$

$$= 0.04$$

(ก) ตัวอย่างข้อมูลเมื่อพิจารณาเฉพาะ PCBA PR

Drive SN	STATUS	PCBA PR
SN-05	Pass	Prime
SN-06	Pass	Prime
SN-07	Pass	Prime
SN-08	Pass	Prime
SN-09	Pass	Prime
SN-10	Pass	Prime
SN-13	Pass	Prime
SN-14	Pass	Prime
SN-15	Pass	Prime
SN-16	Pass	Prime
SN-18	Pass	Prime
SN-19	Pass	Prime
SN-03	Fail	Prime
SN-04	Fail	Prime
SN-20	Fail	Prime
SN-01	Pass	RCY
SN-02	Pass	RCY
SN-12	Pass	RCY
SN-17	Pass	RCY
SN-11	Fail	RCY

(ข) ค่าต่าง ๆ ที่ได้จากการคำนวณเมื่อพิจารณาเฉพาะ PCBA PR

P (PCBA = Prime)	0.75
P (PCBA = RCY)	0.25
P (PCBA = Prime) & Pass	0.80
P (PCBA = Prime) & Fail	0.20
P (PCBA = RCY) & Pass	0.80
P (PCBA =RCY) & Fail	0.20
Entropy PCBA Prime	0.72
Entropy PCBA RCY	0.72
Information Gain	0.00

รูป 2.36 (ก) ตัวอย่างข้อมูลเพื่อการคำนวณค่า Information Gain ของ PCBA PR และ (ข) ค่าที่ได้จากการคำนวณเมื่อพิจารณาเฉพาะคุณลักษณะ PCBA PR

5) คำนวณหา Information Gain สำหรับ PCBA PR

$$\text{Information Gain}_{(\text{PCBA PR})} = \text{Entropy}_{(\text{Initial})} - [P_{(\text{PCBA = Prime})} * \text{Entropy}_{(\text{PCBA = Prime})} + P_{(\text{PCBA = RCY})} * \text{Entropy}_{(\text{PCBA = RCY})}]$$

$$= 0.72 - [0.75*0.72 + 0.25*0.72]$$

$$= 0.72 - 0.72$$

$$= 0.00$$

ตารางที่ 2.11 แสดงค่า Information Gain ของแต่ละคุณลักษณะ

Feature	Information Gain
HSA PR	0.27
MEDIA PR	0.45
MBA PR	0.04
PCBA PR	0.00

ตารางที่ 2.11 เป็นการแสดงค่า Information Gain ของแต่ละคุณลักษณะเพื่อนำไปใช้ในการคัดเลือกคุณลักษณะและนำไปใช้งานในขั้นตอนต่อไป เช่นต้องการคุณลักษณะ 2 คุณลักษณะ ซึ่งคุณลักษณะที่ได้รับเลือกคือ HSA PR และ Media PR

2.11.2 Chi-Square

Chi-Square ไคสแควร์เป็นวิธีการทดสอบทางสถิติวิธีการหนึ่ง (Thaseen and Kumar, 2017; McHugh, 2013; Jin et al., 2006; Janes 2001; Satorra and Bentler, 2001) เพื่อใช้ในการเปรียบเทียบว่าตัวแปร (Variable) หรือคุณลักษณะแต่ละตัวนั้นมีความสัมพันธ์กันมากน้อยเพียงใด หรืออาจจะใช้ทดสอบว่าข้อมูลนั้นมีค่าเป็นไปตามค่าคาดหวัง (Expected Value) หรือไม่ การคัดเลือกคุณลักษณะโดยใช้วิธีการทดสอบของ Chi-Square ประยุกต์ใช้งานนั้นเป็นที่แพร่หลายและสามารถพบได้ในงานวิจัยทั่วไป

ในส่วนของการคำนวณค่า Chi-Square นั้นสามารถอธิบายได้ดังสมการต่อไปนี้

$$X^2 = \sum_{i=1}^{k} \frac{(O_i - E_i)^2}{E_i} \quad (2.16)$$

เมื่อ X^2 คือ ค่า Chi-Sqaure

O_i คือ ค่าความถี่ที่ได้จากการสังเกต

E_i คือ ค่าความถี่ที่คาดหวัง

ตัวอย่างที่จะนำมาใช้อธิบายในส่วนของการคำนวณค่า Chi-Square เป็นข้อมูลชุดเดียวกันกับตัวอย่างการคำนวณของ Information Gain ที่ได้แสดงไว้ข้างต้น ตามรูปที่ 2.32

โดยกระบวนการในการคำนวณสามารถแบ่งออกเป็นขั้นตอนได้ดังนี้

1) การคำนวณหาค่า Chi-Square เมื่อแบ่งกลุ่มข้อมูลตาม HSA PR

2) การคำนวณหาค่า Chi-Square เมื่อแบ่งกลุ่มข้อมูลตาม Media PR

3) การคำนวณหาค่า Chi-Square เมื่อแบ่งกลุ่มข้อมูลตาม MBA PR

4) การคำนวณหาค่า Chi-Square เมื่อแบ่งกลุ่มข้อมูลตาม PCBA PR

(ก) ตัวอย่างข้อมูลเมื่อพิจารณาเฉพาะ HSA PR

Drive SN	STATUS	HSA PR
SN-01	Pass	Prime
SN-02	Pass	Prime
SN-05	Pass	Prime
SN-06	Pass	Prime
SN-07	Pass	Prime
SN-08	Pass	Prime
SN-09	Pass	Prime
SN-10	Pass	Prime
SN-12	Pass	Prime
SN-13	Pass	Prime
SN-14	Pass	Prime
SN-15	Pass	Prime
SN-16	Pass	Prime
SN-17	Pass	Prime
SN-18	Pass	Prime
SN-19	Pass	Prime
SN-03	Fail	Prime
SN-11	Fail	Prime
SN-04	Fail	RCY
SN-20	Fail	RCY

(ข) ค่าต่าง ๆ ที่ได้จากจากคำนวณตัวอย่างข้อมูล เมื่อพิจารณาเฉพาะ HSA PR

	Prime	RCY
Staus=Pass	16	0
Status = Fail	2	2
Total	18	2
Expected = Pass	14.4	1.6
Expected = Fail	3.6	0.4
Deviation (Pass)	1.6	-1.6
Deviation (Fail)	-1.6	1.6
Chi-Square (Pass)	0.422	1.265
Chi-Square (Fail)	0.843	2.530
Chi-Square of HSA PR	5.060	

รูปที่ 2.37 (ก) แสดงตัวอย่างการคำนวณค่า Chi-Square ของ HSA PR และ
(ข) ค่าที่ได้จากการคำนวณเมื่อพิจารณาเฉพาะคุณลักษณะ HSA PR

การคำนวณค่า Chi-Square ในตัวอย่างข้างต้นจะเริ่มจากการนับประชากรของทั้ง 4 กลุ่ม ได้แก่ HSA PR Prime Pass, HSA PR Prime Fail, HSA PR RCY Pass และ HSA PR RCY Fail โดยจะได้ค่า 16, 2, 0 และ 2 ตามลำดับซึ่งได้แสดงไว้ในรูปที่ 2.37 (ข) จากนั้นจึงทำการคำนวณหา คือ Expected = Pass และ Expected Fail ของ HSA PR เมื่อมีค่าเป็น Prime และ RCY ตามการคำนวณดังต่อไปนี้

$$\text{Expected Pass}_{(\text{HSA Prime})} = \text{Pass Proability} * \text{Total Count of HSA Prime}$$

$$= 0.8 * 18$$

$$= 14.4$$

Expected Fail $_{(HSA\ Prime)}$ = Fail Proability * Total Count of HSA Prime

= 0.2 * 18

= 3.6

Expected Pass $_{(HSA\ RCY)}$ = Pass Proability * Total Count of HSA RCY

= 0.8 * 2

= 1.6

Expected Fail $_{(HSA\ RCY)}$ = Fail Proability * Total Count of HSA RCY

= 0.2 * 2

= 0.4

เมื่อคำนวณค่าคาดหวังในแต่ละกรณีเสร็จเรียบร้อยแล้ว ขั้นตอนถัดไปคือการหาค่าความคลาดเคลื่อน (Deviation) ของข้อมูลจริงกับค่าความคาดหวัง ของแต่ละกรณีซึ่งได้แสดงไว้ในรูปที่ 2.37 (ข) จากนั้นจึงเป็นการคำนวณค่าของ Chi-Square ทั้ง 4 กรณีตามการคำนวณดังต่อไปนี้

Chi-Square $_{(HSA\ Prime\ Pass)}$ = Sqrt [(Deviation^2 $_{(HSA\ Prime\ Pass)}$ / Expected $_{(HSA\ Prime\ Pass)}$]

= Sqrt [1.6^2 / 14.4]

= 0.422

$$\text{Chi-Square}_{(\text{HSA Prime Fail})} = \text{Sqrt}\left[(\text{Deviation}^2_{(\text{HSA Prime Fail})} / \text{Expected}_{(\text{HSA Prime Fail})}\right]$$

$$= \text{Sqrt}\left[(-1.6)^2 / 3.6\right]$$

$$= 0.843$$

$$\text{Chi-Square}_{(\text{HSA RCY Pass})} = \text{Sqrt}\left[(\text{Deviation}^2_{(\text{HSA RCY Pass})} / \text{Expected}_{(\text{HSA RCY Pass})}\right]$$

$$= \text{Sqrt}\left[(-1.6)^2 / 1.6\right]$$

$$= 1.265$$

$$\text{Chi-Square}_{(\text{HSA RCY Fail})} = \text{Sqrt}\left[(\text{Deviation}^2_{(\text{HSA RCY Fail})} / \text{Expected}_{(\text{HSA RCY Fail})}\right]$$

$$= \text{Sqrt}\left[1.6^2 / 0.4\right]$$

$$= 2.530$$

ขั้นตอนสุดท้ายสำหรับการหาค่า Chi-Square ของ HSA PR คือการนำผลรวมของค่า Chi-Square ทั้ง 4 กรณีข้างต้นเข้าด้วยกัน ซึ่งมีค่าเท่ากับ 5.060 ดังที่ได้แสดงไว้ในรูป 2.37 โดยตัวอย่างการคำนวณค่าการนับประชากร ค่าคาดหวัง ค่าความคลาดเคลื่อน และค่า Chi-Square ของคุณลักษณะ Media PR MBAPR และ PCBA PR นั้นได้แสดงรายละเอียดไว้ในรูปที่ 2.38 (ข) รูปที่ 2.39 (ข) และ รูปที่ 2.40 (ข) ตามลำดับ

(ก) ตัวอย่างข้อมูลเมื่อพิจารณาเฉพาะ MEDIA PR

Drive SN	STATUS	MEDIA PR
SN-01	Pass	Prime
SN-02	Pass	Prime
SN-05	Pass	Prime
SN-06	Pass	Prime
SN-07	Pass	Prime
SN-08	Pass	Prime
SN-09	Pass	Prime
SN-10	Pass	Prime
SN-12	Pass	Prime
SN-13	Pass	Prime
SN-14	Pass	Prime
SN-15	Pass	Prime
SN-16	Pass	Prime
SN-17	Pass	Prime
SN-18	Pass	Prime
SN-19	Pass	Prime
SN-20	Fail	Prime
SN-03	Fail	RCY
SN-04	Fail	RCY
SN-11	Fail	RCY

(ข) ค่าต่าง ๆ ที่ได้จากจากคำนวณตัวอย่างข้อมูล เมื่อพิจารณาเฉพาะ MEDIA PR

	Prime	RCY
Staus=Pass	16	0
Status = Fail	1	3
Total	17	3
Expected = Pass	13.6	2.4
Expected = Fail	3.4	0.6
Deviation (Pass)	2.4	-2.4
Deviation (Fail)	-2.4	2.4
Chi-Square (Pass)	0.651	1.549
Chi-Square (Fail)	1.302	3.098
Chi-Square of MEDIA PR	6.600	

รูปที่ 2.38 (ก) แสดงตัวอย่างการคำนวณค่า Chi-Square ของ MEDIA PR และ
(ข) ค่าที่ได้จากการคำนวณเมื่อพิจารณาเฉพาะคุณลักษณะ MEDIA PR

(ก) ตัวอย่างข้อมูลเมื่อพิจารณาเฉพาะ MBA PR

Drive SN	STATUS	MBA PR
SN-01	Pass	Prime
SN-02	Pass	Prime
SN-05	Pass	Prime
SN-06	Pass	Prime
SN-07	Pass	Prime
SN-08	Pass	Prime
SN-09	Pass	Prime
SN-10	Pass	Prime
SN-12	Pass	Prime
SN-13	Pass	Prime
SN-14	Pass	Prime
SN-15	Pass	Prime
SN-16	Pass	Prime
SN-17	Pass	Prime
SN-18	Pass	Prime
SN-03	Fail	Prime
SN-04	Fail	Prime
SN-20	Fail	Prime
SN-19	Pass	RCY
SN-11	Fail	RCY

(ข) ค่าต่าง ๆ ที่ได้จากจากคำนวณตัวอย่างข้อมูล เมื่อพิจารณาเฉพาะ MBA PR

	Prime	RCY
Staus=Pass	15	1
Status = Fail	3	1
Total	18	2
Expected = Pass	14.4	1.6
Expected = Fail	3.6	0.4
Deviation (Pass)	0.6	-0.6
Deviation (Fail)	-0.6	0.6
Chi-Square (Pass)	0.158	0.474
Chi-Square (Fail)	0.316	0.949
Chi-Square of MBA PR	1.897	

รูปที่ 2.39 (ก) แสดงตัวอย่างการคำนวณค่า Chi-Square ของ MBA PR และ
(ข) ค่าที่ได้จากการคำนวณเมื่อพิจารณาเฉพาะคุณลักษณะ MBA PR

(ก) ตัวอย่างข้อมูลเมื่อพิจารณาเฉพาะ PCBA PR

Drive SN	STATUS	PCBA PR
SN-05	Pass	Prime
SN-06	Pass	Prime
SN-07	Pass	Prime
SN-08	Pass	Prime
SN-09	Pass	Prime
SN-10	Pass	Prime
SN-13	Pass	Prime
SN-14	Pass	Prime
SN-15	Pass	Prime
SN-16	Pass	Prime
SN-18	Pass	Prime
SN-19	Pass	Prime
SN-03	Fail	Prime
SN-04	Fail	Prime
SN-20	Fail	Prime
SN-01	Pass	RCY
SN-02	Pass	RCY
SN-12	Pass	RCY
SN-17	Pass	RCY
SN-11	Fail	RCY

(ข) ค่าต่าง ๆ ที่ได้จากจากคำนวณตัวอย่างข้อมูล เมื่อพิจารณาเฉพาะ PCBA PR

	Prime	RCY
Staus=Pass	12	4
Status = Fail	3	1
Total	15	5
Expected = Pass	12	4
Expected = Fail	3	1
Deviation (Pass)	0	0
Deviation (Fail)	0	0
Chi-Square (Pass)	0.000	0.000
Chi-Square (Fail)	0.000	0.000
Chi-Square of PCBA PR	0.000	

รูปที่ (ก) 2.40 แสดงตัวอย่างการคำนวณค่า Chi-Square ของ PCBA PR และ
(ข) ค่าที่ได้จากการคำนวณเมื่อพิจารณาเฉพาะคุณลักษณะ PCBA PR

ตารางที่ 2.12 แสดงค่า Chi-Square ของแต่ละคุณลักษณะ

Feature	**Information Gain**
HSA PR	5.060
MEDIA PR	6.600
MBA PR	1.897
PCBA PR	0.000

จากตัวอย่างข้อมูลและการการคำนวณที่ได้แสดงไว้ในข้างต้นนั้น สามารถสรุปออกมาเป็นตารางที่ 2.12 ซึ่งแสดงการเปรียบเทียบค่า Chi-Square ของแต่ละคุณลักษณะ พบว่าถ้าต้องการคัดเลือกคุณลักษณะที่ดีที่สุดสองตัวเพื่อนำไปใช้ในขั้นตอนต่อไป จะได้ MEDIA PR และ HSA PR ซึ่งให้ค่า Chi-Sqaure ที่ 6.600 และ 5.060 ตามลำดับ

2.12 โครงข่ายประสาทเทียม (ANN: Artificial Neural Network)

โครงข่ายประสาทเทียม เป็นอัลกอริทึมที่อาศัยการเลียนแบบการทำงานของระบบประสาทในสมองของมนุษย์ (Jain et al., 1996; Zhang et al., 1998; Yao, 1999; Bertinetto et al., 2016; Chae et al., 2016) ซึ่งในสมองของมนุษย์ประกอบไปด้วยหน่วยเซลล์ประสาทเล็ก ๆ จำนวนมากที่เชื่อมต่อกันเป็นโครงข่ายประสาทที่มีขนาดใหญ่ โดยมีรายละเอียดและความซับซ้อนที่ทำให้สามารถประมวลผลได้มีประสิทธิภาพมากขึ้นตามจำนวนของเซลล์ประสาทที่เชื่อมต่อกัน โดยแสดงให้อยู่ในรูปแบบของสมการทางคณิตศาสตร์

แนวคิดเริ่มต้นเกิดจากการที่นักวิจัยทางด้านปัญญาประดิษฐ์ต้องการจำลองกลไกความฉลาดของมนุษย์ ได้แก่ การจดจำ เรียนรู้ คิด วิเคราะห์ แยกแยะสิ่งต่าง ๆ ให้กับคอมพิวเตอร์ ตัวอย่างเช่น การตั้งคำถามว่า ทำไมมนุษย์จึงรับรู้ได้ว่าสัตว์ชนิดนี้คือแมว สัตว์ชนิดนี้คือสุนัข ทั้ง ๆ ที่มนุษย์ไม่เคยเห็นแมวและสุนัขทุกตัวบนโลกใบนี้ แต่เพียงแค่ได้เห็นครั้งแรกก็สามารถบอกได้ว่าสัตว์ที่เห็นนี้เป็นสัตว์ชนิดใด ซึ่งหากย้อนกลับไปในตอนที่ยังมนุษย์ยังเป็นเด็กตัวเล็ก ๆ ที่ไม่สามารถแยกแยะหรือรับรู้สิ่งเหล่านี้ได้ดีนัก นั่นเพราะยังมีประสบการณ์ที่ไม่มากพอ ดังนั้นหากมนุษย์มีประสบการณ์มากขึ้น เห็นสิ่งต่าง ๆ มากขึ้น ก็จะสามารถบอกได้ว่าสิ่งที่เห็นคือสิ่งใด นักวิจัยทางด้านปัญญาประดิษฐ์จึงจำลองกลไกการทำงานของสมองมนุษย์โดยเริ่มจากเซลล์ประสาทหน่วยเล็ก ๆ เพื่อทำงานง่าย ๆ และไม่ซับซ้อน จากนั้นเชื่อมต่อหน่วยเล็ก ๆ เหล่านี้เข้าด้วยกันให้เป็นเครือข่ายที่ใหญ่และสามารถทำงานที่ซับซ้อนได้ หรือหากหน่วยย่อยบางหน่วยเสียหายไปบ้าง โครงข่ายนี้ก็ยังสามารถทำงานต่อได้เพราะยังมีหน่วยย่อยอื่น ๆ ช่วยกันประมวลผลอยู่อีกมากมาย และหากได้เรียนรู้จากข้อมูลที่มากขึ้น โครงข่ายนี้ก็จะสามารถทำงานที่มีความฉลาดคล้ายกับสมองมนุษย์ หรืออาจจะเหนือกว่าสมองของมนุษย์ก็เป็นได้

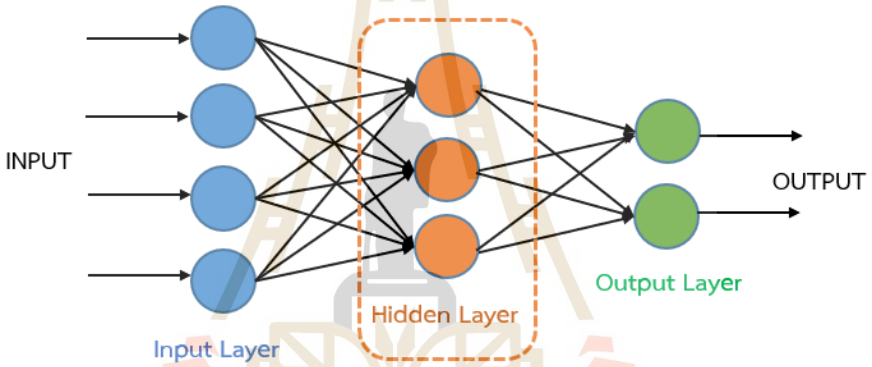
มีงานวิจัยมากมายที่เกี่ยวข้องกับการนำโครงข่ายประสาทเทียมมาประยุกต์ใช้ ได้แก่ งานควบคุมเครื่องจักรในอุตสาหกรรมยานยนต์และอุตสาหกรรมการบิน งานทางด้านดาราศาสตร์ งานทางด้านการพัฒนาเกมส์และการแข่งขัน งานทางด้านธนาคาร และการตรวจจับการทุจริต เป็นต้น โดยยังพบอีกว่าสามารถนำมาใช้แก้ปัญหาในการคำนวณได้อีกด้วย เช่น การจำแนกประเภทข้อมูล การรู้จำรูปแบบ การคัดเลือกคุณลักษณะการหาค่าเฉลี่ยแบบประมาณการณ์ การแบ่งกลุ่มข้อมูล เป็นต้น

โครงข่ายประสาทเทียมประกอบไปด้วยโหนดและเส้นเชื่อมระหว่างโหนด ซึ่งแบ่งระดับการทำงานเป็น 3 ระดับ ได้แก่ ชั้นอินพุต (Input Layer) ชั้นซ่อน (Hidden layer) ซึ่งอาจมีได้มากกว่าหนึ่งชั้นซ่อน และชั้นเอาต์พุต (Output layer) โดยมีรายละเอียดดังต่อไปนี้

1. ชั้นอินพุต ประกอบไปด้วยโหนดที่เรียกว่า Input node และเส้นเชื่อมที่จะเชื่อมต่อไปยังชั้นซ่อน โดยที่จำนวน Input node จะเท่ากับจำนวนคุณลักษณะ (Attribute) ของข้อมูล

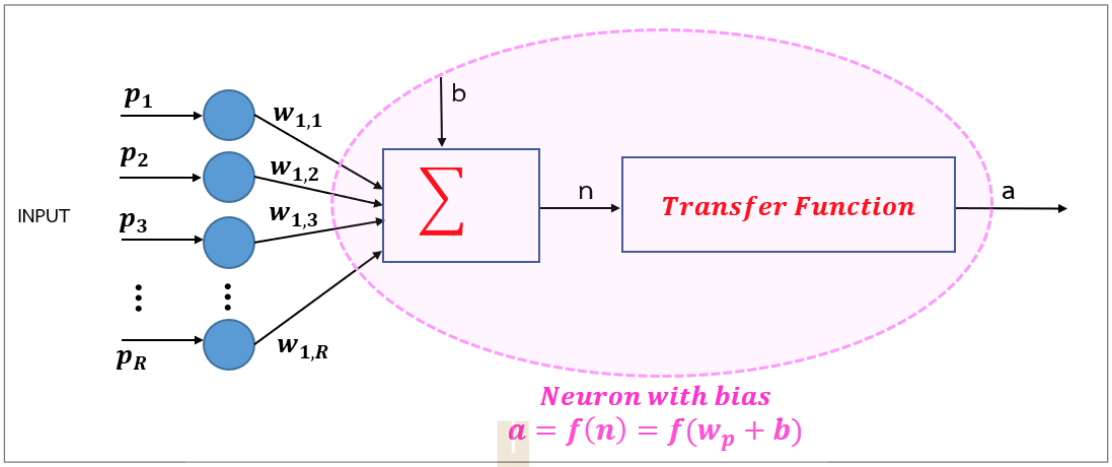
2. ชั้นซ่อน ประกอบไปด้วยโหนดที่เรียกว่า Hidden node และเส้นเชื่อมที่จะเชื่อมต่อไปยังชั้นซ่อนชั้นต่อไป (กรณีที่ผู้ออกแบบกำหนดชั้นซ่อนไว้หลายชั้น) หรือเชื่อมต่อไปยังชั้นเอาต์พุต (ในกรณีที่มีชั้นซ่อนเพียงชั้นเดียว) โดยที่ Hidden node จะรับข้อมูลต่อมาจากชั้นอินพุต หรือชั้นซ่อนในชั้นก่อนหน้า

3. ชั้นเอาต์พุต ประกอบไปด้วยโหนดที่เรียกว่า Output node จำนวนโหนดจะเท่ากับจำนวนกลุ่มหรือจำนวนประเภทของข้อมูลที่ต้องการจำแนก โดยที่ Output node จะรับข้อมูลต่อมาจากชั้นซ่อนชั้นสุดท้าย



รูปที่ 2.41 ตัวอย่างโครงข่ายประสาทเทียมอย่างง่าย

จากรูปที่ 2.41 แสดงให้เห็นถึงตัวอย่างโครงข่ายประสาทเทียมอย่างง่าย ซึ่งจะเห็นว่าการทำงานภายในมีการเชื่อมต่อโครงข่ายประสาทเทียมทั้งหมด 3 ระดับ คือ Input layer, Hidden layer และ Output layer ซึ่งการเชื่อมต่อกันในแต่ละระดับจะใช้วิธีการคำนวณทางคณิตศาสตร์ โดยในโครงข่ายประสาทเทียมนี้จะมีเส้นเชื่อมจากทุกโหนดในชั้นอินพุตไปยังทุกโหนดในชั้นซ่อน และมีเส้นเชื่อมจากทุกโหนดในชั้นซ่อนไปยังทุกโหนดในชั้นเอาต์พุต ซึ่งในแต่ละเส้นเชื่อมจะมีค่าน้ำหนัก (Weight) กำหนดอยู่ ส่วนค่าน้ำหนักในรอบแรกนั้นมาจากการสุ่ม



รูปที่ 2.42 ตัวอย่างโครงข่ายประสาทเทียมหนึ่งหน่วย แบบหลายอินพุต

จากรูปที่ 2.42 แสดงตัวอย่างโครงข่ายประสาทเทียมหนึ่งหน่วย แบบหลายอินพุต โดยลักษณะการทำงานของแต่ละโหนดนั้น เทียบได้กับเซลล์ประสาทในสมองมนุษย์ 1 เซลล์ ซึ่งประกอบไปด้วยอินพุตที่จะเข้าสู่โหนดโดยอยู่ในรูปแบบเวกเตอร์ของข้อมูลคุณลักษณะ ตัวอย่างเช่น มีค่า $p = [p_1, p_2, p_3, \ldots, p_R]$ ซึ่งเป็นค่าอินพุตที่ถูกป้อนเข้าไปและมีจำนวน R องค์ประกอบ และเวกเตอร์น้ำหนัก ค่า $w = [w_1, w_2, w_3, \ldots, w_R]$ จากนั้นนำอินพุตมาคูณกับน้ำหนักในแต่ละเส้นเชื่อม ผลที่ได้จากอินพุตทุก ๆ เส้นเชื่อมของโหนดจะนำมารวมกัน และรวมกับค่า b ซึ่งเป็นค่าเอนเอียงหรือค่าไบแอส จากนั้นจะส่งผลลัพธ์ส่วนนี้ไปยังฟังก์ชันถ่ายโอน (Transfer Function) เมื่อผ่านฟังก์ชันถ่ายโอนแล้วจะได้ค่าเอาต์พุต a หรือกล่าวได้ว่าผลรวมค่าอินพุตและน้ำหนักบวกค่าไบแอส คือ n จากนั้น f ซึ่งเป็นฟังก์ชันถ่ายโอนจะทำการรับอินพุต n เพื่อเปลี่ยนเป็นค่าเอาต์พุต a ซึ่งแสดงในสมการที่ 2.17

$$a = f(n) = f(Wp + b) \tag{2.17}$$

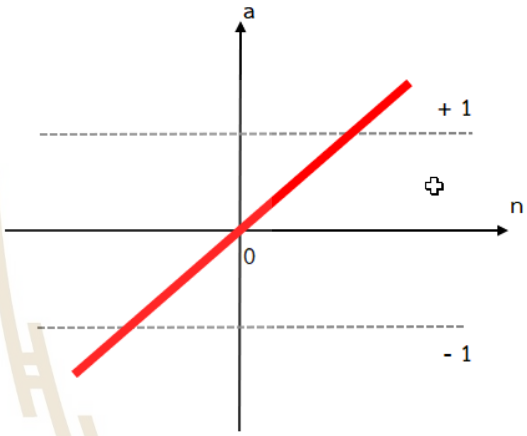
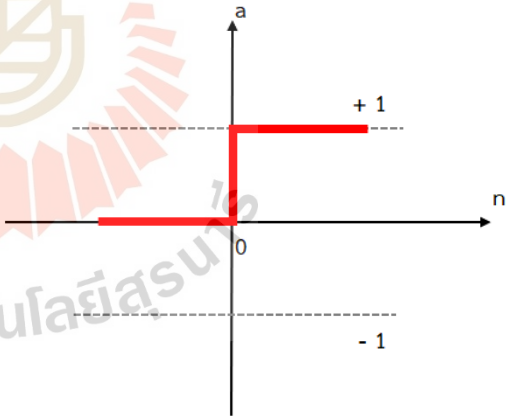
เมื่อ $n = w_{1,1}p_1 + w_{1,2}p_2 + w_{1,3}p_3 + \cdots + w_{1,R}p_R + b = Wp + b$

2.12.1 ฟังก์ชันถ่ายโอน

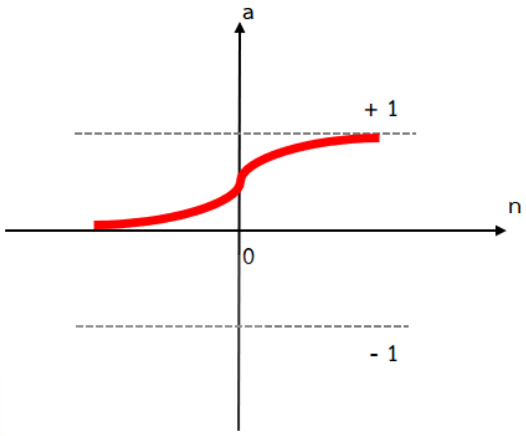
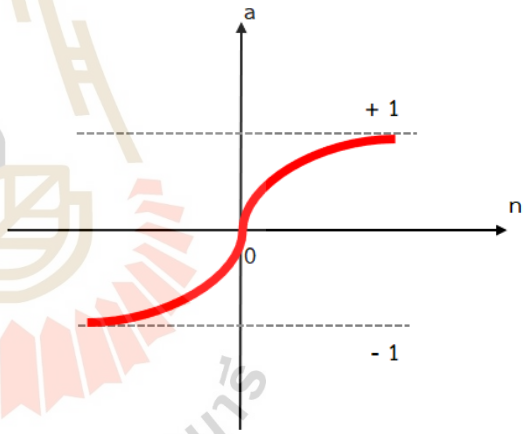
ฟังก์ชันถ่ายโอน หรือฟังก์ชันกระตุ้น (Activation Function) จะเป็นตัวกำหนดค่าของผลลัพธ์ของเซลล์ประสาทเทียม (รติพร จันทร์กลั่น, 2560; Gardner & Dorling, 1998; Zhu, 2017) ซึ่งสามารถเป็นฟังก์ชันใดก็ได้ในทางคณิตศาสตร์ และสามารถเลือกฟังก์ชันได้ตามประเภท

ของโจทย์ปัญหา แล้วแต่ลักษณะผลลัพธ์ว่าเป็นค่าต่อเนื่องหรือไม่ ขอบเขตของผลลัพธ์เป็นอย่างไร ซึ่งฟังก์ชันถ่ายโอนมีอยู่หลายแบบด้วยกัน ดังแสดงในตารางที่ 2.13

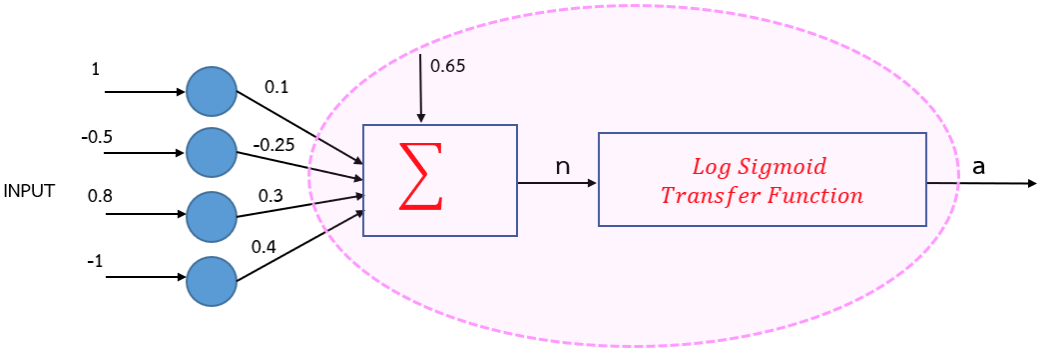
ตารางที่ 2.13 อธิบายฟังก์ชันถ่ายโอน และลักษณะกราฟของฟังก์ชันถ่ายโอนแบบต่าง ๆ

ที่	ฟังก์ชัน	กราฟ
1	ฟังก์ชันถ่ายโอนแบบเชิงเส้น (Linear Transfer Function) ค่าของผลลัพธ์ คือ a = n มีช่วงของค่าอยู่ที่ [-∞,∞] เขียนให้อยู่ในรูปสมการได้ดังนี้ $a = \begin{cases} n, \text{ถ้า } n > 0 \\ 0, \text{ถ้า } n = 0 \\ -n, \text{ถ้า } n < 0 \end{cases}$	
2	ฟังก์ชันถ่ายโอนแบบฮาร์ดลิมิต (Hard Limit Transfer Function) ค่าของผลลัพธ์ ถ้า n มีค่าน้อยกว่า 0 ค่า a จะเท่ากับ 0 แต่หาก n มีค่าเท่ากับ 0 หรือมากกว่า ค่าผลลัพธ์จะเป็น 1 เขียนให้อยู่ในรูปสมการได้ดังนี้ $a = \begin{cases} 0, \text{ถ้า } n < 0 \\ 1, \text{ถ้า } n \geq 0 \end{cases}$	

ตารางที่ 2.13 อธิบายฟังก์ชันถ่ายโอน และลักษณะกราฟของฟังก์ชันถ่ายโอนแบบต่าง ๆ (ต่อ)

ที่	ฟังก์ชัน	กราฟ
3	ฟังก์ชันถ่ายโอนแบบล็อก-ซิกมอยด์ (Log Sigmoid Transfer Function) ค่าของผลลัพธ์อยู่ในช่วง [0,1] โดยมีสมการดังนี้ $a = \frac{1}{1+e^{-n}}$	
4	ฟังก์ชันถ่ายโอนแบบแทนเจนท์-ซิกมอยด์ (Tan Sigmoid Transfer Function) ค่าของผลลัพธ์อยู่ในช่วง [-1,1] โดยมีสมการดังนี้ $a = \frac{2}{1+e^{-2n}} - 1$	

จากตารางที่ 2.13 ทำให้ทราบรูปแบบต่าง ๆ ของฟังก์ชันถ่ายโอน ต่อไปจะมาทดสอบการคำนวณภายในโหนดของโครงข่ายประสาทเทียม ในรูปที่ 2.43 แสดงการคำนวณการทำงานภายในโหนดจำนวน 1 โหนด ซึ่งมีข้อมูลอินพุต 1 แถว และมี 4 ค่าแอททริบิวต์ ได้แก่ p1 = 1, p2 = -0.5 p3 = 0.8 และ p4 = -1 โดยมีเวกเตอร์น้ำหนักแต่ละเส้นเชื่อม คือ w1,1 = 0.1, w1,2 = -0.25 w1,3 = 0.3 และ w1,4 = 0.4 และมีค่าไบแอส คือ b = 0.65 โดยกำหนดให้โหนดนี้ใช้ฟังก์ชันถ่ายโอนแบบล็อก-ซิกมอยด์



รูปที่ 2.43 ค่าข้อมูลอินพุต ค่าไบเอส และเวกเตอร์น้ำหนักในแต่ละเส้นเชื่อมโหนดที่ 1

จากตัวอย่างในรูปที่ 2.43 สามารถแสดงการคำนวณได้ดังนี้

หาค่า n จากสูตร $n = w_{1,1}p_1 + w_{1,2}p_2 + w_{1,3}p_3 + \cdots + w_{1,R}p_R + b$

แทนค่า n = (0.1 * 1) + (-0.25 * -0.5) + (0.3 * 0.8) + (0.4 * -1) + 0.65

= 0.1 + 0.125 + 0.24 + (-0.4) + 0.65

= 0.715

หาค่า a จากสูตร $a = \frac{1}{1+e^{-n}}$

แทนค่า a $= \frac{1}{1+e^{-0.715}}$

$= \frac{1}{1+0.4891}$

$= \frac{1}{1.4891}$

= 0.6715

ดังนั้น ค่าผลลัพธ์ของโหนดนี้ คือ 0.6715

2.12.2 การปรับพารามิเตอร์เพื่อให้โครงข่ายประสาทเทียมจดจำสิ่งที่เรียนรู้

โครงข่ายประสาทเทียมประกอบไปด้วยพารามิเตอร์หลายตัว พารามิเตอร์ที่สามารถปรับหรือเปลี่ยนค่าเพื่อให้เหมาะสมกับสิ่งที่กำลังเรียนรู้อยู่นั้นก็คือค่าน้ำหนัก (Weight) และค่าไบแอส (Bias) ซึ่งค่าเหล่านี้แรกเริ่มจะได้มาจากการสุ่ม (รติพร จันทร์กลั่น, 2560; Russell and Norvig, 2016; Salimans and Kingma, 2016) จากนั้นโครงข่ายจะพยายามจดจำและเรียนรู้จากข้อมูลที่ใช้ฝึกสอนและพยายามปรับค่าน้ำหนักในแต่ละเส้นเชื่อมเพื่อให้สามารถจำแนกประเภทของข้อมูลเหล่านั้นได้อย่างถูกต้องที่สุด หากผลลัพธ์จากการฝึกในแต่ละรอบมีค่าเอาต์พุตที่ผิดไปจากค่าจริง โครงข่ายก็จะพยายามปรับน้ำหนักในรอบใหม่ให้เหมาะสมและเกิดข้อผิดพลาดให้น้อยลง หรือตามแต่เกณฑ์ที่สามารถยอมรับได้ ซึ่งค่าความผิดพลาดสามารถคำนวณได้จากสมการที่ 2.18

$$e = t - y \tag{2.18}$$

เมื่อ e คือ ค่าความผิดพลาด
t คือ ค่าจริงของเป้าหมาย
y คือ ค่าเอาต์พุตจากโครงข่ายประสาทเทียม

ในส่วนของค่าน้ำหนักและไบแอสที่ถูกปรับใหม่สามารถคำนวณโดยใช้สมการที่ 2.19 และ 2.20 ซึ่งแสดงดังสมการต่อไปนี้

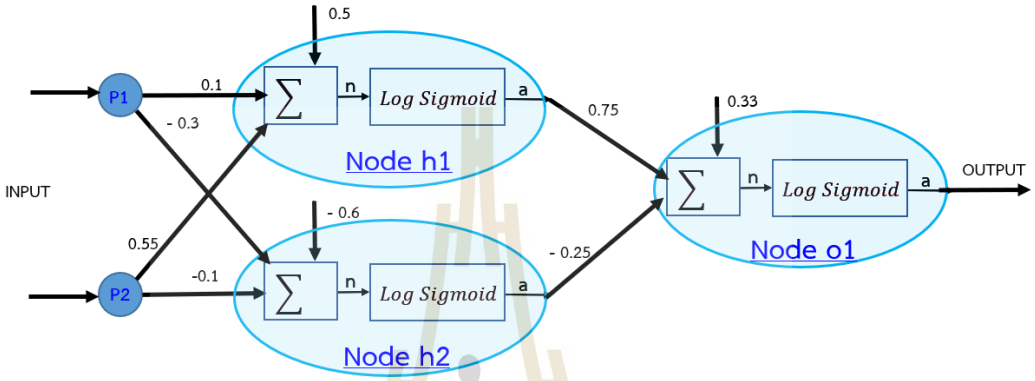
$$w^{new} = w^{old} + ep \tag{2.19}$$

$$b^{new} = b^{old} + e \tag{2.20}$$

เมื่อ w^{new} คือ ค่าน้ำหนักปรับใหม่
w^{old} คือ ค่าน้ำหนักเก่า
b^{new} คือ ค่าไบแอสปรับใหม่
b^{old} คือ ค่าไบแอสเก่า
p คือ ข้อมูล
e คือ ค่าความผิดพลาด

ตารางที่ 2.14 ตัวอย่างข้อมูลเริ่มต้นที่ต้องการให้โครงข่ายประสาทเทียมเรียนรู้

ข้อมูล	Input1 (P_1)	Input2 (P_2)	Output1
1	1.0	0.8	0.6
2	-0.6	-0.4	-0.2



รูปที่ 2.44 ตัวอย่างโครงข่ายประสาทเทียมแบบ 2-2-1 โดยแสดงค่าน้ำหนัก และค่าไบแอสเริ่มต้นของโครงข่าย

จากสมการข้างต้น รวมไปถึงตัวอย่างข้อมูลในตารางที่ 2.14 และโครงข่ายประสาทเทียมจำลองจากรูปที่ 2.44 สามารถแสดงวิธีการคำนวณการปรับค่าพารามิเตอร์น้ำหนักแต่ละเส้นเชื่อมและค่าไบแอสของแต่ละโหนดเพื่อเรียนรู้และจดจำของโครงข่ายประสาทเทียมได้ดังนี้

ข้อมูลอินพุตที่ 1 กำหนดให้ p1 = 1.0, p2 = 0.8

Node h1

n = (0.1 * 1.0) + (-0.3 * 0.8) + 0.5 = 1.04

a $= \frac{1}{1+e^{-1.04}}$ = 0.74

Node h2

n = (-0.3 * 1.0) + (-0.1 * 0.8) - 0.6 = -0.98

a = $\frac{1}{1+e^{-(-0.98)}}$ = 0.27

Node o1

n = (0.75 * 0.74) + (-0.25 * 0.27) + 0.33 = 0.82

a = $\frac{1}{1+e^{-(0.82)}}$ = 0.69

คำนวณค่าความผิดพลาด $e = t - y$

จะได้ e = 0.6 - 0.69

= -0.09

ปรับค่าน้ำหนักจากสมการ $w^{new} = w^{old} + ep$

w1,1 = 0.1 + (-0.09 * 1.0) = 0.010

w1,2 = 0.55 + (-0.09 * 0.8) = 0.478

w2,1 = -0.3 + (-0.09 * 1.0) = -0.390

w2,2 = -0.1 + (-0.09 * 0.8) = -0.172

w0,1 = 0.75 + (-0.09 * 0.74) = 0.683

w0,2 = -0.25+ (-0.09 * 0.27) = -0.274

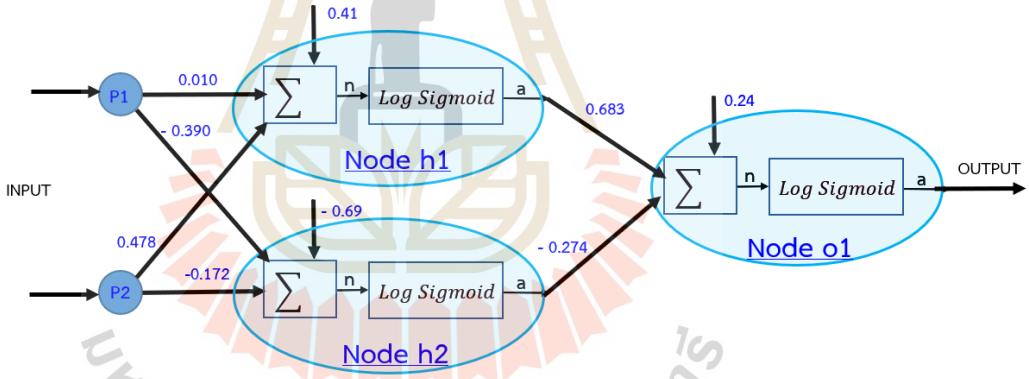
ปรับค่าไบเอสจากสมการ $b^{new} = b^{old} + e$

b1 = 0.5 + (-0.09) = 0.41

b2 = -0.6 + (-0.09) = -0.69

b0 = 0.33 + (-0.09) = 0.24

แสดงค่าน้ำหนักและค่าไบแอสที่ถูกปรับจากการเรียนรู้ข้อมูลอินพุตแถวที่ 1 ของโครงข่ายประสาทเทียมแบบ 2-2-1 แสดงในรูป 2.45



รูปที่ 2.45 แสดงค่าน้ำหนักและค่าไบแอสที่ปรับค่ารอบที่ 1 ของโครงข่ายประสาทเทียมแบบ 2-2-1

ข้อมูลอินพุตที่ 2 กำหนดให้ p1 = -0.6, p2 = -0.4

Node h1

n = (0.010 * -0.6) + (0.478 * -0.4) + 0.41

= 0.21

a $= \frac{1}{1+e^{-0.21}}$

$= 0.55$

Node h2

n = (-0.390 * -0.6) + (-0.172 * -0.4) - 0.69

= -0.39

a $= \frac{1}{1+e^{-(-0.39)}}$

= 0.40

Node o1

n = (0.683 * 0.55) + (-0.274 * 0.40) + 0.24

= 0.51

a $= \frac{1}{1+e^{-(0.51)}}$

= 0.62

คำนวณค่าความผิดพลาด $e = t - y$

จะได้ e = (-0.2) - 0.62

= -0.82

ปรับค่าน้ำหนักจากสมการ $w^{new} = w^{old} + ep$

w1,1 = 0.010 + (-0.82 * -0.6) = 0.502

w1,2 = 0.478 + (-0.82 * -0.4) = 0.806

w2,1 = -0.390 + (-0.82 * -0.6) = 0.102

w2,2 = -0.172 + (-0.82 * -0.4) = 0.156

w0,1 = 0.683 + (-0.82 * -0.6) = 1.175

w0,2 = -0.274+ (-0.82 * -0.4) = 0.054

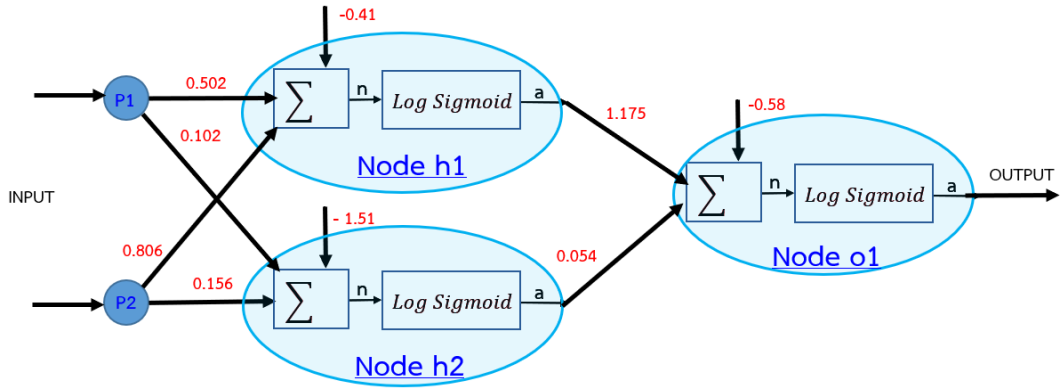
ปรับค่าไบเอสจากสมการ $b^{new} = b^{old} + e$

b1 = 0.41 + (-0.82) = -0.41

b2 = -0.69 + (-0.82) = -1.51

b0 = 0.24 + (-0.82) = -0.58

แสดงค่าน้ำหนักและค่าไบแอสที่ถูกปรับจากการเรียนรู้ข้อมูลอินพุตแถวที่ 2 ของโครงข่ายประสาทเทียมแบบ 2-2-1 แสดงในรูป 2.46



รูปที่ 2.46 แสดงค่าน้ำหนักและค่าไบแอสที่ปรับค่ารอบที่ 2 ของโครงข่ายประสาทเทียมแบบ 2-2-1

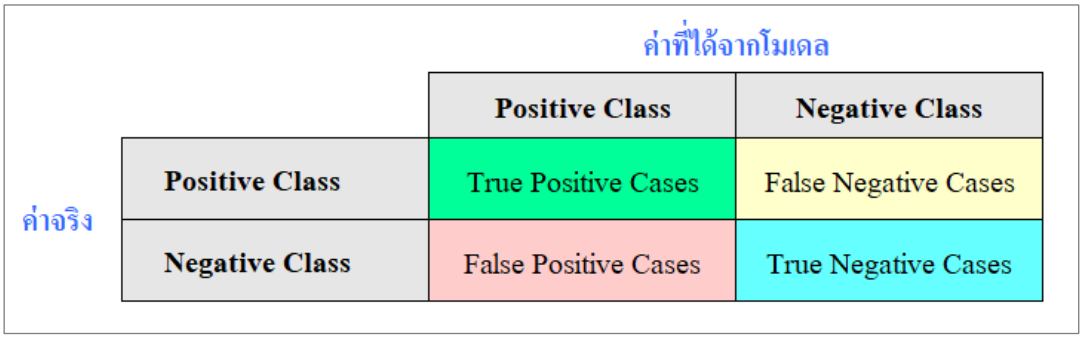
จะเห็นได้ว่าในแต่ละรอบที่มีข้อมูลนำเข้าโครงข่ายประสาทเทียม โครงข่ายประสาทเทียมจะพยายามปรับค่าพารามิเตอร์ (ค่าน้ำหนักของเส้นเชื่อมและค่าไบแอส) เพื่อลดข้อผิดพลาดของการทำนายให้ได้มากที่สุด ซึ่งสิ่งนี้เรียกว่าเป็นกระบวนการเรียนรู้จากข้อมูลนำเข้า ส่วนในกรณีที่มีข้อมูลนำเข้ามากกว่า 2 ข้อมูล ขั้นตอนนี้ก็จะต้องวนซ้ำไปจนกว่าจะครบทุกข้อมูล ทำให้สุดท้ายแล้วจะได้ค่าพารามิเตอร์ที่เหมาะสมกับชุดข้อมูลนี้มากที่สุด หรือหากครบเงื่อนไขที่ตั้งไว้ก่อนก็จะหยุดทำและส่งผลลัพธ์ที่ดีที่สุด ณ ขณะนั้นมาให้

2.13 มาตรวัดประสิทธิภาพสำหรับการจำแนกประเภท

มาตรวัดประสิทธิภาพที่ใช้สำหรับงานทางด้านการจำแนกประเภทข้อมูล (เพื่อใช้พิจารณาประกอบในการคัดเลือกคุณลักษณะ) ได้แก่ การประเมินด้วยค่าความแม่นยำในการจำแนกข้อมูล (Accuracy) และค่าพื้นที่ใต้กราฟ ROC (Receiver Operating Characteristic) แต่ก่อนอื่นต้องทำความรู้จักเมทริกซ์วัดประสิทธิภาพ (Confusion Matrix) โดยมีรายละเอียดดังต่อไปนี้

2.13.1 เมทริกซ์วัดประสิทธิภาพ (Confusion Matrix)

เมทริกซ์วัดประสิทธิภาพ คือ เมทริกซ์ที่ใช้สำหรับแสดงผลลัพธ์ที่ได้จากการนำโมเดลไปทำการจำแนกข้อมูลจริงหรือข้อมูลชุดทดสอบ (Story and Congalton, 1986; Visa et al., 2011) โดยประเมินจากข้อมูลที่ได้มาจากการทำนายของโมเดล (Predicted Labels) กับผลลัพธ์จริง ๆ (Actual Labels) จากข้อมูลที่ทราบผลเฉลยอยู่แล้ว ดังรูปที่ 2.47 ซึ่งเป็นเมทริกซ์แสดงผลลัพธ์การจำแนกของข้อมูล 2 คลาส คือ คลาสบวก (Positive Class) และคลาสลบ (Negative Class)



รูปที่ 2.47 ตัวอย่างเมทริกซ์วัดประสิทธิภาพสำหรับการจำแนกในกรณีที่ข้อมูลมี 2 คลาส

จากรูปที่ 2.47 ค่าที่ได้จากเมทริกซ์วัดประสิทธิภาพสำหรับการจำแนกในกรณีที่ข้อมูลมี 2 คลาส โดยสามารถแบ่งออกเป็น 4 กรณี ได้แก่

True Positive Cases คือ ข้อมูลจากผลเฉลยเป็นคลาสบวก และโมเดลทำนายว่าเป็นคลาสบวกด้วย ซึ่งถือว่าโมเดลทำนายถูกต้องตรงกับข้อมูลจริง

True Negative Cases คือ ข้อมูลจากผลเฉลยเป็นคลาสลบ และโมเดลทำนายว่าเป็นคลาสลบด้วย ซึ่งถือว่าโมเดลทำนายถูกต้องตรงกับข้อมูลจริง

False Positive Cases คือ ข้อมูลจากผลเฉลยเป็นคลาสลบ แต่โมเดลทำนายว่าเป็นคลาสบวก ซึ่งถือว่าโมเดลทำนายผิดไปจากข้อมูลจริง

False Negative Cases คือ ข้อมูลจากผลเฉลยเป็นคลาสบวก แต่โมเดลทำนายว่าเป็นคลาสลบ ซึ่งถือว่าโมเดลทำนายผิดไปจากข้อมูลจริง

2.13.2 ค่าความแม่นยำในการจำแนกข้อมูล (Accuracy)

ค่าความแม่นยำในการจำแนกข้อมูล คือ ค่าความแม่นที่บ่งบอกว่าโมเดลนั้น ๆ ใช้ทำนายข้อมูลได้ถูกต้องอยู่ในระดับใด โดยเป็นการประเมินประสิทธิภาพการจำแนกโดยรวมทุกคลาสของโมเดล (Baldi et al., 2000) ดังสมการที่ 2.21

$$Accuracy = \frac{True\ Positive\ Cases + True\ Negative\ Cases}{All\ Data} \tag{2.21}$$

2.13.3 ค่าพื้นที่ใต้กราฟ ROC (Receiver Operating Characteristic)

ค่าพื้นที่ใต้กราฟ หรือมักถูกเรียกสั้น ๆ กันว่าค่า AUC (Area Under ROC Curve) คือ ค่าที่ใช้สำหรับการวิเคราะห์ความไว (Sensitivity) ของโมเดลที่จะจำแนกข้อมูลได้ถูกต้อง ซึ่งเป็นการแสดงความสัมพันธ์ระหว่างข้อมูลที่ทำนายถูก หรือ TP Rate (แสดงในแกน Y) และข้อมูลที่ทำนายผิด FP Rate (แสดงในแกน X) เพื่อใช้แสดงประสิทธิภาพของโมเดลสำหรับการจำแนกข้อมูลที่มีสองคลาส หากพื้นที่ใต้กราฟมีค่าเข้าใกล้ 1 แสดงว่าประสิทธิภาพของโมเดลนั้นมีประสิทธิภาพที่ดีมาก (Davis and Goadrich, 2006; Fawcett, 2006; Powers, 2020) ซึ่ง TPR และ FPR สามารถคำนวณได้จากสมการที่ 2.22 และ 2.23

$$True\ Positive\ Rate\ (TPR) = \frac{True\ Positive\ Cases}{True\ Positive\ Cases + False\ Negative\ Cases} \tag{2.22}$$

$$False\ Positive\ Rate\ (FPR) = \frac{False\ Positive\ Cases}{False\ Positive\ Cases + True\ Negative\ Cases} \tag{2.23}$$

2.14 มาตรวัดประสิทธิภาพในการคาดการณ์ผลลัพธ์

มาตรวัดประสิทธิภาพในการคาดการณ์ผลลัพธ์ หรือ Prediction ซึ่งมีตัวแปร Y เป็นค่าตัวเลข ทำให้ต้องใช้มาตรวัดอีกประเภทหนึ่งซึ่งแตกต่างจากการจำแนกประเภทข้อมูลดังอธิบายข้างต้น โดยมาตรวัดประสิทธิภาพที่นิยมใช้ในขั้นตอนนี้ ได้แก่ ค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ย (Mean Absolute Error :MAE) และค่ารากที่สองของความคลาดเคลื่อนกำลังสองเฉลี่ย (Root Mean Squared Error :RMSE) (Willmott and Matsuura, 2005; Chai and Draxler, 2014)

2.14.1 ค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ย (Mean Absolute Error: MAE)

ค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ย คือ การหาเฉลี่ยของความผิดพลาดในข้อมูลทุกตัวด้วยการหาผลต่างสัมบูรณ์ระหว่างค่าที่แท้จริงกับค่าที่โมเดลคาดการณ์ โดยยิ่งมีค่าน้อยแสดงว่าโมเดลที่ได้ยิ่งมีความแม่นยำสูง ซึ่งคำนวณได้จากสมการที่ 2.24

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y_i}| \tag{2.24}$$

โดย MAE คือ ค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ย

n คือ จำนวนข้อมูล

y_i คือ ค่าที่แท้จริง

$\hat{y}_i$ คือ ค่าที่ได้จากโมเดลคาดการณ์

2.14.2 ค่ารากที่สองของความคลาดเคลื่อนกำลังสองเฉลี่ย (Root Mean Squared Error: RMSE)

ค่ารากที่สองของความคลาดเคลื่อนกำลังสองเฉลี่ย คือ การวัดค่าความคลาดเคลื่อนที่นิยมใช้กันอย่างแพร่หลาย ใช้ระบุความแตกต่างหว่างค่าที่แท้จริงและค่าที่โมเดลคาดการณ์ โดยการหาค่าเฉลี่ยผลต่างกำลังสองของข้อมูลระหว่างค่าที่แท้จริงกับค่าที่โมเดลคาดการณ์ จากนั้นนำมาหารากที่สอง หากค่า RMSE มีค่าน้อยแสดงว่าโมเดลมีความแม่นยำมาก ซึ่งคำนวณได้จากสมการที่ 2.25

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \quad (2.25)$$

โดย RMSE คือ ค่ารากที่สองของความคลาดเคลื่อนกำลังสองเฉลี่ย

n คือ จำนวนข้อมูล

y_i คือ ค่าที่แท้จริง

$\hat{y}_i$ คือ ค่าที่ได้จากโมเดลคาดการณ์

2.15 งานวิจัยที่เกี่ยวข้อง

ในส่วนนี้เป็นรายละเอียดที่จะกล่าวถึงงานวิจัยที่ผู้วิจัยได้ศึกษาค้นคว้าและมีความเกี่ยวโยงกับการทำงานวิจัยชิ้นนี้ เพราะเนื่องจากงานวิจัยชิ้นนี้มีขอบเขตและเนื้อหาที่กว้าง ผู้วิจัยจึงได้จัดกลุ่มงานวิจัยที่เกี่ยวข้องออกเป็น 3 หมวดหมู่ดังนี้ 1. งานวิจัยที่เกี่ยวข้องกับการจัดการปัญหาข้อมูลที่ไม่สมดุล 2. งานวิจัยที่เกี่ยวข้องกับการคัดเลือกคุณลักษณะ 3. งานวิจัยที่เกี่ยวข้องกับการคาดการณ์ผลผลิต

2.15.1 งานวิจัยที่เกี่ยวข้องกับการจัดการปัญหาข้อมูลที่ไม่สมดุล

ปัญหาเรื่องความไม่สมดุลของข้อมูลนั้นพบได้ทั่วไปโดยเฉพาะอย่างยิ่งในกระบวนการผลิตของอุตสาหกรรมต่าง ๆ ดังนั้นจึงมีความจำเป็นที่จะต้องจัดการกับปัญหาความไม่สมดุลนั้นเสียก่อน โดยในหลายงานวิจัยพบว่าการปรับข้อมูลระหว่างคลาสข้อมูลส่วนมากและข้อมูลส่วนน้อยให้เป็นอัตราส่วน 1:1 จะให้ผลลัพธ์ที่ดีที่สุด

Haoyue Liu และ MengChu Zhou (Liu and Zhou, 2017) ได้นำเสนองานวิจัยที่ทดลองใช้อัลกอริทึม Classification and Regression Tree (CART) ในการช่วยคัดเลือกคุณลักษณะ โดยใช้เกณฑ์ในการสร้างโมเดลเป็น Gini Index จากนั้นนำไปเปรียบเทียบประสิทธิภาพการจำแนกด้วยการคัดเลือกคุณลักษณะแบบฟิลเตอร์ (Filter Method) คือ Chi-Square และ F-Statistic ซึ่งผลลัพธ์ที่ได้นั้นพบว่าวิธีการคัดเลือกคุณลักษณะด้วย CART ให้ค่า AUC ที่ดีกว่าอีกสองวิธีการที่นำมาเปรียบเทียบ

Ruangthong และ Jaiyen (Ruangthong and Jaiyen, 2016) ได้ทำการทดลองรวมเอาเทคนิคการจำแนกข้อมูลด้วยอัลกอริทึม Decision Tree และ Bayesian Network เข้าด้วยกัน เพื่อจัดการกับปัญหาชุดข้อมูลไม่สมดุล เทคนิคใหม่ที่ได้จากการทดลองนี้ถูกเรียกว่า Bayesian Network, Alternating Decision Tree, Tree-J48 และ REPTree (BNRAC) โดยเทคนิคนี้ได้นำไปทดลองกับชุดข้อมูลจริงในด้านการสำรวจความพึงพอใจของลูกค้าที่ได้รับการบริการจากธนาคารแห่งหนึ่งในประเทศโปรตุเกส โดยพบว่าเทคนิคใหม่ที่ได้นี้ให้ประสิทธิภาพในการจำแนกที่ดีกว่าเทคนิคและอัลกอริทึมที่นำมาเปรียบเทียบ ได้แก่ Random Forest, Rotation Forest, Bayesian Network, SMOTE และ RusBoost

2.15.2 งานวิจัยที่เกี่ยวข้องกับการคัดเลือกคุณลักษณะ

เทคนิคและอัลกอริทึมที่ถูกนำมาใช้ในการคัดเลือกคุณลักษณะนั้นมีอยู่หลากหลายวิธีและมีจุดเด่นที่แตกต่างกันออกไป การคัดเลือกคุณลักษณะมีความสำคัญอย่างมากในการสร้างตัวแบบโมเดลเพราะนอกเหนือจากการลดเวลาในการประมวลผลของคอมพิวเตอร์ลงแล้ว ยังส่งผลกระทบโดยตรงต่อประสิทธิภาพในการจำแนกและการคาดการณ์ของโมเดลอีกด้วย

Hamza Turabieh และคณะ (Turabieh et al., 2019) ได้ตีพิมพ์งานวิจัยเกี่ยวกับการพัฒนาอัลกอริทึมเพื่อใช้ในการคัดเลือกคุณลักษณะสำหรับการนำไปทำนายการเกิดข้อผิดพลาดของซอฟต์แวร์ (Software Fault Prediction) ซึ่งอัลกอริทึมที่พัฒนาขึ้นมานี้มีพื้นฐานมาจากอัลกอริทึม ANN ในส่วนของอัลกอริทึมที่นำมาใช้ในการเปรียบเทียบนั้นมีหลายชนิดด้วยกัน ได้แก่ Naïve Bayes (NB), ANN, k-NNs, Decision Tree และ Linear Regression โดยผลลัพธ์ที่ได้จากการทดลองนั้นมีประสิทธิภาพสูงกว่าอัลกอริทึมที่นำมาเปรียบเทียบอย่างเห็นได้ชัดเจน

2.15.3 งานวิจัยที่เกี่ยวข้องกับการคาดการณ์ผลผลิต

ในปี 2011 Tao Yuan (Yuan et al., 2011) ได้ทดลองทำการวิจัยเกี่ยวกับ Yield หรือ Defect Forecasting ในอุตสาหกรรมการผลิตเวเฟอร์ โดยได้ทดลองทำการปรับปรุงโมเดลการทำนายที่มีพื้นฐานจาก อัลกอริทึม Simple Linear Regression ไปใช้ Zero-Inflated Poisson (ZIP) Regression และ Zero-Inflated Negative Binomial (ZINB) Regression ซึ่งเหมาะกับข้อมูลที่มีค่า

ส่วนใหญ่เป็นศูนย์ โดยผลการทดลองนั้นพบว่าทั้ง Regression แบบ ZIP และZINB นั้นให้ผลลัพธ์ที่ดีกว่า Simple Linear Regression

Hoyeop Lee (Lee et al., 2015) ได้นำเสนอวิธีการใหม่ในสำหรับการคาดการณ์ผลผลิตในวงการอุตสาหกรรมการผลิตแผ่นบอร์ดวงจร PCB (Printed Circuit Board) และเรียกวิธีการนี้ว่า Predictive Association Rule Considering the Event Sequence (PARCOS) โดยนำไปเปรียบเทียบกับ 3 โมเดลดั้งเดิมที่มีพื้นฐานจาก Regression ได้แก่ LASSO Regression, PLS Regression และ CART ผลปรากฏว่าวิธีการ PARCOS นั้นให้ผลลัพธ์ด้านการทำนายที่ดีกว่า โดยงานวิจัยชิ้นนี้ใช้ตัววัดประสิทธิภาพ คือ MAPE (Mean Absolute Percentage Error)

Toly Chen (Chen, 2017) ได้นำเสนอและมีส่วนร่วมในงานวิจัยมากมายที่เกี่ยวข้องกับการทำ Yield prediction ในวงการอุตสาหกรรมการผลิตของผลิตภัณฑ์เกี่ยวกับ Semi-Conductor โดยในปี ค.ศ. 2016 ได้ทำการทดลองสร้างโมเดลสำหรับกับการทำ Yield Prediction โดยใช้พื้นฐานจากอัลกอริทึม Artificial Neural Network (ANN) จากชุดข้อมูลฝึกสอนหลาย ๆ แหล่ง ซึ่งผลลัพธ์ที่ได้นั้นพบว่ามีความสามารถในการทำนายได้อย่างแม่นยำโดยให้ผลที่ดีกว่าการใช้ชุดข้อมูลฝึกสอนจากแหล่งเดียว นอกเหนือจากนั้นแล้วยังพบว่าการเพิ่มจำนวน Epoch ให้มากขึ้นนั้นจะให้ผลลัพธ์ของการทำนายยีลด์ที่ดีขึ้นอีกด้วย ส่วนในปี ค.ศ. 2017 Toly Chen ได้เสนองานวิจัยที่มีแนวคิดใหม่โดยได้พื้นฐานมาจาก Neural Network และ Fuzzy logic โดยเรียกเทคนิคใหม่ที่ได้จากทั้ง 2 อัลกอริทึมว่า Heterogeneous Fuzzy Collaborative Intelligence Approach (Heterogeneous FCI) เทคนิควิธีการใหม่นี้ถูกเปรียบเทียบประสิทธิภาพกับวิธีการอื่น ๆ ที่แตกต่างกันออกไปอีก 8 วิธีการ ซึ่งผลลัพธ์ที่ได้นั้นพบว่าดีกว่าวิธีการอื่น ๆ อย่างชัดเจน โดยงานวิจัยชิ้นนี้ได้ใช้ตัววัดประสิทธิภาพถึงสามชนิดด้วยกัน ได้แก่ MAE, MAPE และ RMSE

ในปี ค.ศ. 2019 Lee Chia-Yen และ Tsai (Lee and Tsai, 2019) ได้ทำการทดลองนำโมเดลที่มาจาก Stepwise Regression, Decision Tree และ Random Forest มาใช้ในงานด้านการคัดเลือกคุณลักษณะ เพื่อหา Key Factor หรือปัจจัยหลักที่ส่งผลต่อการทำให้เกิดยีลด์และความผิดพลาดในกระบวนการผลิตของ TFT-LCD (Thin film Technology Liquid Crystal Display) โดยใช้อัลกอริทึม Partial Least Squares Regression และ Neural Network เป็นตัวแบบของ Yield Prediction ผลปรากฏว่าหากมีการคัดเลือกคุณลักษณะก่อนนำเข้าสู่กระบวนการทำนายแล้ว จะทำให้มีประสิทธิภาพการทำนายที่ดีขึ้นอย่างเห็นได้ชัดเจน

Unchalisa Taetragool และ Tiranee Achalakul (Taetragool and Achalakul, 2011) ได้ทำงานวิจัยเกี่ยวกับการหาสาเหตุและรูปแบบการเกิดผลิตภัณฑ์ที่ไม่ผ่านการทดสอบ (Failure Root Cause Analysis and Failure Pattern Analysis) ในอุตสากรรมการผลิตฮาร์ดดิสก์ไดรฟ์ ด้วยการ

ใช้อัลกอริทึม Decision Tree โดยจะมุ่งเน้นไปที่กระบวนการผลิต HGA: Head Gimbal Assembly ซึ่งเป็นเพียงชิ้นส่วนหนึ่งในฮาร์ดดิสก์ไดรฟ์เท่านั้น ผลลัพธ์ที่ได้จากงานวิจัยชิ้นนี้ทำให้สามารถทำการคัดเลือกคุณลักษณะที่มีความสำคัญต่อการเกิดผลิตภัณฑ์ที่ไม่ผ่านการทดสอบ ซึ่งทำให้เวลาในการประมวลผลของคอมพิวเตอร์ลดลงไปถึง 300% นอกเหนือจากนั้นแล้วยังทำให้วิศวกรที่ดูแลกระบวนการผลิตสามารถหาสาเหตุการเกิดผลิตภัณฑ์ที่ไม่ผ่านการทดสอบได้รวดเร็วยิ่งขึ้นอีกด้วย

Zhongwei Li และ คณะ (Li et al., 2014) นำเสนองานวิจัยที่ใช้อัลกอริทึม CART ในการทำนายผลลัพธ์ของการทดสอบผลิตภัณฑ์ฮาร์ดดิสก์ไดรฟ์ โดยเปรียบเทียบกับวิธีการเดิมที่ทำงานด้วยอัลกอริทึม BPANN (Back Propagation Artificial Neural Networks) ซึ่งผลลัพธ์ที่ได้นั้นพบว่าอัลกอริทึม CART ให้ประสิทธิภาพในการทำนายที่เทียบเท่ากับอัลกอริทึม BPANN โดยจะให้ค่า FAR (False Alarm Rate) ที่แย่กว่าเล็กน้อย แต่ให้ค่า FDR (Failure Detect Rate) ที่สูงกว่ามาก อีกทั้งอัลกอริทึม CART ยังมีข้อดีในด้านการนำไปใช้งานจริงเนื่องจากสามารถตีความได้ง่ายกว่า ANN

จากงานวิจัยที่ได้กล่าวมาทั้งหมดในข้างต้นนั้น สามารถนำมาทำการสรุปเปรียบเทียบกับงานวิจัยของวิทยานิพนธ์นี้ได้ตามตารางที่ 2.15

ตารางที่ 2.15 สรุปเปรียบเทียบงานวิจัยที่เกี่ยวข้อง

กระบวนการทำงาน	งานวิจัยที่เกี่ยวข้อง									
	ก	ข	ค	ง	จ	ฉ	ช	ซ	ฌ	ญ*
ชุดข้อมูลที่ใช้ในการทดสอบ										
อุตสาหกรรมการผลิตฮาร์ดดิสก์ไดรฟ์								✓	✓	✓
อุตสาหกรรมการผลิต				✓	✓	✓	✓			
ความพึงพอใจของลูกค้าธนาคาร		✓								
การเกิดความผิดพลาดของซอฟต์แวร์			✓							
ข้อมูลด้านอื่น ๆ	✓									
ลักษณะชุดข้อมูลที่ใช้										
Non-Public	✓		✓	✓	✓	✓	✓	✓	✓	✓
Public		✓								
ลักษณะและหัวข้อที่สำคัญในงานวิจัย										
การคาดการณ์ผลผลิต (Yield Prediction)				✓	✓	✓	✓			✓
การคัดเลือกคุณลักษณะ (Feature Selection)	✓		✓				✓	✓		✓
การจำแนกข้อมูล (Classification)	✓	✓	✓							✓
การเพิ่มประสิทธิภาพโดยการทำสมดุลข้อมูล		✓								✓
การวิเคราะห์ความเสียหาย (Failure Analysis)								✓	✓	
การจำแนกข้อมูลเพื่อช่วยในการคัดเลือกคุณลักษณะ	✓									
การประยุกต์เทคนิคและอัลกอริทึมต่าง ๆ เข้าด้วยกันเพื่อจัดการปัญหาข้อมูลที่ไม่สมดุล		✓								✓
การประยุกต์เทคนิคและอัลกอริทึมต่าง ๆ เข้าด้วยกันเพื่อเพิ่มประสิทธิภาพในการจำแนก			✓							
การเพิ่มประสิทธิภาพในการคาดการณ์ผลผลิตด้วยเทคนิคที่สร้างขึ้นมาใหม่										✓
การเพิ่มประสิทธิภาพในการคาดการณ์ผลผลิตด้วยการประยุกต์เทคนิควิธีการเดิมเข้าด้วยกัน				✓	✓	✓	✓			✓

ตารางที่ 2.15 สรุปเปรียบเทียบงานวิจัยที่เกี่ยวข้อง (ต่อ)

กระบวนการทำงาน	งานวิจัยที่เกี่ยวข้อง									
	ก	ข	ค	ง	จ	ฉ	ช	ซ	ฌ	ญ*
อัลกอริทึมและเทคนิคที่ใช้ในงานวิจัย										
k-Nearest Neighbors			✓							✓
k-Means Clustering										✓
Support Vector Machine (SVM)										✓
Decission Tree		✓	✓				✓	✓		✓
Regression Algorithms				✓	✓					✓
Artificial Neural Networks (ANN)			✓			✓	✓		✓	✓
Classification and Regression Tree (CART)	✓				✓		✓		✓	✓
Genetic Algorithm (GA)	✓				✓		✓		✓	✓
Statistic Chi-Square	✓									✓
Statistic Information Gain										✓
F-Statistic	✓									
Baysian Network		✓								
Naïve Bayes			✓							
มาตรวัดที่ใช้วัดประสิทธิภาพของโมเดล										
Accuracy		✓								
False Positive Rate									✓	
True Positive Rate									✓	
AUC	✓	✓								
Posterrior Median				✓						
Computational Time								✓		
Means Absolute Pecentage Error (MAPE)					✓					
Means Square Error (MSE)						✓	✓			
Means Absolute Error (MAE)										✓
Root Means Square Error (RMSE)						✓	✓			✓

หมายเหตุ งานวิจัยที่เกี่ยวข้องประกอบด้วย

ก แทนงานวิจัยของ Liu และ Zhou (2017)

ข แทนงานวิจัยของ Ruangthong และ Jaiyen (2016)

ค แทนงานวิจัยของ Turabieh และ คณะ (2019)

ง แทนงานวิจัยของ Yuan และ คณะ (2011)

จ แทนงานวิจัยของ Lee และ คณะ (2015)

ฉ แทนงานวิจัยของ Chen (2017)

ช แทนงานวิจัยของ Lee และ Tsai (2019)

ซ แทนงานวิจัยของ Taetragool และ Achalakul (2011)

ฌ แทนงานวิจัยของ Li และ คณะ (2014)

ญ* แทนงานวิจัยของวิทยานิพนธ์ฉบับนี้

บทที่ 3
วิธีดำเนินงานวิจัย

งานวิจัยชิ้นนี้มีวัตถุประสงค์เพื่อพัฒนาการคาดการณ์ผลผลิตในอุตสาหกรรมการผลิตฮาร์ดดิสก์ไดรฟ์ให้มีความแม่นยำที่มากขึ้น เริ่มจากการนำเสนอวิธีการเพื่อจัดการข้อมูลที่ไม่สมดุล อันเนื่องมาจากผลลัพธ์หรือสถานะของการทดสอบผลิตภัณฑ์นั้นจะมีอัตราส่วนความไม่สมดุล (Imbalance Ratio) ที่ค่อนข้างสูง โดยการประยุกต์ใช้องค์ความรู้ในการทำ k-Means Clustering ผนวกกับอัลกอริทึม k-Nearest Neighbors และสุ่มข้อมูลมาใช้งานร่วมกัน วิธีการทำสมดุลข้อมูลที่นำเสนอมีชื่อว่า DBC-2KAR (Data Balancing by k-Means Clustering k-Nearest Neighbors and Re-Sampling) หลังจากนั้นจึงนำวิธีการคัดเลือกคุณลักษณะ (Feature Selection) ทั้ง 7 ชนิดซึ่งมาจากอัลกอริทึมทางการเรียนรู้ของเครื่อง 5 ชนิด ได้แก่ Decision Tree (C5, CART), SVM, Stepwise Regression, Genetic Algorithm และวิธีการทางสถิติ 2 ชนิด ได้แก่ Chi-Square และ Information Gain ซึ่งเป็นที่นิยมและให้ผลลัพธ์ที่ดีในอุตสาหกรรมอิเล็กทรอนิกส์ซึ่งถือว่าเป็นอุตสาหกรรมที่ใกล้เคียงฮาร์ดดิสก์ไดรฟ์เพื่อนำมาใช้ในการจำแนกหรือหาค่าน้ำหนักที่ส่งผลมากที่สุดต่อผลลัพธ์ของกระบวนการทดสอบผลิตภัณฑ์ ขั้นตอนต่อไปจึงนำคุณลักษณะที่มีความสำคัญต่อผลการทดสอบฮาร์ดดิสก์ไดรฟ์ไปใช้ในการสร้างโมเดลการคาดการณ์ผลผลิต (Yield Prediction) ซึ่งอัลกอริทึมที่จะนำมาสร้างโมเดลการเรียนรู้นั้นมีด้วยกัน 2 ชนิด คือ MLR และ ANN โดยชุดข้อมูลที่ใช้ในการสร้างโมเดลจะถูกประยุกต์การรวมกลุ่มข้อมูล (Data Aggregation) ด้วยเทคนิคการใช้จำนวนค่าคงที่ เพื่อศึกษาว่าการเพิ่มจำนวนค่าคงที่ของการรวมกลุ่มนั้นจะส่งผลกระทบต่อประสิทธิภาพในการคาดการณ์ผลผลิตของกระบวนการหรือไม่อย่างไร

3.1 ข้อมูลที่นำมาใช้ในงานวิจัย

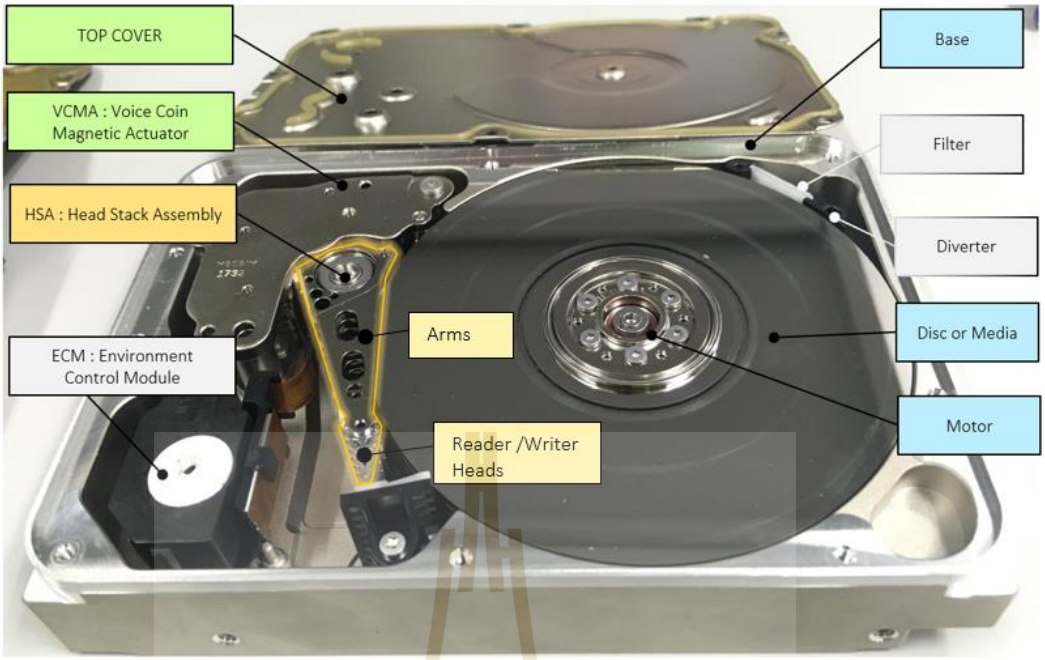
ชุดข้อมูลที่นำมาใช้ในงานวิจัยชิ้นนี้เป็นชุดข้อมูลที่ได้มาจากการเก็บรวบรวมข้อมูลทางการผลิตและทดสอบผลิตภัณฑ์ฮาร์ดดิสก์ไดรฟ์ของโรงงานแห่งหนึ่งเป็นระยะเวลายาวนานกว่า 3 ปี ซึ่งจะทำให้ได้ข้อมูลมาใช้ในงานวิจัยเป็นจำนวนประมาณ 10,000,000 แถว (Row) นั่นหมายความว่าข้อมูลนี้เป็นตัวแทนของข้อมูลการผลิตและทดสอบฮาร์ดดิสก์ไดรฟ์จำนวนถึงสิบล้านยูนิต และนอกเหนือจากนั้นแล้วในฮาร์ดดิสก์ไดรฟ์แต่ละยูนิตจะมีจำนวนคุณลักษณะซึ่งมีจำนวนมากกว่า 400 คอลัมน์ (Column) โดยค่าของแต่ละคุณลักษณะนั้นจะแสดงถึงส่วนประกอบทุกอย่างของ

ฮาร์ดดิสก์ยูนิตนั้น ๆ ซึ่งบอกรายละเอียดว่าชิ้นส่วนแต่ละชิ้นมาจากผู้ผลิตรายย่อยรายใด ใช้เครื่องจักรเครื่องใดในการประกอบชิ้นงานแต่ละชิ้นงาน สิ่งต่าง ๆ เหล่านี้สามารถทราบได้จากข้อมูลคุณลักษณะทั้งหมดที่มีอยู่ในฮาร์ดดิสก์ไดรฟ์

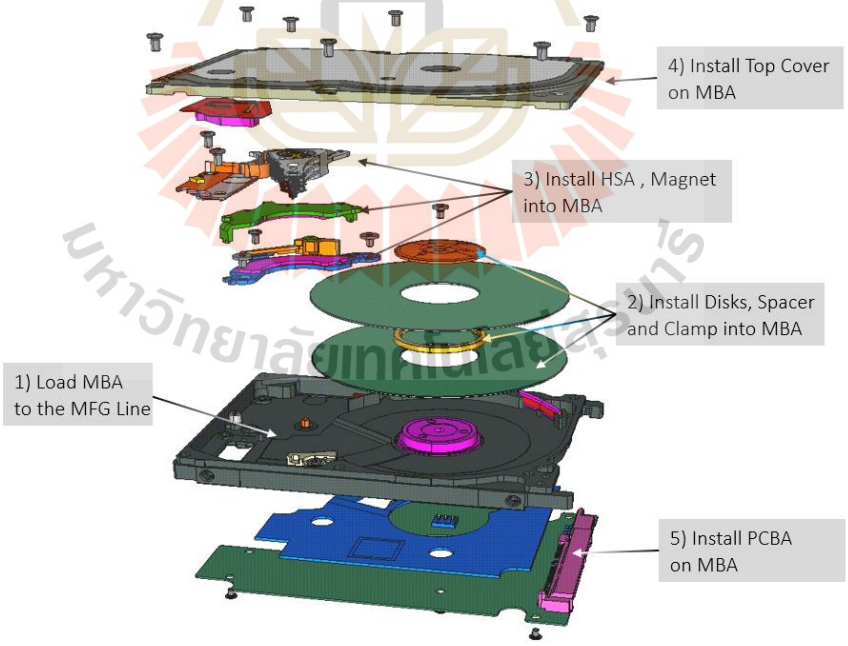
ฮาร์ดดิสก์ไดรฟ์แต่ละยูนิตนั้นประกอบด้วยชิ้นส่วนที่สำคัญหลายส่วน (Kiatwanidvilai and Praserttaweelap, 2018; Simon et al., 2018; Li et al., 2017; Nwosu et al., 2016; Samattapapong and Afzulpurkar, 2016; Sankar et al., 2013; Ye et al., 2013; Song et al., 2012) ซึ่งแต่ละชิ้นส่วนมีความสำคัญต่อการทำงานของฮาร์ดดิสก์ไดรฟ์ทั้งสิ้น แต่ในที่นี้จะกล่าวถึงชิ้นส่วนหลักบางชิ้นส่วน (แสดงไว้ในรูปที่ 3.1) เพื่อให้บุคคลทั่วไปที่ไม่ได้อยู่ในอุตสาหกรรมนี้สามารถทำความเข้าใจได้พอสังเขปโดยมีรายละเอียดดังต่อไปนี้

1. หัวอ่านเขียน (Reader / Writer Head) มีหน้าที่อ่านเขียนข้อมูลลงบนแผ่นจานบันทึก
2. แขน (Arm) มีหน้าที่เป็นชิ้นส่วนเพื่อรองรับและเคลื่อนที่หัวอ่านเขียน
3. แผ่นจานบันทึก (Media / Disk) มีหน้าที่เป็นสื่อบันทึกข้อมูล
4. มอเตอร์ (Motor) มีหน้าที่หมุนแผ่นจานบันทึก
5. ฐาน (Base) มีหน้าที่รองรับและปกป้องชิ้นส่วนต่าง ๆ ของฮาร์ดดิสก์ไดรฟ์
6. แท่งแม่เหล็กถาวร (VCMA : Voice Coil Magnetic Actuator) มีหน้าที่สร้างสนามแม่เหล็กไฟฟ้าเพื่อขับเคลื่อนแขนและหัวอ่านเขียน
7. โมดูลควบคุมสิ่งแวดล้อม (ECM : Environment Control Module) มีหน้าที่ควบคุมสิ่งแวดล้อมภายในฮาร์ดดิสก์ไดรฟ์ให้มีสภาวะที่เหมาะสมแก่การทำงาน
8. ฝาครอบ (Top Cover) มีหน้าที่ปิดผนึกฮาร์ดดิสก์ไดรฟ์เพื่อทำให้เกิดสภาพแวดล้อมแบบปิด
9. แผ่นวงจรควบคุม (PCBA: Printed Circuit Board Assembly) มีหน้าที่ควบคุมการทำงานของฮาร์ดดิสก์และเชื่อมต่อไปยังคอมพิวเตอร์ หรือตู้ทดสอบผลิตภัณฑ์

การประกอบชิ้นส่วนแต่ชิ้นเข้าด้วยกันนั้นจะกระทำใน Cleanroom โดยลำดับการประกอบนั้นได้แสดงไว้ในรูปที่ 3.2 โดยแต่ละสายการผลิต (Manufacturing Line) อาจจะมีศักยภาพในการผลิตที่แตกต่างกันไปบ้างและปัจจัยเหล่านี้อาจจะส่งผลต่อการทดสอบผลิตภัณฑ์



รูปที่ 3.1 ส่วนประกอบที่สำคัญของฮาร์ดดิสก์ไดรฟ์



รูปที่ 3.2 การประกอบชิ้นส่วนต่าง ๆ ของฮาร์ดดิสก์ไดรฟ์เข้าด้วยกัน

ข้อมูลคุณลักษณะกว่า 400 คุณลักษณะนั้นได้ถูกบันทึกไว้ในฐานข้อมูลการผลิตและทดสอบของฮาร์ดดิสก์ไดรฟ์แต่ละยูนิต โดยรูปที่ 3.3 จะเป็นตารางที่แสดงถึงตัวอย่างข้อมูลคุณลักษณะของฮาร์ดดิสก์ไดรฟ์แต่ละยูนิตโดยจะกล่าวถึงความหมายของคุณลักษณะที่สำคัญ ได้แก่

1. Drive SN (Drive Serial Number) หมายถึง รหัสประจำตัวฮาร์ดดิสก์ไดรฟ์ โดยฮาร์ดดิสก์ไดรฟ์แต่ละตัวนั้นจะมีคุณลักษณะนี้เพียงค่าเดียว และจะไม่ซ้ำกับฮาร์ดดิสก์ไดรฟ์ตัวอื่น

2. WEEK หมายถึง สัปดาห์ที่ฮาร์ดดิสก์ไดรฟ์ตัวนั้นได้ถูกประกอบขึ้นมาใน Cleanroom ซึ่งในแต่ละสัปดาห์นั้นอาจจะมีจำนวนฮาร์ดดิสก์ไดรฟ์ที่ถูกประกอบขึ้นมาเป็นจำนวนมาก น้อย หรืออาจจะไม่มีการประกอบขึ้นมาเลย ตามแต่การวางแผนการผลิต

3. STATUS หมายถึง สถานะของการทดสอบผลิตฮาร์ดดิสก์ไดรฟ์โดยจะมีสองค่า ได้แก่ “Pass” คือ สถานะที่แสดงถึงการที่ฮาร์ดดิสก์ไดรฟ์ตัวนั้นผ่านการทดสอบและพร้อมเข้าสู่กระบวนการถัดไป และ “Fail” คือ สถานะที่แสดงถึงการที่ฮาร์ดดิสก์ไดรฟ์ตัวนั้นไม่ผ่านการทดสอบ ซึ่งจะนำไปเข้าสู่กระบวนการซ่อมแซม ทดสอบซ้ำ หรือทำลาย ตามแต่การวินิจฉัยผลลัพธ์จากการทดสอบ

4. HSA_PR (HSA Prime-Rework) หมายถึง “สภาพของชิ้นส่วน” ของ HSA (Head Stack Assembly) โดยจะมีสองค่า ได้แก่ “Prime” คือ สภาพของชิ้นส่วนที่สร้างขึ้นมาใหม่และไม่เคยถูกประกอบลงไปในฮาร์ดดิสก์ไดรฟ์ตัวใดมาก่อน และ “Rework” คือ สถานะของชิ้นส่วนที่เคยถูกประกอบลงไปในฮาร์ดดิสก์ไดรฟ์ยูนิตอื่นมาก่อนหน้านี้ แต่ฮาร์ดดิสก์ไดรฟ์ยูนิตนั้นไม่ผ่านการทดสอบบางอย่างจึงถูกแยกชิ้นส่วน โดยชิ้นส่วนเหล่านั้นจะถูกนำไปเข้าสู่กระบวนการซ่อมแซม (Rework) และถูกนำมาประกอบลงไปในฮาร์ดดิสก์อีกยูนิตหนึ่ง

5. MEDIA_PR (Media Prime-Rework) หมายถึง สภาพของชิ้นส่วนของแผ่นจานบันทึก โดยจะมีสองค่า ได้แก่ “Prime” และ “Rework” โดยมีความหมายเหมือนกับ HSA

6. MBA_PR (MBA Prime-Rework) หมายถึง สภาพของชิ้นส่วนของ MBA (Motor-Base Assembly) โดยจะมีสองค่า ได้แก่ “Prime” และ “Rework” โดยมีความหมายเหมือน HSA

7. VCM_PR (Voice Coil Magnetic Prime-Rework) หมายถึง สภาพของชิ้นส่วนของชุดแท่งแม่เหล็กขับเคลื่อน HSA โดยจะมีสองค่า ได้แก่ “Prime” และ “Rework” โดยมีความหมายเหมือนกับ HSA

8. TC_PR (Top Cover Prime-Rework) หมายถึง สภาพของชิ้นส่วนของฝาครอบ โดยจะมีสองค่า ได้แก่ “Prime” และ “Rework” โดยมีความหมายเหมือนกับ HSA

9. PCBA_PR (Printed Circuit Board Assembly Prime-Rework) หมายถึง สภาพของชิ้นส่วน ของแผ่นวงจรควบคุมโดยจะมีสองค่า ได้แก่ “Prime” และ “Rework” โดยมีความหมายเหมือนกับ HSA

10. DB_Line (Drive Build line) หมายถึงไลน์การผลิตหรือสายเครื่องจักรการผลิต (Manufacturing Line) ในแต่ละสายเครื่องจักรการผลิตอาจจะมีศักยภาพในการประกอบชิ้นส่วนฮาร์ดดิสก์ไดรฟ์ในแต่ละรุ่นผลิตภัณฑ์ที่ไม่เท่ากัน ซึ่งในรุ่นผลิตภัณฑ์ชนิดที่นำมาใช้ในงานวิจัยมีจำนวนของสายเครื่องจักรการผลิตประมาณ 4 สายเครื่องจักรการผลิต

11. PCBA_Line หมายถึง สายเครื่องจักรการประกอบชิ้นส่วนแผ่น PCBA ลงบนตัวฮาร์ดดิสก์ไดรฟ์

12. HSA_Line หมายถึง สายเครื่องจักรการผลิต HSA ซึ่งจะเป็นการประกอบชิ้นส่วนเล็ก ๆ หลายชิ้นขึ้นมารวมกันกลายเป็น HSA ในขั้นตอนการผลิตนี้จะเป็นการทำงานร่วนกันระหว่างการผลิตโดยเครื่องจักรอัตโนมัติ (Automation Line) และการผลิตโดยใช้แรงงานฝีมือ (Manual Line)

นอกเหนือจากคุณลักษณะที่ได้ยกตัวอย่างมาข้างต้นนี้ยังมีคุณลักษณะที่มากมายอีกกว่า 400 คุณลักษณะซึ่งมีความหมายและรายละเอียดปลีกย่อยที่แตกต่างกันไป

Drive SN	WEEK	STATUS	HSA_PR*	MEDIA_PR*	MBA_PR*	VCM_PR*	TC_PR	PCBA_PR*	DB_Line	PCBA_Line	HSA_Line
SN0000001	WK01	Pass	Prime	Prime	Prime	Prime	Prime	Rework	DB_1	PCBA_2	HSA_3
SN0000002	WK01	Pass	Prime	Prime	Prime	Prime	Prime	Rework	DB_1	PCBA_2	HSA_3
SN0000003	WK01	Fail	Prime	Rework	Prime	Prime	Prime	Rework	DB_1	PCBA_2	HSA_3
SN0000004	WK01	Fail	Rework	Rework	Prime	Prime	Prime	Prime	DB_1	PCBA_2	HSA_3
SN0000005	WK01	Pass	Prime	Prime	Prime	Prime	Prime	Prime	DB_1	PCBA_2	HSA_1
SN0000006	WK01	Pass	Prime	Prime	Prime	Prime	Prime	Prime	DB_1	PCBA_2	HSA_1
SN0000007	WK01	Pass	Prime	Prime	Prime	Prime	Prime	Prime	DB_1	PCBA_2	HSA_1
SN0000008	WK01	Pass	Prime	Prime	Prime	Prime	Prime	Prime	DB_1	PCBA_2	HSA_1
SN0000009	WK01	Pass	Prime	Prime	Prime	Rework	Prime	Prime	DB_1	PCBA_1	HSA_1
SN0000010	WK01	Pass	Prime	Prime	Prime	Prime	Rework	Rework	DB_2	PCBA_1	HSA_1
SN0000011	WK02	Fail	Prime	Rework	Rework	Prime	Rework	Rework	DB_2	PCBA_1	HSA_2
SN0000012	WK02	Pass	Prime	Prime	Prime	Prime	Prime	Prime	DB_2	PCBA_1	HSA_2
SN0000013	WK02	Pass	Prime	Prime	Prime	Prime	Prime	Prime	DB_2	PCBA_1	HSA_2
SN0000014	WK02	Pass	Prime	Prime	Prime	Prime	Prime	Prime	DB_2	PCBA_1	HSA_2
SN0000015	WK03	Pass	Prime	Prime	Prime	Prime	Prime	Prime	DB_2	PCBA_1	HSA_2
SN0000016	WK03	Pass	Prime	Prime	Prime	Prime	Prime	Prime	DB_2	PCBA_1	HSA_2
SN0000017	WK03	Pass	Prime	Prime	Prime	Prime	Prime	Rework	DB_2	PCBA_1	HSA_2
SN0000018	WK03	Pass	Prime	Prime	Prime	Prime	Prime	Prime	DB_2	PCBA_1	HSA_2
SN0000019	WK03	Pass	Prime	Prime	Rework	Rework	Rework	Prime	DB_2	PCBA_1	HSA_2
SN0000020	WK03	Fail	Rework	Prime	Prime	Rework	Prime	Prime	DB_1	PCBA_1	HSA_3

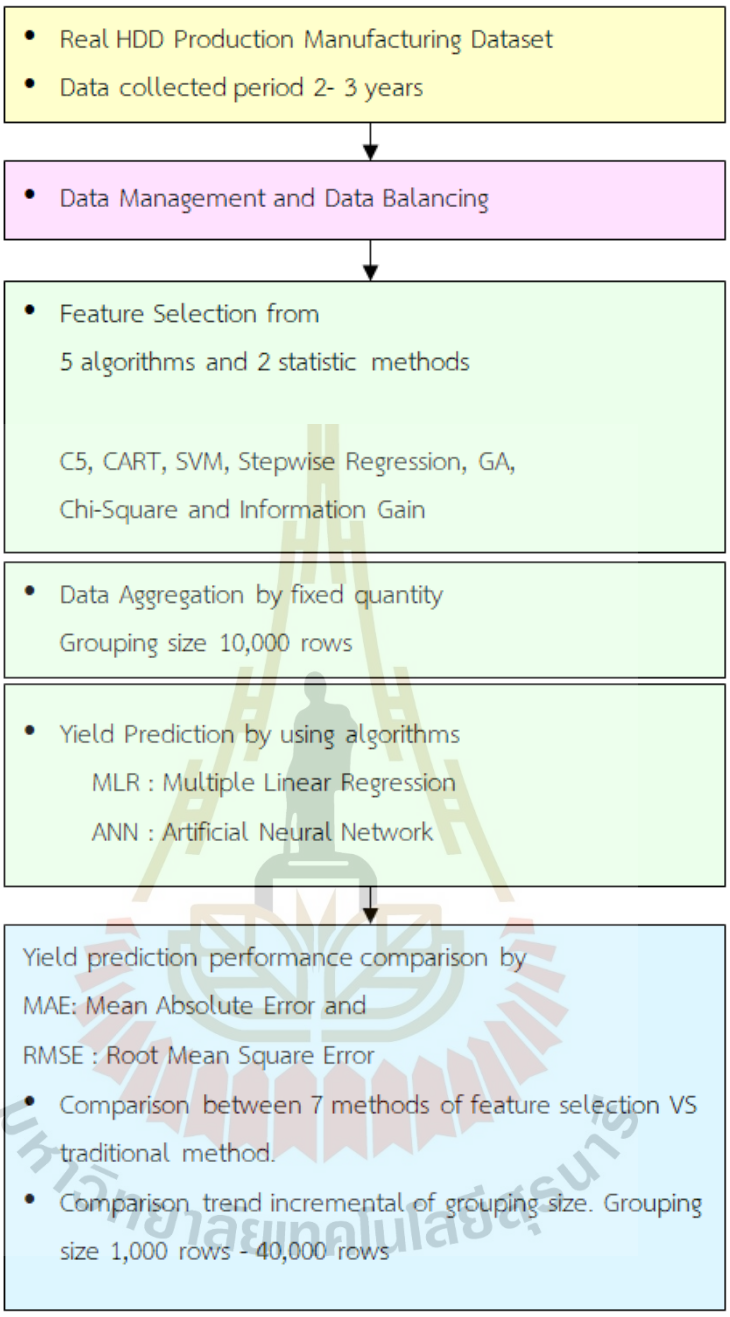
รูปที่ 3.3 ตัวอย่างแสดงคุณลักษณะของฮาร์ดดิสก์ไดรฟ์แต่ละยูนิต

3.2 กรอบแนวคิดและขั้นตอนการดำเนินงานวิจัย

กรอบการดำเนินงานวิจัยชิ้นนี้ถูกแบ่งออกเป็น 5 ส่วนหลักได้แก่

1. การจัดการข้อมูลที่นำมาใช้ในการทดลอง (Data Management and Data Balancing)
2. การคัดเลือกคุณลักษณะ (Feature Selection)
3. การรวมกลุ่มข้อมูล (Data Aggregation) ตามขนาดจำนวนข้อมูลที่คงที่
4. การสร้างโมเดลการคาดการณ์ผลผลิตกระบวนการ (Yield Prediction)
5. การประเมินผลโดยการเปรียบเทียบประสิทธิภาพการคาดการณ์ผลผลิตของกระบวนการ

โดยกรอบแนวคิดของงานวิจัยชิ้นนี้ได้ถูกแสดงไว้ในรูปแบบของแผนภาพ ในรูปที่ 3.4

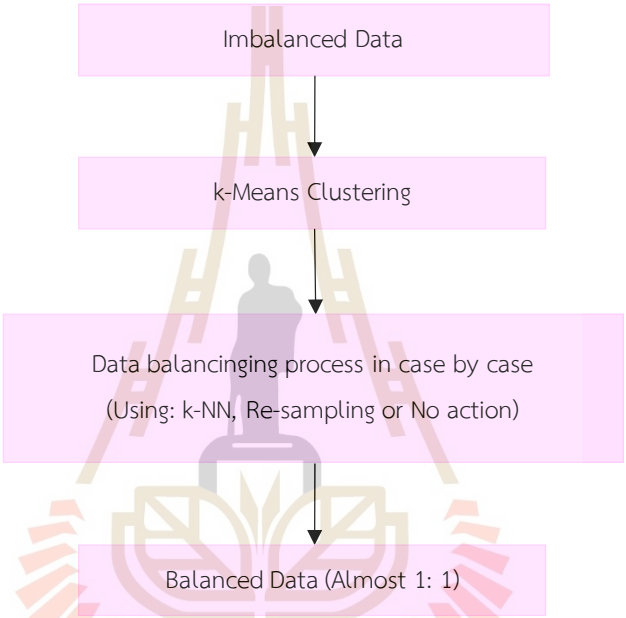


รูปที่ 3.4 กรอบแนวคิดของงานวิจัย

3.2.1 การดำเนินงานวิจัยในส่วนของการจัดการข้อมูล

ขั้นตอนนี้จะเป็นการจัดการข้อมูลให้อยู่ในรูปแบบที่เหมาะสมกับการดำเนินงานในขั้นตอนถัดไป เนื่องจากข้อมูลที่ใช้ในการทำงานวิจัยชิ้นนี้นั้นมาจากชุดข้อมูลจริงซึ่งมีจำนวนมหาศาล และนอกเหนือจากนั้นมีความไม่สมดุลของข้อมูลอยู่ในระดับที่สูงมาก ทางผู้วิจัยจึงต้องทำ

การนำชุดข้อมูลเข้าสู่กระบวนการเพื่อทำให้ข้อมูลมีความสมดุล (Data Balancing) ซึ่งเป็นการแก้ปัญหาไม่ให้โมเดลการจำแนกโน้มเอียงไปในทางคลาสเป้าหมายที่เป็นคลาสส่วนมากจนทำให้ประสิทธิภาพของการจำแนกลดลง (กีระชาติ สุขสุทธิ์, 2559; Lee and Kim, 2018; Sadrawi et al., 2018; Lin et al., 2017; Liu and Zhou 2017; Zhang et al., 2017; Martin-Diaz et al., 2016; Sun et al., 2015; Peng et al., 2014; Yang et al., 2011; Chawla, 2009; Liu and Zhou, 2006) โดยกระบวนการนี้สามารถแบ่งเป็นขั้นตอนย่อย ๆ ดังรูปที่ 3.5



รูป 3.5 แสดงขั้นตอนย่อยในการจัดการข้อมูลและทำข้อมูลให้สมดุล

จากรูปที่ 3.5 นำข้อมูลทางการผลิตฮาร์ดดิสก์ไดรฟ์ที่มีความไม่สมดุลอยู่ในอัตราที่สูงถึง 28 : 1 ระหว่างยูนิตที่ผ่านการทดสอบ (Pass unit) ซึ่งเป็นข้อมูลส่วนมาก และยูนิตที่ไม่ผ่านการทดสอบ (Fail unit) ซึ่งนับเป็นข้อมูลส่วนน้อย เข้าสู่อัลกอริทึม k-Means Clustering เพื่อจัดกลุ่มตามความคล้ายคลึงกันของข้อมูล (โดยกำหนดให้ค่า k = 5) หลังจากนั้นนำผลลัพธ์ที่ได้จากขั้นตอน Clustering มาพิจารณาว่าจะต้องใช้วิธีการใดต่อไปในการทำข้อมูลให้สมดุล และในขั้นตอนนี้จะมีการนำอัลกอริทึม k-NN ผนวกกับการทำ Re-sampling ข้อมูลเข้ามาช่วยในการทำให้ข้อมูลเกิดความสมดุลมากยิ่งขึ้น โดยจะมีกรณีที่อาจเป็นไปได้ถึง 5 กรณีด้วยกัน ซึ่งในแต่ละกรณีผู้วิจัยได้ออกแบบวิธีการจัดการข้อมูลที่ต่างกันออกไปโดยแสดงรายละเอียดไว้ในตารางที่ 3.1 ซึ่งจะทำให้ได้ผลลัพธ์

สุดท้ายของกระบวนการนี้คือทำให้ได้ข้อมูลที่มีความสมดุลกันระหว่างคลาสส่วนมากและคลาสส่วนน้อยที่อัตราส่วนใกล้เคียง 1:1

ตารางที่ 3.1 แสดงแนวทางและวิธีการจัดการข้อมูลให้สมดุลหลังจากขั้นตอน k-Means Clustering

กรณี	ลักษณะข้อมูลในแต่ละคลัสเตอร์	แนวทางการจัดการข้อมูล	วิธีการจัดการเพื่อทำให้ข้อมูลสมดุล
1	Pass > Fail	ลดจำนวนข้อมูล Pass ให้เท่ากับ Fail	หาตัวแทนของ Pass จากนั้นใช้เทคนิค k-NN ในการลดข้อมูล Pass ลงมาให้เท่ากับ Fail (โดยกำหนดให้ k = จำนวน Fail -1)
2	Pass < Fail	เพิ่มจำนวนข้อมูล Pass ให้เท่ากับ Fail	ใช้เทคนิคเพิ่มข้อมูลแบบสุ่มกับข้อมูล Pass ให้ใกล้เคียงกับ Fail
3	Pass = Fail	-	นำข้อมูลทั้งหมดใน Cluster นี้มาใช้งาน
4	Pass only	-	ไม่นำข้อมูลใน Cluster นี้มาใช้งาน
5	Fail only	-	ไม่นำข้อมูลใน Cluster นี้มาใช้งาน

จากตารางที่ 3.1 สามารถอธิบายแนวทางการทำข้อมูลให้สมดุลในแต่ละแต่ละกรณีได้ดังนี้

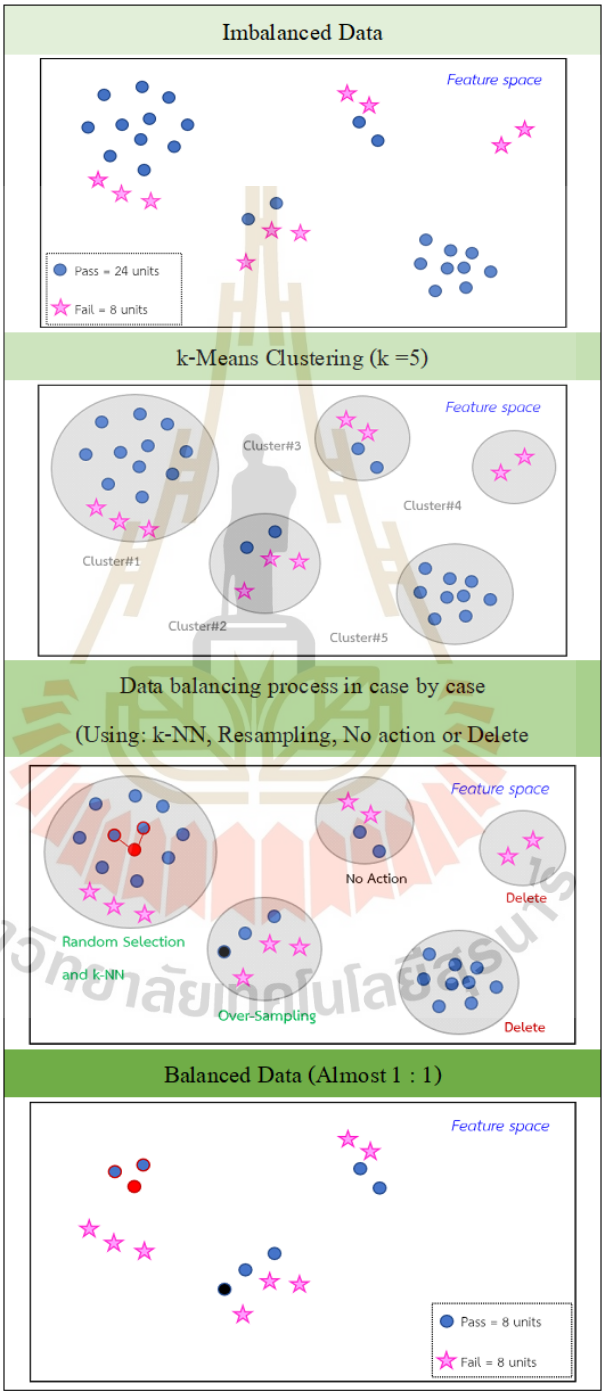
กรณีที่ 1 ภายในคลัสเตอร์มีจำนวนข้อมูล Pass unit มากกว่าจำนวน Fail unit เป้าหมาย คือต้องการลดจำนวนคลาสมากลง ส่วนคลาสน้อยให้คงไว้ โดยวิธีการ คือคัดเลือกตัวแทนของคลาสมากมา 1 ข้อมูล จากนั้นใช้เทคนิคเพื่อนบ้านที่ใกล้ที่สุด (k-NN) โดยกำหนดค่า k เท่ากับจำนวนข้อมูลของคลาสน้อย – 1 เพื่อใช้ในการคัดเลือกตัวแทนคลาสมาก ซึ่งหลังจากจบกระบวนการนี้จะทำให้ได้ข้อมูลที่สมดุลกันระหว่างคลาสส่วนมากและคลาสส่วนน้อย

กรณีที่ 2 ภายในคลัสเตอร์มีจำนวนข้อมูล Pass unit น้อยกว่าจำนวน Fail unit เป้าหมาย คือต้องการเพิ่มจำนวนข้อมูล Pass unit ให้เท่ากับจำนวน Fail unit มีวิธีการ คือการใช้เทคนิค Oversampling ในคลาสส่วนมากให้เพิ่มจำนวนจนเท่ากับจำนวนข้อมูลในคลาสส่วนน้อย

กรณีที่ 3 ภายในคลัสเตอร์มีจำนวนข้อมูล Pass unit เท่ากับจำนวน Fail unit ในกรณีนี้จะไม่ต้องทำอะไรกับข้อมูลที่อยู่ในคลัสเตอร์เพราะถือว่าข้อมูลสมดุลกันดีแล้ว

กรณีที่ 4 และกรณีที่ 5 ภายในคลัสเตอร์เดียวกันมีข้อมูล Pass unit หรือข้อมูล Fail unit เท่านั้น ซึ่งทั้งสองกรณีจะถือว่าข้อมูลที่อยู่ในคลัสเตอร์เหล่านี้ไม่ถูกเลือกเพื่อนำไปใช้งานต่อในกระบวนการถัดไป

แนวทางที่นำเสนอขึ้นใหม่นี้ผู้วิจัยเรียกว่าวิธี Data Balancing by k-Means Clustering k-Nearest Neighbors and Re-sampling (DBC-2KAR) โดยการจัดการในแต่ละกรณีได้แสดงเพิ่มเติมไว้ในรูปที่ 3.6



รูป 3.6 แสดงตัวอย่างการทำข้อมูลให้สมดุลในแต่ละขั้นตอนของวิธีการ DBC-2KAR

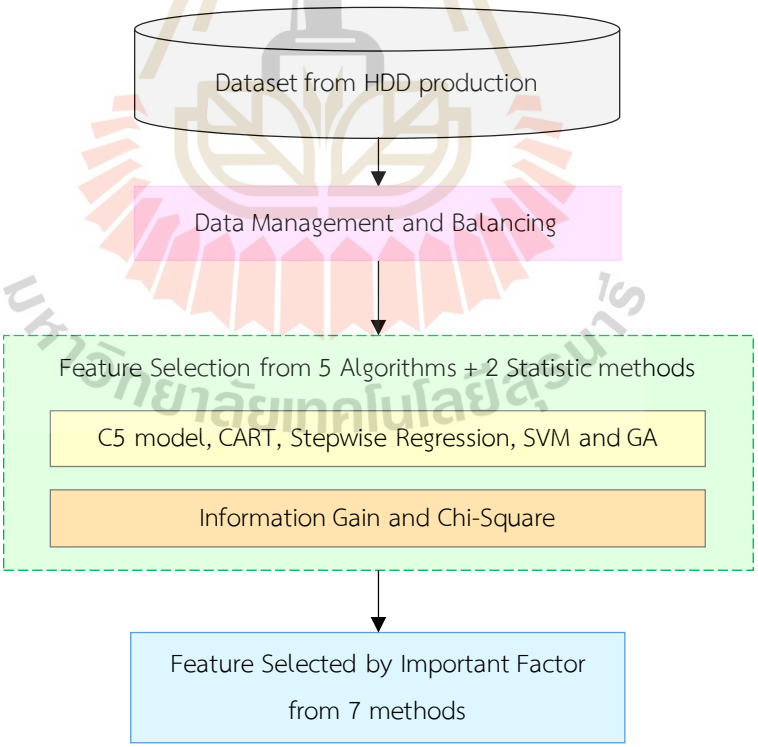
จากรูปที่ 3.6 รูปย่อยที่ 1 แสดงตัวอย่างของข้อมูลที่ไม่สมดุลของฮาร์ดดิสก์ไดรฟ์ พร้อมด้วยคุณลักษณะซึ่งวางอยู่บนฟีเจอร์สเปซรวมทั้งหมด 32 ยูนิต โดยมีไดรฟ์ที่ผ่านการทดสอบ 24 ยูนิต (ถูกแสดงด้วยสัญลักษณ์วงกลมสีน้ำเงิน) และไดรฟ์ที่ไม่ผ่านการทดสอบ 8 ยูนิต (ถูกแสดงด้วยสัญลักษณ์ดาวสีชมพู) สามารถคิดเป็นอัตราส่วนความไม่สมดุลได้เป็น 4 : 1 รูปย่อยที่ 2 ข้อมูลที่ไม่สมดุลนี้จะถูกนำไปเข้าสู่อัลกอริทึม k-Means Clustering ข้อมูลจะถูกจัดกลุ่มตามความคล้ายคลึงกันของข้อมูล โดยกำหนดค่า k เป็น 5 ซึ่งจะทำให้ได้ข้อมูล 5 คลัสเตอร์ และหลังจากพิจารณาในแต่ละคลัสเตอร์ พบว่ามีจำนวนไดรฟ์ที่ผ่านและไม่ผ่านการทดสอบในแต่ละคลัสเตอร์ดังนี้ คลัสเตอร์ที่หนึ่ง 11 : 3 ยูนิต คลัสเตอร์ที่สอง 2 : 3 ยูนิต คลัสเตอร์ที่สาม 2 : 2 ยูนิต คลัสเตอร์ที่สี่ 0 : 2 ยูนิต และคลัสเตอร์ที่ห้า 9 : 0 ยูนิต ซึ่งจะนำไปสู่วิธีการจัดการข้อมูลในแต่ละรูปแบบซึ่งถูกแสดงในรูปย่อยที่ 3 โดยคลัสเตอร์ที่หนึ่ง ซึ่งมีจำนวนไดรฟ์ที่ผ่านการทดสอบมากกว่าไดรฟ์ที่ไม่ผ่านการทดสอบ จะถูกใช้วิธี Random Sampling กับข้อมูลซึ่งเป็นไดรฟ์ที่ผ่านการทดสอบมาหนึ่งข้อมูล (ถูกแสดงด้วยสัญลักษณ์วงกลมสีแดง) จากนั้นใช้อัลกอริทึม k-Nearest Neighbors เพื่อหาเพื่อนบ้านที่ใกล้ที่สุดมาให้เท่ากับจำนวนไดรฟ์ที่ไม่ผ่านการทดสอบในคลัสเตอร์นี้ลบด้วยค่า 1 ซึ่งเท่ากับ 2 (ถูกแสดงด้วยสัญลักษณ์วงกลมสีน้ำเงินขอบสีแดง) หลังจบกระบวนการนี้จะทำให้ได้ข้อมูลจำนวนไดรฟ์ที่ผ่านและไม่ผ่านการทดสอบในคลัสเตอร์ที่หนึ่งเป็น 3 : 3 ยูนิต ส่วนในคลัสเตอร์ที่สอง ซึ่งมีจำนวนไดรฟ์ที่ผ่านการทดสอบน้อยกว่าไดรฟ์ที่ไม่ผ่านการทดสอบ จะใช้วิธี Over-Sampling กับข้อมูลซึ่งเป็นไดรฟ์ที่ผ่านการทดสอบมาหนึ่งข้อมูล ถูกแสดงด้วยสัญลักษณ์วงกลมสีดำ) หลังจบกระบวนการนี้จะทำให้ได้ข้อมูลจำนวนไดรฟ์ที่ผ่านและไม่ผ่านการทดสอบในคลัสเตอร์ที่สองเป็น 3 : 3 ยูนิต ส่วนในคลัสเตอร์ที่สามมีจำนวนไดรฟ์ที่ผ่านการทดสอบและไดรฟ์ที่ไม่ผ่านการทดสอบเท่ากัน ซึ่งถือว่ามีความสมดุลของข้อมูลดีแล้วจึงไม่ต้องมีกระบวนการจัดการข้อมูลในคลัสเตอร์นี้ ส่วนในคลัสเตอร์ที่สี่ ซึ่งมีข้อมูลของไดรฟ์ที่ไม่ผ่านการทดสอบเท่านั้น ดังนั้นข้อมูลในคลัสเตอร์นี้จะไม่ถูกนำมาใช้งาน เช่นเดียวกันกับคลัสเตอร์ที่ห้าซึ่งมีข้อมูลของไดรฟ์ที่ผ่านการทดสอบเท่านั้น ข้อมูลในคลัสเตอร์นี้จะไม่ถูกนำมาใช้งานเช่นกัน จากนั้นนำข้อมูลในคลัสเตอร์ที่หนึ่ง คลัสเตอร์ที่สอง และคลัสเตอร์ที่สามมารวมกัน จะได้ข้อมูลที่มีความสมดุลกันของฮาร์ดดิสก์ไดรฟ์ระหว่างข้อมูลของไดรฟ์ที่ผ่านการทดสอบ 8 ยูนิตและไดรฟ์ไม่ผ่านการทดสอบ 8 ยูนิตซึ่งเป็นอัตราส่วนที่สมดุล 1 : 1 ดังรูปย่อยที่ 4 หลังจากนั้นจะทำการนำข้อมูลนี้ไปใช้ในกระบวนการคัดเลือกคุณลักษณะในกระบวนการถัดไป

วิธีการ DBC-2KAR สามารถใช้เป็นแนวทางเพื่อช่วยจัดการกับปัญหาข้อมูลไม่สมดุลได้ในทุกชุดข้อมูล โดยไม่ได้อิงกับเฉพาะข้อมูลชุดนี้เท่านั้น หากมีการเปลี่ยนไปใช้ชุดข้อมูลอื่น วิธีการนี้สามารถนำไปปรับใช้ต่อได้ในส่วนของการปรับสมดุลข้อมูล โดยทำให้รู้ทิศทางว่า

หลังจากได้ผลลัพธ์ในแต่ละคลัสเตอร์แล้ว คลัสเตอร์ใดควรสุ่มเพิ่มข้อมูลหรือควรสุ่มลดข้อมูล คลัสเตอร์ใดสามารถนำข้อมูลมาใช้ได้เลยหรือไม่สามารถนำข้อมูลมาใช้ได้ ซึ่งผู้วิจัยคิดว่าวิธีการนี้จะเป็นประโยชน์แก่ผู้ที่สนใจศึกษาไม่มากก็น้อย

3.2.2 การดำเนินงานวิจัยในส่วนของการคัดเลือกคุณลักษณะ

การทดลองในขั้นตอนนี้จะเป็นส่วนของการคัดเลือกคุณลักษณะ เริ่มจากการนำชุดข้อมูลเข้าสู่กระบวนการคัดเลือกคุณลักษณะทั้ง 7 ชนิด (Zhou et al., 2021; Xu et al., 2020; Arefnezhad et al., 2019; Cheng et al., 2018; Khalid et al., 2014; Köksal et al., 2011) โดยจะแบ่งเป็นการนำอัลกอริทึมการเรียนรู้เพื่อสร้างโมเดลการจำแนก (Classification) 5 ชนิด มาประยุกต์ใช้งานในการคัดเลือกคุณลักษณะ ได้แก่ C5, CART, SVM, Stepwise Regression และ Genetic Algorithm นอกเหนือจากอัลกอริทึมการเรียนรู้ทั้ง 5 ชนิด ทางผู้วิจัยได้ทดลองนำวิธีทางสถิติเข้ามาประยุกต์ใช้ในการคัดเลือกอีก 2 ชนิด ได้แก่ Chi-Square และ Information Gain ซึ่งวิธีการทั้ง 2 ชนิดนี้จะมีการจัดเรียงคุณลักษณะตามความสำคัญต่อคลาสเป้าหมาย (Important Factor) โดยการพิจารณาจากค่าน้ำหนัก (Weight) ซึ่งแผนภาพการทำงานของขั้นตอนนี้ได้ถูกแสดงไว้ในรูปที่ 3.7



รูปที่ 3.7 แผนภาพแสดงการทำงานในขั้นตอนการคัดเลือกคุณลักษณะ

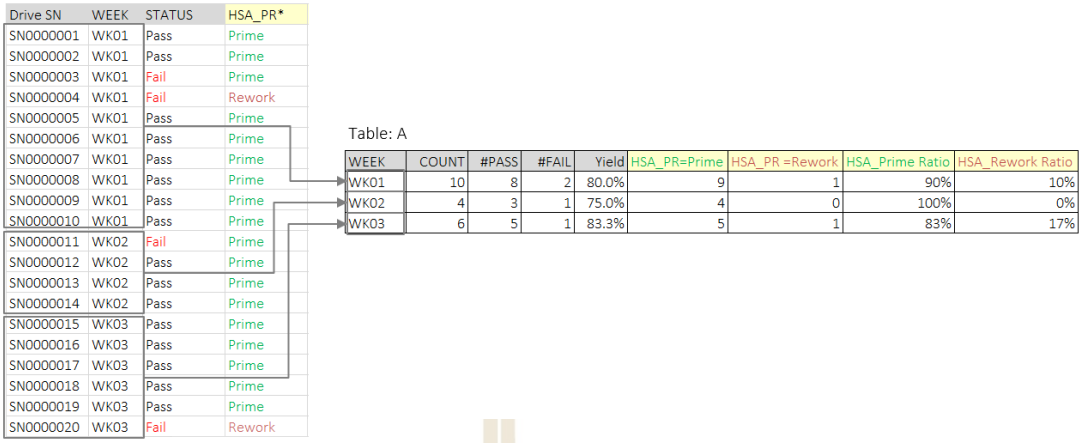
3.2.3 การรวมกลุ่มข้อมูลตามขนาดจำนวนข้อมูลที่คงที่

ส่วนนี้เป็นการนำเสนอแนวคิดใหม่เพื่อเพิ่มประสิทธิภาพการรวมกลุ่มข้อมูลของขั้นตอนการคำนวณผลผลิต (Yield Calculation) เพื่อใช้ในการคาดการณ์ผลผลิตของกระบวนการ (Yield Prediction) วิธีการดั้งเดิมของการทำงานในขั้นตอนนี้จะใช้การรวมกลุ่มข้อมูลโดยอ้างอิงตามระยะเวลาของปฏิทิน ซึ่งอาจจะเป็นการรวมข้อมูลตามปี เดือน วัน หรือสัปดาห์ โดยปัจจุบันการรวมข้อมูลที่นิยมจะเป็นการรวมกลุ่มตามรายสัปดาห์ เนื่องจากจะทำให้ได้จำนวนข้อมูลที่มากเพียงพอและเหมาะสมต่อการนำไปใช้ในการคำนวณผลผลิตและคาดการณ์ผลผลิต ตัวอย่างเช่นจากชุดข้อมูลสมมติที่แสดงไว้ในรูปที่ 3.3 ด้วยวิธีการรวมกลุ่มข้อมูลตามรายสัปดาห์นั้นจะทำให้เกิดตารางข้อมูลใหม่ตาม Table: A ของรูปที่ 3.8 โดยสามารถตีความตามตารางได้ดังนี้

ใน WK01 มีฮาร์ดดิสก์ไดรฟ์ถูกผลิตทั้งหมด 10 ยูนิต มีฮาร์ดดิสก์ไดรฟ์ที่ผ่านกระบวนการทดสอบ 8 ยูนิต และไม่ผ่านกระบวนการทดสอบ 2 ยูนิตสามารถคำนวณออกมาเป็น Yield ได้ 80% โดยมีจำนวนฮาร์ดดิสก์ไดรฟ์ที่ประกอบจาก “HSA = Prime” 9 ยูนิต “HSA = Rework” 1 ยูนิตสามารถคำนวณออกมาเป็น HSA Prime Ratio ได้ 90%

ใน WK02 มีฮาร์ดดิสก์ไดรฟ์ถูกผลิตทั้งหมด 4 ยูนิต ฮาร์ดดิสก์ไดรฟ์ทั้งหมด 3 ยูนิต ผ่านกระบวนการทดสอบ และไม่ผ่านกระบวนการทดสอบ 1 ยูนิตสามารถคำนวณออกมาเป็น Yield 75% โดยมีจำนวนฮาร์ดดิสก์ไดรฟ์ที่ประกอบจาก “HSA = Prime” 4 ตัว สามารถคำนวณออกมาเป็น HSA Prime Ratio = 100%

ใน WK03 มีฮาร์ดดิสก์ไดรฟ์ถูกผลิตทั้งหมด 6 ยูนิต มีฮาร์ดดิสก์ไดรฟ์ที่ผ่านกระบวนการทดสอบ 5 ยูนิต และไม่ผ่านกระบวนการทดสอบ 1 ยูนิตสามารถคำนวณออกมาเป็น Yield = 83.3% โดยมีจำนวนฮาร์ดดิสก์ไดรฟ์ที่ประกอบจาก “HSA = Prime” 5 ยูนิต “HSA = Rework” 1 ยูนิตสามารถคำนวณออกมาเป็น HSA Prime Ratio = 83%



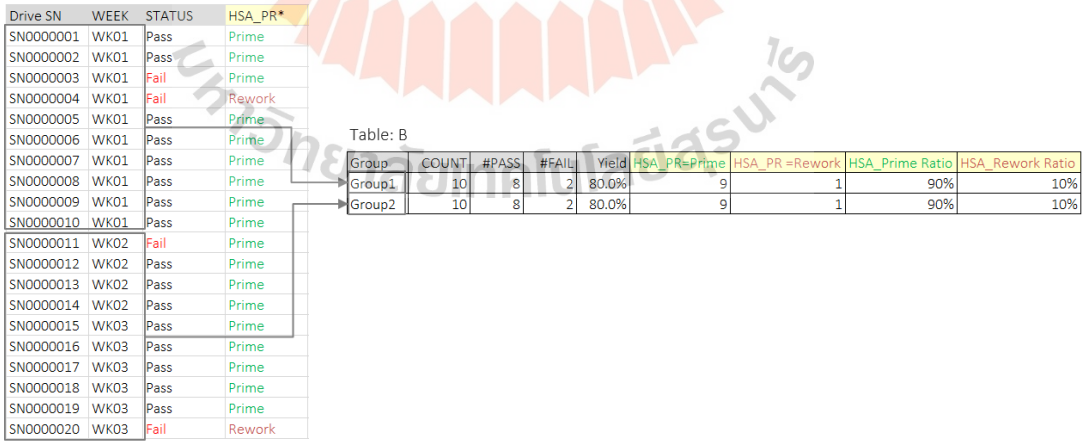
Drive SN	WEEK	STATUS	HSA_PR*
SN0000001	WK01	Pass	Prime
SN0000002	WK01	Pass	Prime
SN0000003	WK01	Fail	Prime
SN0000004	WK01	Fail	Rework
SN0000005	WK01	Pass	Prime
SN0000006	WK01	Pass	Prime
SN0000007	WK01	Pass	Prime
SN0000008	WK01	Pass	Prime
SN0000009	WK01	Pass	Prime
SN0000010	WK01	Pass	Prime
SN0000011	WK02	Fail	Prime
SN0000012	WK02	Pass	Prime
SN0000013	WK02	Pass	Prime
SN0000014	WK02	Pass	Prime
SN0000015	WK03	Pass	Prime
SN0000016	WK03	Pass	Prime
SN0000017	WK03	Pass	Prime
SN0000018	WK03	Pass	Prime
SN0000019	WK03	Pass	Prime
SN0000020	WK03	Fail	Rework

Table: A

WEEK	COUNT	#PASS	#FAIL	Yield	HSA_PR=Prime	HSA_PR =Rework	HSA_Prime Ratio	HSA_Rework Ratio
WK01	10	8	2	80.0%	9	1	90%	10%
WK02	4	3	1	75.0%	4	0	100%	0%
WK03	6	5	1	83.3%	5	1	83%	17%

รูปที่ 3.8 การรวมกลุ่มข้อมูลแบบรายสัปดาห์

แนวคิดใหม่ของงานวิจัยชิ้นนี้จะเป็นการรวมกลุ่มของข้อมูลในลักษณะของการแบ่งกลุ่มข้อมูลตามจำนวนคงที่ (Fixed number) ที่กำหนดไว้ ตัวอย่างเช่นการรวมกลุ่มละ 10 แถว โดยจะทำการเรียงข้อมูลทั้งหมดตามเวลาที่ผลิตออกมา จากนั้นจึงทำการกำกับชื่อกลุ่มลงไปโดยให้ 10 ยูนิตแรก (ฮาร์ดดิสก์ไดรฟ์ยูนิตที่ 1 ถึงยูนิตที่ 10) เป็นกลุ่มที่ 1 จากนั้น 10 ยูนิตถัดไป (ฮาร์ดดิสก์ไดรฟ์ยูนิตที่ 11 ถึงยูนิตที่ 20) เป็นกลุ่มที่ 2 ซึ่งตัวอย่างวิธีการรวมกลุ่มข้อมูลด้วยค่าคงที่ถูกแสดงไว้ในรูปที่ 3.9



Drive SN	WEEK	STATUS	HSA_PR*
SN0000001	WK01	Pass	Prime
SN0000002	WK01	Pass	Prime
SN0000003	WK01	Fail	Prime
SN0000004	WK01	Fail	Rework
SN0000005	WK01	Pass	Prime
SN0000006	WK01	Pass	Prime
SN0000007	WK01	Pass	Prime
SN0000008	WK01	Pass	Prime
SN0000009	WK01	Pass	Prime
SN0000010	WK01	Pass	Prime
SN0000011	WK02	Fail	Prime
SN0000012	WK02	Pass	Prime
SN0000013	WK02	Pass	Prime
SN0000014	WK02	Pass	Prime
SN0000015	WK03	Pass	Prime
SN0000016	WK03	Pass	Prime
SN0000017	WK03	Pass	Prime
SN0000018	WK03	Pass	Prime
SN0000019	WK03	Pass	Prime
SN0000020	WK03	Fail	Rework

Table: B

Group	COUNT	#PASS	#FAIL	Yield	HSA_PR=Prime	HSA_PR =Rework	HSA_Prime Ratio	HSA_Rework Ratio
Group1	10	8	2	80.0%	9	1	90%	10%
Group2	10	8	2	80.0%	9	1	90%	10%

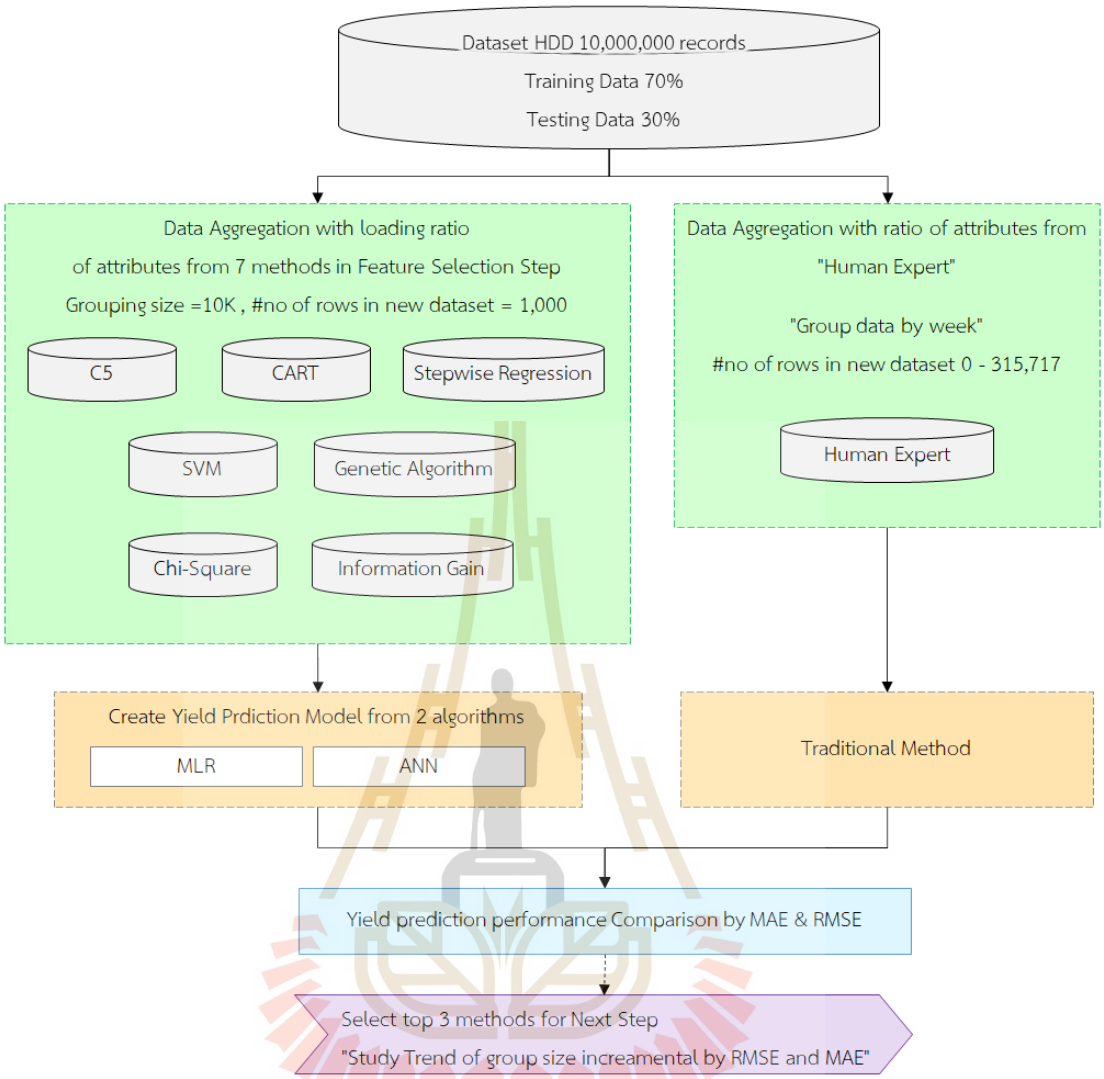
รูปที่ 3.9 แสดงการรวมกลุ่มข้อมูลด้วยค่าคงที่ของจำนวนข้อมูลในแต่ละกลุ่มเท่ากับ 10

จากรูปที่ 3.9 แสดงให้เห็นถึงการรวมกลุ่มข้อมูลด้วยวิธีการกำหนดค่าคงที่ของจำนวนการแบ่งกลุ่มเท่ากับ 10 ข้อมูลต่อ 1 กลุ่ม ทำให้ตารางใหม่ที่สร้างขึ้นมานั้นจะมีจำนวนข้อมูลกลุ่มละ 10 ข้อมูลเท่ากัน

ค่าคงที่ที่นำมาใช้ในการรวมกลุ่มข้อมูลนั้นทางผู้วิจัยได้ทดลองเริ่มต้นที่กลุ่มละ 10,000 โดยจะเพิ่มขึ้นไปเรื่อย ๆ ครั้งละ 10,000 จนกระทั่งถึงกลุ่มละ 40,000 ข้อมูล โดยการเพิ่มค่าคงที่ในลักษณะนี้เพื่อเป็นการศึกษาว่าจำนวนข้อมูลในกลุ่มที่สูงขึ้นนั้นจะส่งผลกระทบอย่างไรต่อประสิทธิภาพในการคาดการณ์ผลผลิตของกระบวนการ เนื่องจากทางผู้วิจัยได้เคยศึกษาเรื่องการรวมกลุ่มข้อมูลและได้ตีพิมพ์งานวิจัย “Efficient Machine Learning Methods for Hard Disk Drive Yield prediction Improvement” (Hirunyawanakul et al., 2020) โดยในงานวิจัยดังกล่าวพบว่าการรวมกลุ่มของข้อมูลจะให้ผลลัพธ์ที่ดีที่สุดเมื่อใช้ค่าคงที่ของการรวมกลุ่มข้อมูลเท่ากับ 10,000 โดยจะมีการทดลองด้วยค่าคงที่เท่ากับ 1,000 และ 5,000 ดังนั้นในขั้นตอนการรวมกลุ่มของงานวิจัยนี้จะทำการเพิ่มค่าคงที่ของการรวมกลุ่มข้อมูลที่ละเอียดยิ่งขึ้น โดยจะมีค่าคงที่ของการรวมกลุ่มข้อมูลเท่ากับ 1,000 2,000 และ 5,000 เข้าไปเพื่อแสดงให้เห็นถึงแนวโน้มที่มีความชัดเจนมากยิ่งขึ้นโดยการอธิบายถึงภาพรวมของขั้นตอนนี้จะอยู่ในช่วงต้นของแผนภาพกระบวนการทำงานรูปที่ 3.10

3.2.4 ส่วนการสร้างโมเดลการคาดการณ์ผลผลิตของกระบวนการ

การดำเนินงานในส่วนนี้จะเป็นส่วนที่สำคัญที่สุดของงานวิจัยเนื่องจากเป็นขั้นตอนในการสร้างโมเดลเพื่อทำนายยีลด์ ข้อมูลหลังจากการรวมกลุ่มตามค่าคงที่แล้วจะทำให้มีชุดข้อมูลที่เกิดขึ้นมาใหม่ 7 ชุดข้อมูล ตามวิธีการคัดเลือกคุณลักษณะทั้ง 7 โดยอัลกอริทึมที่นำมาใช้ในขั้นตอนการสร้างโมเดลการเรียนรู้จากชุดข้อมูลใหม่มี 2 ชนิด ได้แก่ MLR (Multiple Linear Regression) และ ANN (Artificial Neural Networks) (An et al., 2019; Cheng et al., 2018; Chen, 2017; Lee et al., 2015; Köksal et al., 2011; Yuan et al., 2011) โดยขั้นตอนการทำงานในกระบวนการนี้ได้แสดงไว้ในรูปที่ 3.10



รูปที่ 3.10 ขั้นตอนการทำงานวิจัยในส่วนของการรวมกลุ่มข้อมูลตามค่าคงที่ที่กำหนด การสร้างโมเดลทำนายยีลด์ และการวัดประสิทธิภาพของการคาดการณ์ผลผลิตของกระบวนการ

3.2.5 การประเมินผลด้านประสิทธิภาพการคาดการณ์ผลผลิต

ส่วนนี้จะเป็นส่วนที่แสดงผลลัพธ์ของงานวิจัยโดยการเปรียบเทียบประสิทธิภาพในการคาดการณ์ผลผลิตของกระบวนการในแต่ละอัลกอริทึม (การสร้างโมเดลและทดสอบโมเดล) ซึ่งพิจารณาจากผลการใช้ข้อมูลทดสอบโมเดลเป็นหลัก โดยมาตรวัดประสิทธิภาพของขั้นตอนนี้คือ MAE และ RMSE รวมไปจนถึงพิจารณาผลกระทบหรือแนวโน้มจากการเพิ่มค่าคงที่ของจำนวนข้อมูลว่าส่งผลต่อการคาดการณ์ผลผลิตของกระบวนการมากน้อยเพียงใด

3.3 ตัวอย่างข้อมูลที่ใช้ในการดำเนินงาน

ส่วนนี้เป็นการแสดงตัวอย่างของข้อมูลจำลองที่นำมาใช้ในการดำเนินงานวิจัย รวมไปถึงการเปลี่ยนแปลงของชุดข้อมูลในแต่ละขั้นตอนตั้งแต่ชุดข้อมูลตั้งต้นจนจบที่ตารางข้อมูลการนำไปใช้เพื่อการคาดการณ์ผลผลิต เนื่องจากชุดข้อมูลจริงที่นำมาใช้ในงานวิจัยชิ้นนี้มีจำนวนประมาณ 10,000,000 แถว ซึ่งจะทำให้เป็นการยากที่จะแสดงกระบวนการอันซับซ้อนด้วยจำนวนข้อมูลอันมหาศาลตามที่กล่าวมา ทางผู้วิจัยจึงสร้างชุดข้อมูลจำลองที่แสดงถึงการผลิตของฮาร์ดดิสก์ไดรฟ์ 20 ยูนิต โดยชุดข้อมูลนี้จะมีขนาดเพียง 20 แถว เพื่อใช้ในการอธิบายลักษณะของข้อมูลในแต่ละขั้นตอนการดำเนินงาน

Drive SN	WEEK	STATUS	HSA PR	MEDIA PR	MBA PR	PCBA PR	DB Line	…	ATR400
SN-001	WK01	Pass	Prime	Prime	Prime	RCY	DB_1	..	0
SN-002	WK01	Pass	Prime	Prime	Prime	RCY	DB_1	..	0
SN-003	WK01	Fail	Prime	RCY	Prime	Prime	DB_1	..	1
SN-004	WK01	Fail	RCY	RCY	Prime	Prime	DB_1	..	0
SN-005	WK01	Pass	Prime	Prime	Prime	Prime	DB_1	..	0
SN-006	WK01	Pass	Prime	Prime	Prime	Prime	DB_1	..	0
SN-007	WK01	Pass	Prime	Prime	Prime	Prime	DB_1	..	0
SN-008	WK01	Pass	Prime	Prime	Prime	Prime	DB_1	..	0
SN-009	WK01	Pass	Prime	Prime	Prime	Prime	DB_1	..	1
SN-010	WK01	Pass	Prime	Prime	Prime	Prime	DB_2	..	1
SN-011	WK02	Fail	Prime	RCY	RCY	RCY	DB_2	..	1
SN-012	WK02	Pass	Prime	Prime	Prime	RCY	DB_2	..	1
SN-013	WK02	Pass	Prime	Prime	Prime	Prime	DB_2	..	1
SN-014	WK02	Pass	Prime	Prime	Prime	Prime	DB_2	..	0
SN-015	WK03	Pass	Prime	Prime	Prime	Prime	DB_2	..	0
SN-016	WK03	Pass	Prime	Prime	Prime	Prime	DB_2	..	0
SN-017	WK03	Pass	Prime	Prime	Prime	RCY	DB_2	..	1
SN-018	WK03	Pass	Prime	Prime	Prime	Prime	DB_2	..	0
SN-019	WK03	Pass	Prime	Prime	RCY	Prime	DB_2	..	1
SN-020	WK03	Fail	RCY	Prime	Prime	Prime	DB_1	..	0

รูปที่ 3.11 ข้อมูลตั้งต้นที่จำลองขึ้นเพื่อประกอบการอธิบาย

รูปที่ 3.11 แสดงถึงข้อมูลจำลองที่อยู่ในรูปแบบของตาราง มีจำนวนแถวเท่ากับ 20 แถวตามจำนวนของฮาร์ดดิสก์ไดรฟ์ โดยมีข้อมูล Drive SN (Drive Serial Number) ซึ่งเป็นคอลัมน์แรกสุดที่จะไม่ซ้ำกันในแต่ละแถว ข้อมูลของสัปดาห์การผลิต สถานะการทดสอบ และข้อมูลของคุณลักษณะอื่น ๆ ได้ถูกแสดงไว้ในคอลัมน์ถัดมาตามลำดับ โดยในข้อมูลจำลองชุดนี้จะมีจำนวนคุณลักษณะทั้งหมด 400 คุณลักษณะ

ขั้นตอนแรกของงานวิจัยนี้คือการจัดการกับข้อมูลและการทำสมดุลข้อมูล โดยขั้นตอนนี้เป็นการทำให้จำนวนข้อมูลของสถานะ Pass และ Fail เท่ากันที่อย่างละ 4 ยูนิต ดังนั้นจำนวนแถวของชุดข้อมูลดั้งเดิมจะถูกลดลงจาก 20 แถวลงไปเหลือ 8 แถว ดังที่ได้แสดงไว้ในรูปที่ 3.12

Drive SN	WEEK	STATUS	HSA PR	MEDIA PR	MBA PR	PCBA PR	DB Line	…	ATR400
SN-002	WK01	Pass	Prime	Prime	Prime	RCY	DB_1	..	0
SN-003	WK01	Fail	Prime	RCY	Prime	Prime	DB_1	..	1
SN-004	WK01	Fail	RCY	RCY	Prime	Prime	DB_1	..	0
SN-005	WK01	Pass	Prime	Prime	Prime	Prime	DB_1	..	0
SN-011	WK02	Fail	Prime	RCY	RCY	RCY	DB_2	..	1
SN-012	WK02	Pass	Prime	Prime	Prime	RCY	DB_2	..	1
SN-019	WK03	Pass	Prime	Prime	RCY	Prime	DB_2	..	1
SN-020	WK03	Fail	RCY	Prime	Prime	Prime	DB_1	..	0

รูปที่ 3.12 ข้อมูลจำลองหลังจากที่ผ่านกระบวนการทำสมดุลข้อมูล

Drive SN	WEEK	STATUS	HSA PR	MEDIA PR	MBA PR	PCBA PR	DB Line	…	ATR400
SN-002	WK01	Pass	Prime	Prime	Prime	RCY	DB_1	..	0
SN-003	WK01	Fail	Prime	RCY	Prime	Prime	DB_1	..	1
SN-004	WK01	Fail	RCY	RCY	Prime	Prime	DB_1	..	0
SN-005	WK01	Pass	Prime	Prime	Prime	Prime	DB_1	..	0
SN-011	WK02	Fail	Prime	RCY	RCY	RCY	DB_2	..	1
SN-012	WK02	Pass	Prime	Prime	Prime	RCY	DB_2	..	1
SN-019	WK03	Pass	Prime	Prime	RCY	Prime	DB_2	..	1
SN-020	WK03	Fail	RCY	Prime	Prime	Prime	DB_1	..	0

Feature Selection

Drive SN	WEEK	STATUS	HSA PR	MEDIA PR
SN-002	WK01	Pass	Prime	Prime
SN-003	WK01	Fail	Prime	RCY
SN-004	WK01	Fail	RCY	RCY
SN-005	WK01	Pass	Prime	Prime
SN-011	WK02	Fail	Prime	RCY
SN-012	WK02	Pass	Prime	Prime
SN-019	WK03	Pass	Prime	Prime
SN-020	WK03	Fail	RCY	Prime

รูปที่ 3.13 ข้อมูลที่ผ่านกระบวนการทำสมดุลข้อมูลนำไปเพื่อคัดเลือกคุณลักษณะ

เมื่อได้ข้อมูลตามที่ได้แสดงไว้ในรูปที่ 3.12 แล้ว ข้อมูลทั้ง 8 แถวนี้ยังคงมีจำนวนคอลัมน์ทั้งหมด 400 คอลัมน์เช่นเดียวกับข้อมูลตั้งต้น ขั้นตอนต่อไปคือการคัดเลือกคุณลักษณะโดยจะทำ

การคัดเลือกคุณลักษณะให้เหลือเฉพาะคุณลักษณะที่สำคัญสำหรับการคาดการณ์ผลผลิต ในตัวอย่างนี้จะกำหนดให้เหลือเพียงสองคุณลักษณะ ในกรณีนี้คุณลักษณะที่ได้รับคัดเลือกมา ได้แก่ HSA PR และ Media PR ตามที่แสดงในรูปที่ 3.13

สิ่งสำคัญของการคัดเลือกคุณลักษณะนั้นไม่ใช่ชุดข้อมูลที่มีจำนวน 8 ยูนิต ในรูปที่ 3.13 แต่คือคุณลักษณะที่ส่งผลต่อการคาดการณ์ผลผลิต ซึ่งจะถูกนำไปใช้งานกับชุดข้อมูลดั้งเดิมที่มีจำนวนข้อมูลทั้งสิ้น 20 ยูนิต โดยจะมีผลลัพธ์อยู่ในรูปแบบของตารางที่จำนวนคอลัมน์ลดลงจาก 400 คอลัมน์เหลือเพียง 5 คอลัมน์ดังที่ได้แสดงไว้ในรูปที่ 3.14

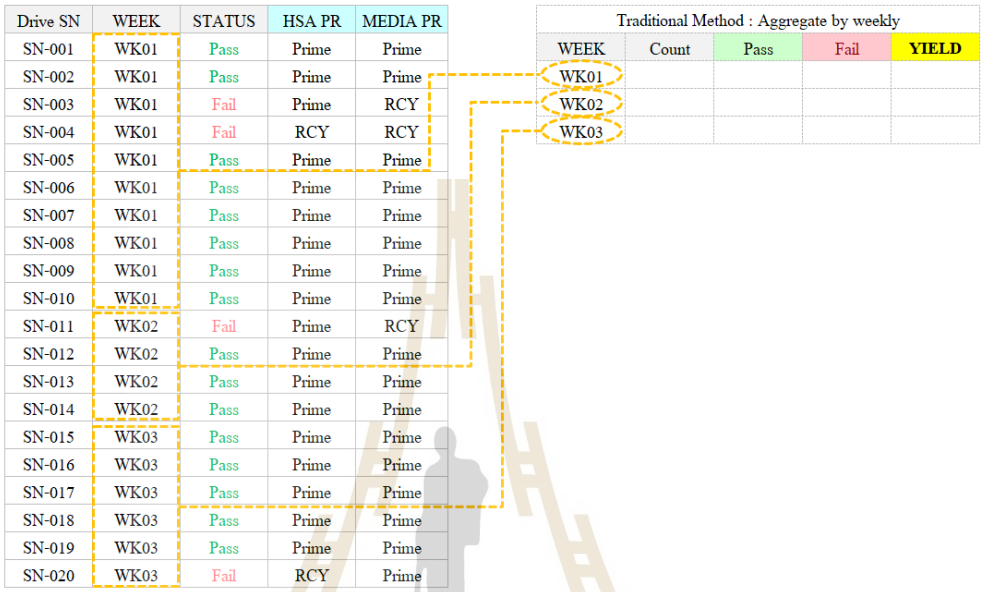
Drive SN	WEEK	STATUS	HSA PR	MEDIA PR
SN-001	WK01	Pass	Prime	Prime
SN-002	WK01	Pass	Prime	Prime
SN-003	WK01	Fail	Prime	RCY
SN-004	WK01	Fail	RCY	RCY
SN-005	WK01	Pass	Prime	Prime
SN-006	WK01	Pass	Prime	Prime
SN-007	WK01	Pass	Prime	Prime
SN-008	WK01	Pass	Prime	Prime
SN-009	WK01	Pass	Prime	Prime
SN-010	WK01	Pass	Prime	Prime
SN-011	WK02	Fail	Prime	RCY
SN-012	WK02	Pass	Prime	Prime
SN-013	WK02	Pass	Prime	Prime
SN-014	WK02	Pass	Prime	Prime
SN-015	WK03	Pass	Prime	Prime
SN-016	WK03	Pass	Prime	Prime
SN-017	WK03	Pass	Prime	Prime
SN-018	WK03	Pass	Prime	Prime
SN-019	WK03	Pass	Prime	Prime
SN-020	WK03	Fail	RCY	Prime

รูปที่ 3.14 ข้อมูลตั้งต้นที่ผ่านการคัดเลือกคุณลักษณะเรียบร้อยแล้ว

ขั้นตอนสุดท้ายจะเป็นการรวมกลุ่มข้อมูล โดยตัวอย่างการอธิบายนั้นจะอ้างอิงจากวิธีการดั้งเดิมซึ่งเป็นการรวมกลุ่มข้อมูลตามรายสัปดาห์ มีขั้นตอนย่อยในรายละเอียดดังต่อไปนี้

1. หาจำนวนสัปดาห์ของข้อมูลชุดนี้ ซึ่งในชุดข้อมูลนี้มีค่าเท่ากับ 3 ได้แก่ WK01, WK02 และ WK03 ดังนั้นชุดข้อมูลใหม่ที่เกิดจากการรวมกลุ่มชุดข้อมูลเดิมนั้นจะมีจำนวน 3 แถว ดังที่ได้แสดงการสร้างชุดข้อมูลใหม่ไว้ในรูปที่ 3.15

2. สร้างคอลัมน์ใหม่เพื่อแสดงผลลัพธ์ของจำนวนข้อมูลในแต่ละสัปดาห์ โดยในตัวอย่างนี้กำหนดให้คอลัมน์ใหม่ชื่อว่า Count ซึ่งในแต่ละแถวจะมีจำนวนข้อมูลเดิมอยู่ 10, 4 และ 6 ตามลำดับตามแผนภาพที่ได้แสดงไว้ในรูปที่ 3.16

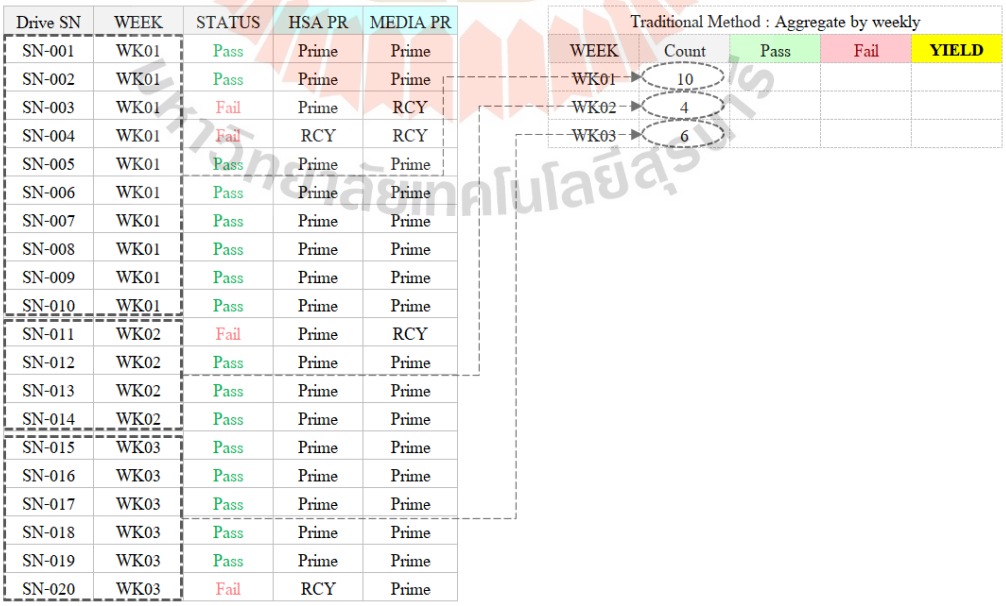


Drive SN	WEEK	STATUS	HSA PR	MEDIA PR
SN-001	WK01	Pass	Prime	Prime
SN-002	WK01	Pass	Prime	Prime
SN-003	WK01	Fail	Prime	RCY
SN-004	WK01	Fail	RCY	RCY
SN-005	WK01	Pass	Prime	Prime
SN-006	WK01	Pass	Prime	Prime
SN-007	WK01	Pass	Prime	Prime
SN-008	WK01	Pass	Prime	Prime
SN-009	WK01	Pass	Prime	Prime
SN-010	WK01	Pass	Prime	Prime
SN-011	WK02	Fail	Prime	RCY
SN-012	WK02	Pass	Prime	Prime
SN-013	WK02	Pass	Prime	Prime
SN-014	WK02	Pass	Prime	Prime
SN-015	WK03	Pass	Prime	Prime
SN-016	WK03	Pass	Prime	Prime
SN-017	WK03	Pass	Prime	Prime
SN-018	WK03	Pass	Prime	Prime
SN-019	WK03	Pass	Prime	Prime
SN-020	WK03	Fail	RCY	Prime

Traditional Method : Aggregate by weekly

WEEK	Count	Pass	Fail	YIELD
WK01				
WK02				
WK03				

รูปที่ 3.15 ข้อมูลในรูปแบบดั้งเดิมถูกนำไปสร้างชุดข้อมูลใหม่ที่มีการรวมกลุ่มข้อมูลตามสัปดาห์



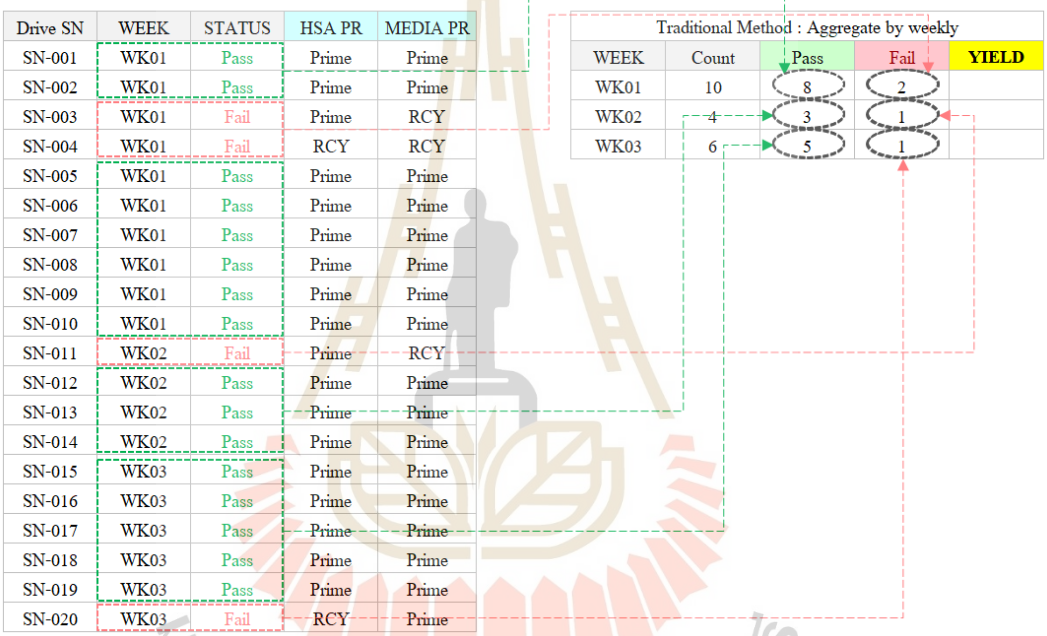
Drive SN	WEEK	STATUS	HSA PR	MEDIA PR
SN-001	WK01	Pass	Prime	Prime
SN-002	WK01	Pass	Prime	Prime
SN-003	WK01	Fail	Prime	RCY
SN-004	WK01	Fail	RCY	RCY
SN-005	WK01	Pass	Prime	Prime
SN-006	WK01	Pass	Prime	Prime
SN-007	WK01	Pass	Prime	Prime
SN-008	WK01	Pass	Prime	Prime
SN-009	WK01	Pass	Prime	Prime
SN-010	WK01	Pass	Prime	Prime
SN-011	WK02	Fail	Prime	RCY
SN-012	WK02	Pass	Prime	Prime
SN-013	WK02	Pass	Prime	Prime
SN-014	WK02	Pass	Prime	Prime
SN-015	WK03	Pass	Prime	Prime
SN-016	WK03	Pass	Prime	Prime
SN-017	WK03	Pass	Prime	Prime
SN-018	WK03	Pass	Prime	Prime
SN-019	WK03	Pass	Prime	Prime
SN-020	WK03	Fail	RCY	Prime

Traditional Method : Aggregate by weekly

WEEK	Count	Pass	Fail	YIELD
WK01	10			
WK02	4			
WK03	6			

รูปที่ 3.16 การสร้างคอลัมน์ Count ในชุดข้อมูลใหม่ที่มีการรวมกลุ่มข้อมูลตามสัปดาห์

3. สร้างคอลัมน์ใหม่เพื่อแสดงผลลัพธ์ของจำนวนยูนิตที่ Pass และ Fail ของแต่ละสัปดาห์ โดย WK01 จำนวนข้อมูลที่ Pass = 8 และ Fail = 2 จากนั้นจึงทำซ้ำใน WK02 และ WK03 ดังที่ได้แสดงไว้ในรูปที่ 3.17 และในขั้นตอนนี้จะรวมไปถึงการคำนวนผลผลิตในแต่ละสัปดาห์ ซึ่งจะเป็นไปตามสมการที่ได้กล่าวไว้ในบทที่ 2 โดยยีลด์คำนวณได้จากการนำจำนวนยูนิต Pass หารด้วยจำนวนยูนิตที่เข้าสู่กระบวนการทดสอบผลิตภัณฑ์ ข้อมูลจำลองชุดนี้จะได้ยีลด์ ของ WK01 เท่ากับ 80 เปอร์เซ็นต์ ยีลด์ของ WK02 เท่ากับ 75 เปอร์เซ็นต์ และยีลด์ของ WK03 เท่ากับ 83 เปอร์เซ็นต์ โดยผลลัพธ์การคำนวณยีลด์ได้ถูกแสดงไว้ในตารางที่ 3.2



Drive SN	WEEK	STATUS	HSA PR	MEDIA PR
SN-001	WK01	Pass	Prime	Prime
SN-002	WK01	Pass	Prime	Prime
SN-003	WK01	Fail	Prime	RCY
SN-004	WK01	Fail	RCY	RCY
SN-005	WK01	Pass	Prime	Prime
SN-006	WK01	Pass	Prime	Prime
SN-007	WK01	Pass	Prime	Prime
SN-008	WK01	Pass	Prime	Prime
SN-009	WK01	Pass	Prime	Prime
SN-010	WK01	Pass	Prime	Prime
SN-011	WK02	Fail	Prime	RCY
SN-012	WK02	Pass	Prime	Prime
SN-013	WK02	Pass	Prime	Prime
SN-014	WK02	Pass	Prime	Prime
SN-015	WK03	Pass	Prime	Prime
SN-016	WK03	Pass	Prime	Prime
SN-017	WK03	Pass	Prime	Prime
SN-018	WK03	Pass	Prime	Prime
SN-019	WK03	Pass	Prime	Prime
SN-020	WK03	Fail	RCY	Prime

Traditional Method : Aggregate by weekly				
WEEK	Count	Pass	Fail	YIELD
WK01	10	8	2	
WK02	4	3	1	
WK03	6	5	1	

รูปที่ 3.17 การสร้างคอลัมน์ใหม่ขึ้นมาสองคอลัมน์เพื่อแสดงจำนวน Pass และ Fail

ตารางที่ 3.2 แสดงผลลัพธ์การคำนวณผลผลิต (Yield Calculation) ในแต่ละสัปดาห์

Week	Count	Pass	Fail	Yield (%)
WK01	10	8	2	8/10*100 = 80.00
WK02	4	3	1	3/4*100 = 75.00
WK03	6	5	1	5/6*100 = 83.33

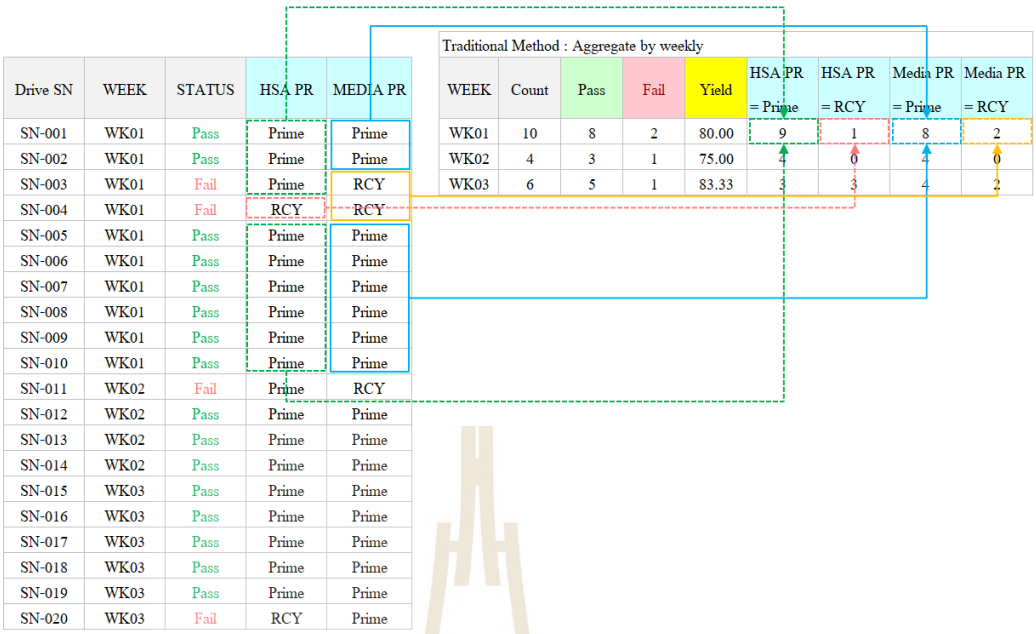
4. สร้างคอลัมน์ใหม่เพื่อแสดงผลลัพธ์ของจำนวนยูนิตของแต่ละสัปดาห์ ตามจำนวนค่าความเป็นไปได้ของคุณลักษณะที่ได้คัดเลือกมาแล้ว ณ ที่นี้ ได้แก่ HSA PR ที่มีค่าความเป็นไปได้ 2 ค่าคือ Prime และ RCY ส่วน Media PR ก็มีค่าความเป็นไปได้ 2 ค่าคือ Prime และ RCY เช่นกัน ดังนั้นจำนวนคอลัมน์ที่ถูกสร้างขึ้นมาใหม่จากขั้นตอนนี้มีทั้งหมด 4 คอลัมน์ ได้แก่ HSA PR = Prime, HSA PR = RCY, Media PR = Prime และ Media PR = RCY ซึ่งได้แสดงไว้ในตารางที่ 3.3

ตารางที่ 3.3 แสดงการเพิ่มคอลัมน์ใหม่จำนวน 4 คอลัมน์เพื่อเตรียมพร้อมสำหรับการนำไปใช้งานในขั้นตอนการคาดการณ์ผลผลิต

Week	Count	Pass	Fail	Yield (%)	HSA PR = Prime	HSA PR = RCY	Media PR = Prime	Media PR = RCY
WK01	10	8	2	80.00				
WK02	4	3	1	75.00				
WK03	6	5	1	83.33				

5. การนับจำนวนข้อมูลตามแต่ละเงื่อนไขและนำไปเติมในแต่ละช่องข้อมูล เริ่มจาก WK01 มี HSA PR = Prime จำนวน 9 ยูนิต มี HSA PR = RCY จำนวน 1 ยูนิต มี Media PR = Prime จำนวน 8 ยูนิต และมี Media PR = RCY จำนวน 2 ยูนิต ซึ่งจะสามารถนำข้อมูลเติมลงไปในทั้ง 4 คอลัมน์ที่สร้างขึ้นมาใหม่ได้จนครบ จากนั้นจึงทำซ้ำเช่นเดียวกันใน WK02 และ WK03 จนครบทั้งหมด โดยรูปที่ 3.18 เป็นการแสดงให้เห็นถึงการเชื่อมโยงการนับข้อมูลรวมไปถึงตารางผลลัพธ์

แต่ว่าการนับจำนวนของข้อมูลแต่ละสัปดาห์เติมลงไปตารางที่สร้างขึ้นมาใหม่นั้นไม่เป็นที่นิยมเนื่องจากไม่สามารถนำไปใช้ในการคาดการณ์ผลผลิตได้ ด้วยเหตุผลที่ว่าจำนวนข้อมูลของแต่ละสัปดาห์นั้นจะมีจำนวนไม่เท่ากัน ดังนั้นจึงได้มีการทำให้เป็นค่าเปอร์เซ็นต์อัตราส่วนของจำนวนยูนิตทั้งหมดเสียก่อน (Rationalization) โดยข้อมูลที่ได้รับการแปลงได้ถูกแสดงไว้ในตารางที่ 3.4



Drive SN	WEEK	STATUS	HSA PR	MEDIA PR
SN-001	WK01	Pass	Prime	Prime
SN-002	WK01	Pass	Prime	Prime
SN-003	WK01	Fail	Prime	RCY
SN-004	WK01	Fail	RCY	RCY
SN-005	WK01	Pass	Prime	Prime
SN-006	WK01	Pass	Prime	Prime
SN-007	WK01	Pass	Prime	Prime
SN-008	WK01	Pass	Prime	Prime
SN-009	WK01	Pass	Prime	Prime
SN-010	WK01	Pass	Prime	Prime
SN-011	WK02	Fail	Prime	RCY
SN-012	WK02	Pass	Prime	Prime
SN-013	WK02	Pass	Prime	Prime
SN-014	WK02	Pass	Prime	Prime
SN-015	WK03	Pass	Prime	Prime
SN-016	WK03	Pass	Prime	Prime
SN-017	WK03	Pass	Prime	Prime
SN-018	WK03	Pass	Prime	Prime
SN-019	WK03	Pass	Prime	Prime
SN-020	WK03	Fail	RCY	Prime

Traditional Method : Aggregate by weekly

WEEK	Count	Pass	Fail	Yield	HSA PR = Prime	HSA PR = RCY	Media PR = Prime	Media PR = RCY
WK01	10	8	2	80.00	9	1	8	2
WK02	4	3	1	75.00	4	0	4	0
WK03	6	5	1	83.33	3	3	4	2

รูปที่ 3.18 การเติมข้อมูลลงในคอลัมน์ใหม่มาเพื่อแสดงจำนวนของฮาร์ดดิสก์ไดรฟ์ของแต่ละเงื่อนไข

ตารางที่ 3.4 แสดงผลลัพธ์การเติมข้อมูลลงในคอลัมน์ใหม่ของแต่ละเงื่อนไข รวมการคำนวณเพื่อแปลงข้อมูลเป็นค่าที่ Rationalization

Week	**Count**	**Pass**	**Fail**	**Yield (%)**	**HSA PR = Prime Ratio%**	**HSA PR = RCY Ratio%**	**Media PR = Prime Ratio%**	**Media PR = RCY Ratio%**
WK01	10	8	2	80.00	9/10*100 = 90	1/10 *100 = 10	8/10*100 = 80	2/10*100 = 20
WK02	4	3	1	75.00	4/4*100 = 100	0/4 *100 = 0	4/4*100 = 100	0/4*100 = 0
WK03	6	5	1	83.33	3/6*100 = 50	3/6*100 = 50	4/6*100 = 66.67	2/6*100 = 33.33

หลังจากที่กระบวนการทุกขั้นตอนเสร็จสิ้นจะได้ตารางที่จะนำไปทำการคาดการณ์ผลผลิตได้ตามตารางที่ 3.5 โดย HSA Prime Ratio, HSA RCY ratio, Media Prime Ratio และ Media RCY Ratio จะถูกนำไปใช้เป็นคุณลักษณะเพื่อสร้างโมเดลการคาดการณ์ผลผลิตซึ่งนั่นก็คือข้อมูลในคอลัมน์ Yield นั่นเอง

ตารางที่ 3.5 แสดงข้อมูลที่ Rationalization ซึ่งพร้อมสำหรับการใช้งานในการคาดการณ์ผลผลิต

Week	Count	Pass	Fail	Yield (%)	HSA Prime Ratio%	HSA RCY Ratio%	Media Prime Ratio%	Media RCY Ratio%
WK01	10	8	2	80.00	90	10	80	20
WK02	4	3	1	75.00	100	0	100	0
WK03	6	5	1	83.33	50	50	66.67	33.33

3.4 เครื่องมือที่ใช้สำหรับการวิจัย

เครื่องมือที่ใช้ในการพัฒนางานวิจัยนี้ ประกอบด้วย

1. เครื่องคอมพิวเตอร์สำหรับพัฒนา มีรายละเอียดดังนี้

   หน่วยประมวลผลกลาง : Intel(R) Core(TM) i5-7300HQ CPU @ 2.5GHz

   หน่วยประมวลผลกราฟิก : NVIDIA GeForce GTX 1050

   หน่วยความจำหลัก : 8 GB

   หน่วยความจำสำรอง HDD/SSD : 1 TB/250GB

2. ระบบปฏิบัติการและโปรแกรมประยุกต์สำหรับพัฒนา ประกอบด้วย

   ระบบปฏิบัติการ : Windows 10 (64-bits Operating System)

   เครื่องมือที่ใช้ในการพัฒนา : IBM SPSS Modeler Version, Microsoft Excel

3. ภาษาที่ใช้ในการพัฒนาและการประมวลผลบนระบบคลาวด์ มีรายละเอียดดังนี้

   ผู้ให้บริการ : Kaggle

   ภาษาที่ใช้ในการพัฒนา : ภาษา R

   หน่วยประมวลผลกลาง : Intel(R) Xeon(R) CPU @ 2.30GHz

   หน่วยความจำหลัก : 16 GB

   หน่วยความจำสำรอง : 20 GB

   หน่วยประมวลผลกราฟิก : NVIDIA P100 @1.32GHz

บทที่ 4
ผลการศึกษาและการวิเคราะห์ผล

ในบทนี้จะกล่าวถึงผลลัพธ์ของการดำเนินงานในแต่ละขั้นตอนของงานวิจัยชิ้นนี้ โดยรวมไปถึงการวิเคราะห์ผลการทดลองเหล่านั้นอีกด้วย

4.1 ผลการดำเนินงานวิจัยในส่วนของการจัดการและทำสมดุลข้อมูล

ข้อมูลดิบที่นำมาใช้ในงานวิจัยชิ้นนี้มีจำนวนมากถึง 10,000,000 แถว และนอกเหนือจากนั้นแล้วคุณลักษณะที่เป็นคลาสเป้าหมายนั่นคือ สถานะของการทดสอบผลิตภัณฑ์ฮาร์ดดิสก์ไดรฟ์ (Status Pass/Fail) นั้นมีความไม่สมดุลเป็นอย่างมาก เนื่องจากยีลด์ของกระบวนการทดสอบผลิตภัณฑ์นั้นค่อนข้างสูง จึงทำให้จำนวนของข้อมูลที่ค่าคุณลักษณะ STATUS = FAIL มีจำนวนน้อย ซึ่งทางผู้ดำเนินงานวิจัยได้ทำการปรับสมดุลข้อมูลโดยใช้ อัลกอริทึม k-Means Clustering ในการจัดกลุ่มข้อมูลแล้วจึงทำสมดุลข้อมูลด้วยอัลกอริทึม k-NN ผนวกกับการทำ Re-sampling (Data Balancing by k-Means Clustering k-NN and Re-sampling : DBC-2KAR) โดยมีผลลัพธ์ที่ได้ดังนี้

ตารางที่ 4.1 แสดงข้อมูลที่ถูกแบ่งออกเป็น 5 คลัสเตอร์ตามอัลกอริทึม k-Means Clustering โดยข้อมูลของยูนิตที่ Pass จะมีจำนวนมากกว่า ยูนิตที่ Fail ทั้งห้าคลัสเตอร์ และแสดงให้เห็นถึงค่า Imbalance Ratio ของแต่ละคลัสเตอร์อีกด้วย โดยคลัสเตอร์ที่มีค่า Imbalance Ratio น้อยที่สุดอยู่ที่ 27.17 : 1 และมากที่สุดอยู่ที่ 29.00 : 1

ตารางที่ 4.1 แสดงค่าระดับความไม่สมดุลของข้อมูลของแต่ละคลัสเตอร์

k-Means Clustering	Passer	Failure	Imbalanced Ratio
Cluster#1	2,064,114	71,181	29.00 : 1
Cluster#2	1,981,558	70,550	28.09 : 1
Cluster#3	1,850,018	65,561	28.22 : 1
Cluster#4	1,924,591	70,846	27.17 : 1
Cluster#5	1,834,891	66,690	27.51 : 1
TOTAL	9,655,172	344,828	28.00 : 1

หลังจากทำการแบ่งข้อมูลออกเป็น 5 คลัสเตอร์เรียบร้อยแล้วจึงทำการปรับสมดุลข้อมูล โดยตารางที่ 4.2 แสดงให้เห็นว่าทุกคลัสเตอร์ได้ถูกปรับสมดุลจน Imbalance Ratio มีค่าเป็น 1:1 ซึ่งการปรับสมดุลของข้อมูลนี้จะส่งผลดีต่อการจำแนกข้อมูลในกระบวนการคัดเลือกคุณลักษณะในขั้นตอนถัดไป

ตารางที่ 4.2 แสดงข้อมูลของแต่ละคลัสเตอร์หลังจากทำสมดุลข้อมูลเรียบร้อยแล้ว

After Balanced	Passer	Failure	Imbalance Ratio
Cluster#1	71,181	71,181	1 : 1
Cluster#2	70,550	70,550	1 : 1
Cluster#3	65,561	65,561	1 : 1
Cluster#4	70,846	70,846	1 : 1
Cluster#5	66,690	66,690	1 : 1
TOTAL	344,828	344,828	1 : 1

4.2 ผลการดำเนินงานวิจัยในส่วนของการคัดเลือกคุณลักษณะ

ขั้นตอนในการคัดเลือกคุณลักษณะเพื่อนำไปใช้งานต่อในขั้นตอนการคาดการณ์ผลผลิตของกระบวนการนั้น จะเป็นการนำชุดข้อมูลที่ได้จากการปรับสมดุลข้อมูลเรียบร้อยแล้วจากกระบวนการก่อนหน้า เพื่อเข้าสู่กระบวนการคัดเลือกคุณลักษณะที่ส่งผลต่อการคำนวณผลผลิตของกระบวนการได้แม่นยำมากที่สุด โดยวิธีการที่นำมาใช้ในการสร้างโมเดลมีทั้งหมด 7 ชนิด ได้แก่ ต้นไม้ตัดสินใจ C5, CART, Support Vectors Machine, Stepwise Regression, Genetic Algorithm, Chi-square และ Information Gain โดยที่ 5 ชนิดแรกนั้นจะเป็นการสร้างโมเดลการเรียนรู้เพื่อคัดเลือกคุณลักษณะ และ 2 ชนิดหลังนั้นใช้หลักการทางสถิติเพื่อให้ค่าน้ำหนักและเรียงลำดับคุณลักษณะตามความสำคัญที่ส่งผลต่อการการจำแนกคุณลักษณะของคลาสเป้าหมาย โดย 8-10 อันดับแรกของคุณลักษณะที่ส่งผลต่อการจำแนกสถานะ “Pass” และ “Fail” มากที่สุดของแต่ละวิธีการจะถูกนำไปใช้งานในการคาดการณ์ผลผลิตของกระบวนการทดสอบผลิตภัณฑ์ซึ่งเป็นขั้นตอนถัดไป โดยคุณลักษณะของแต่ละวิธีการที่ได้ถูกคัดเลือกมาแล้วนั้นแสดงในตารางที่ 4.3 นอกเหนือจากคุณลักษณะที่ได้รับมาจากการคัดเลือกคุณลักษณะจากทั้ง 7 ชนิด ตารางนี้ได้แสดงถึง 5 คุณลักษณะหลักที่วิศวกรผู้ชำนาญการ (Human Expert)ได้ใช้ในการคาดการณ์ผลผลิตของกระบวนการในการทำงานจริง

ตารางที่ 4.3 แสดงคุณลักษณะที่ถูกคัดเลือกมาจากแต่ละวิธีการ

Human Expert	C5	CART	SVM
HGSA_PRIME	HSA_VENDOR	PWALINE	PWALINE
MEDIA_PRIME	MEDIA_PRIME	CUST_SCORE	CUST_SCORE
VCM_PRIME	PWALINE	DISC_INSTALL_ID	DISC_INSTALL_ID
MBA_PRIME	CELL_ID	DRIVE_PART_NUM	SWG_CELL_ID
PCBA_PRIME	CUST_SCORE	PRIME	DRV_COMP_TRK
	HGSA_PRIME	CMS_CONFIG	DRV_COMP_TRK
	CRX_CNT	BUILD G	HSA_VENDOR
	HSA_COH	MBA_PN	HSA_VENDOR
	DL_IMG		SPUTTER_WC

ตารางที่ 4.3 แสดงคุณลักษณะที่ถูกคัดเลือกมาจากแต่ละวิธีการ (ต่อ)

Stepwise Regression	Genetic Algorithm	Chi-Square	Information Gain
HSA_VENDOR	MEDIA_PRIME	HSA_VENDOR	HSA_VENDOR
CUST_SCORE	PRIME	HSA_RWK	HSA_RWK
HSA_COH	LVCM	DRV_2	DRV_2
DL_IMG	UVCM	DRV_COMP_TRK	DRV_COMP_TRK
DRV_COMP_TRK	DRV_5	MEDIA_PRIME	MEDIA_PRIME
BUILD G	HSA_RWK	PRIME	PRIME
CMS_CONFIG	LINE	LVCM	LVCM
DRIVE_PART_NUM	HSA_COH	UVCM	UVCM
PCBA_CNT	HSA_PN	MEDIA_RCY	MEDIA_RCY
	DRV_3	DRV_5	DRV_5

ตารางที่ 4.4 แสดงการเปรียบเทียบค่าเวลาที่ใช้ในการประมวลผลของ 7 ชนิด

วิธีการการคัดเลือกคุณลักษณะ	เวลาที่ใช้ในการประมวลผล (Second)
C5	6,791
CART	318
SVM	12,768
Stepwise Regression	566
Genetic Algorithm	50,280
Chi-Square	10
Information Gain	10

ตารางที่ 4.4 แสดงถึงการเปรียบเทียบประสิทธิภาพในแง่ของเวลาที่ใช้ในการคำนวณ (Computation Time) ซึ่งในแง่นี้ Chi-Square และ Information Gain เป็นสองวิธีการที่มีความโดดเด่น ซึ่งใช้เวลาเพียงแค่ 10 วินาที ในขณะที่ CART และ Stepwise Regression ใช้เวลาในการประมวลผล 318 วินาที และ 566 วินาทีตามลำดับ ส่วนอัลกอริทึม C5 ใช้เวลา 6,791 วินาที (หรือประมาณ 1 ชั่วโมง 52 นาที) SVM ใช้เวลา 12,768 วินาที (หรือประมาณ 3 ชั่วโมง 32 นาที) และ Genetic Algorithm เป็นวิธีการคัดเลือกคุณลักษณะที่ใช้เวลาในการคำนวณและสร้างโมเดลการเรียนรู้ที่นานที่สุดโดยใช้เวลา 50,280 วินาที (หรือประมาณ 14 ชั่วโมง 7 นาที)

4.3 ผลการดำเนินงานวิจัยในการสร้างโมเดลเพื่อการคาดการณ์ผลผลิต

สำหรับในขั้นตอนการสร้างโมเดลเพื่อการคาดการณ์ผลผลิตนั้น งานวิจัยนี้ได้ใช้อัลกอริทึมการเรียนรู้ 2 ชนิด ได้แก่ MLR (Multiple Linear Regression) และ ANN (Artificial Neural Network) ซึ่งตัววัดประสิทธิภาพในการทดลองนี้มี 2 ชนิด ได้แก่ RMSE (Root Means Square Error) และ MAE (Mean Absolute Error) โดยคุณลักษณะทั้งหมดที่ได้รับจากกระบวนการคัดเลือกคุณลักษณะด้วยวิธีการทั้ง 7 ชนิด ได้ถูกนำมาใช้ในการสร้างโมเดลการเรียนรู้เพื่อการคาดการณ์ผลผลิต รวมไปถึงการคาดการณ์ผลผลิตด้วยวิธีการดั้งเดิมจากคุณลักษณะทั้ง 5 ที่ได้รับจากประสบการณ์ของวิศวกรผู้ชำนาญการได้ถูกนำมาเปรียบเทียบประสิทธิภาพด้วยเช่นกัน

ในการคาดการณ์ผลผลิตของการทดสอบกระบวนการหรือการคาดการณ์ผลผลิตนั้น จะเริ่มจากการคำนวณยีลด์ ตามที่ได้กล่าวไว้ในข้างต้น (ในหัวข้อที่ 3.2.3) งานวิจัยนี้ได้นำเสนอแนวคิดใหม่ จากวิธีการดั้งเดิมที่จะใช้การรวมกลุ่มข้อมูลกันตามระยะเวลาของปฏิทินซึ่งจะมีความไม่

สม่ำเสมอของข้อมูลในระดับสูง ซึ่งแนวคิดใหม่นี้จะเป็นการรวมกลุ่มข้อมูลด้วยขนาดจำนวนข้อมูลที่คงที่ โดยใช้ค่าคงที่ของจำนวนข้อมูลที่ 10,000 แถว

ตารางที่ 4.5 แสดงจำนวนของข้อมูลของวิธีการดั้งเดิมและวิธีการที่นำเสนอ

	วิธีการดั้งเดิม	**วิธีการที่นำเสนอ**
จำนวนข้อมูลทั้งหมด	10,000,000	10,000,000
จำนวนแถวของข้อมูลที่ได้นำไปใช้ในการทดลอง	157	1,000
ค่ามากที่สุดของจำนวนข้อมูลในหนึ่งแถว	315,717	10,000
ค่าน้อยที่สุดของจำนวนข้อมูลในหนึ่งแถว	0	10,000

จากตารางที่ 4.5 แสดงให้เห็นว่าด้วยชุดข้อมูลที่มีการรวมกลุ่มข้อมูลด้วยวิธีการดั้งเดิมนั้นจะได้จำนวน 157 แถว ซึ่งในกรณีนี้เป็นการรวมกลุ่มข้อมูลตามสัปดาห์การผลิตฮาร์ดดิสก์ไดรฟ์ โดยจำนวนข้อมูลมากที่สุดในหนึ่งแถวนั้นสูงถึง 315,717 ข้อมูลนั้นสามารถแปลความหมายได้ว่าในสัปดาห์นั้นมีการผลิตฮาร์ดดิสก์ไดรฟ์ออกมาเป็นจำนวน 315,717 ยูนิต และจำนวนที่น้อยที่สุดคือ 0 ข้อมูลต่อหนึ่งแถว ซึ่งหมายถึงไม่มีการผลิตฮาร์ดดิสก์ไดรฟ์ออกมาเลย ในขณะที่วิธีการใหม่ที่รวมกลุ่มข้อมูลด้วยวิธีการที่นำเสนอนั้นจะมีจำนวนข้อมูลต่อแถวคงที่เป็นจำนวน 10,000 ข้อมูลต่อแถว และนั่นทำให้ในการทดลองนี้จะมีจำนวนแถว 1,000 แถวเพื่อใช้ในการสร้างโมเดลการเรียนรู้เพื่อฝึกสอนและทดสอบ

ข้อมูลที่ได้รับการปรับปรุงการรวมกลุ่มข้อมูลด้วยจำนวนข้อมูลต่อแถวที่คงที่ ถูกนำไปสร้างโมเดลคาดการณ์ผลผลิตด้วยอัลกอริทึม MLR และ ANN ผลการประเมินประสิทธิภาพโมเดลสรุปได้ดังตารางที่ 4.6

ตารางที่ 4.6 แสดงการเปรียบเทียบประสิทธิภาพการคาดการณ์ผลผลิตจากโมเดล MLR และ ANN โดยใช้คุณลักษณะที่ได้รับจากการคัดเลือกคุณลักษณะทั้ง 7 ชนิด

Feature Selection Method	Yield Prediction Algorithm	Test Data	
		RMSE	MAE
Human Engineers	Traditional	1.7000	1.4000
C5	MLR	0.8660	0.6050
	ANN	1.7070	1.2630
CART	MLR	24.1050	5.9130
	ANN	1.6300	1.2510
SVM	MLR	2.0370	1.2470
	ANN	1.8640	1.3840
Stepwise Regression	MLR	10.3260	2.8420
	ANN	1.8510	1.3060
Genetic Algorithm	MLR	0.7320	0.5590
	ANN	1.7060	1.2690
Chi-Square	MLR	0.8210	0.6900
	ANN	1.7070	1.2620
Information Gain	MLR	0.8210	0.6900
	ANN	1.7070	1.2620

4.3.1 ผลการทดลองด้วยตัววัดประสิทธิภาพ RMSE

จากข้อมูลในตารางที่ 4.6 เมื่อประเมินด้วยตัววัดประสิทธิภาพ RMSE นั้น วิธีการคัดเลือกคุณลักษณะที่เป็นตัวแทนของกลุ่มวิธีการทางสถิติได้แก่ Chi-Square และ Information Gain เมื่อทำงานร่วมกับการใช้ ANN ในการเป็นอัลกอริทึมการเรียนรู้ในการสร้างโมเดลการคาดการณ์ผลผลิต นั้นจะมีประสิทธิภาพในการคาดการณ์ผลผลิตที่ไม่แตกต่างจากวิธีการดั้งเดิมมากนัก โดยค่าของ RMSE อยู่ที่ 1.707 ในขณะที่วิธีการดั้งเดิมนั้นมีค่า RMSE เท่ากับ 1.700 ในส่วนของวิธีการคัดเลือกคุณลักษณะด้วย C5, SVM, Stepwise Regression และ Genetic Algorithm นั้นเมื่อทำงานร่วมกับการสร้างโมเดลด้วยอัลกอริทึม ANN จะได้ผลลัพธ์เป็นค่า RMSE ที่แย่กว่าเดิมเล็กน้อย ในอัลกอริทึม SVM และ Stepwise Regression ให้ค่าอยู่ที่ 1.864 และ 1.851 ตามลำดับ ส่วนอัลกอริทึม

C5 และ GA ถือว่าผลลัพธ์ไม่ต่างจากเดิมมากนัก ให้ค่าอยู่ที่ 1.707 และ 1.706 ตามลำดับ ซึ่งจะมีเพียงวิธีการคัดเลือกคุณลักษณะด้วยอัลกอริทึม CART ที่จะให้ค่า RMSE ที่ดีกว่าวิธีการดั้งเดิมอย่างชัดเจนโดยค่าที่ได้เท่ากับ 1.630

ในขณะที่การใช้ MLR เพื่อเป็นอัลกอริทึมการเรียนรู้ในการสร้างโมเดลนั้นจะให้ผลลัพธ์ที่ดีกว่าในการคัดเลือกคุณลักษณะด้วยวิธีการ CART, C5, Chi-square, Information Gain และ Genetic Algorithm โดยค่าที่ได้นั้นคือ 1.630, 0.866, 0.821, 0.821 และ 0.732 ซึ่งค่า 0.732 ของ Genetic Algorithm ที่ทำงานร่วมกับ MLR เป็นค่า RMSE ที่ดีที่สุดในการทดลองนี้

4.3.2 ผลการทดลองด้วยตัววัดประสิทธิภาพ MAE

ในส่วนของการทดสอบวัดประสิทธิภาพด้วย MAE นั้น วิธีการคัดเลือกคุณลักษณะทั้ง 7 วิธี เมื่อทำงานร่วมกับโมเดลการคาดการณ์ผลผลิตที่สร้างขึ้นมาด้วยอัลกอริทึม ANN จะให้ผลลัพธ์ที่ดีกว่าวิธีการดั้งเดิมทั้งหมด ซึ่งค่า MAE ที่ได้จากการทดลองในข้อมูลชุดนี้จะอยู่ระหว่าง 1.251 ถึง 1.384 ในขณะที่ค่า MAE ที่ได้จากการคาดการณ์ผลผลิตด้วยวิธีการดั้งเดิมนั้นอยู่ที่ 1.400

การสร้างโมเดลคาดการณ์ผลผลิตด้วยอัลกอริทึม MLR นั้นให้ผลลัพธ์ค่า MAE ที่แตกต่างกันออกไปในแต่และวิธีการคัดเลือกคุณลักษณะ วิธีการ CART และ Stepwise Regression เป็น 2 วิธีที่ให้ค่า MAE ที่แย่กว่าวิธีการดั้งเดิม โดยจะมีค่า 5.913 และ 2.842 วิธีการคัดเลือกด้วย SVM มีค่าเท่ากับ 1.247 ซึ่งจะดีกว่าเล็กน้อย ในส่วนของวิธีการคัดเลือกคุณลักษณะด้วย Chi-Square, Information Gain, C5 และ Genetic Algorithm นั้นจะมีประสิทธิภาพในการคาดการณ์ผลผลิตที่ดีกว่าวิธีการดั้งเดิมอย่างเห็นได้ชัด โดยจะมีค่าเท่ากับ 0.690, 0.690, 0.605 และ 0.559

นอกเหนือจากการแสดงผลลัพธ์ด้วยค่า RMSE และ MAE แล้ว ในตารางที่ 4.6 ได้แสดงให้เห็นถึงการเปรียบเทียบประสิทธิภาพของแต่ละวิธีการโดยใช้การคำนวณอัตราส่วนร้อยละของค่าผิดพลาดที่ลดลง (Percent of Error Reduction หรือ Error Reduction Rate) เมื่อเทียบกับค่า RMSE และ MAE วิธีการดั้งเดิม สมการที่ 4.1 แสดงถึงการคำนวน Error Reduction Rate ของ RMSE และสมการที่ 4.2 แสดงถึงการคำนวน Error Reduction Rate ของ RMSE

$$Error\ Reduction\ Rate_{(RMSE)} = \frac{RMSE_{(Proposed\ method)} - RMSE_{(Traditional\ method)}}{RMSE_{(Traditional\ method)}} \quad (4.1)$$

$$Error\ Reduction\ Rate_{(MAE)} = \frac{MAE_{(Proposed\ method)} - MAE_{(Traditional\ method)}}{MAE_{(Traditional\ method)}} \quad (4.2)$$

ตารางที่ 4.7 แสดงค่า Error Reduction ของวิธีการทั้ง 7 เมื่อเปรียบเทียบกับวิธีการดั้งเดิม

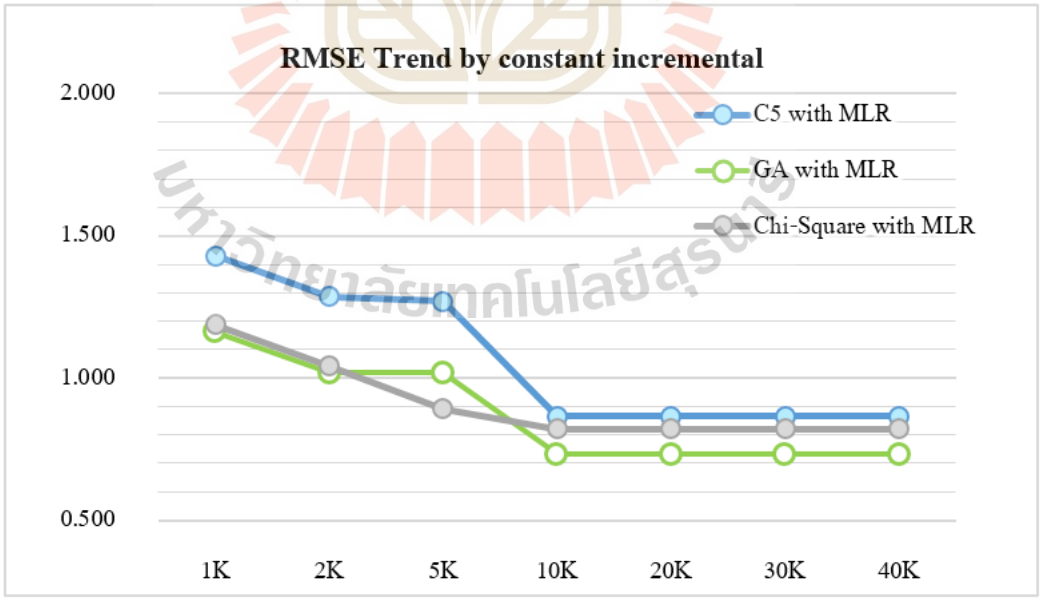
Feature Selection and Model Building Scheme	**Error Reduction**	
	RMSE	**MAE**
C5 with MLR **	**-49%**	**-57%**
C5 with ANN	0%	-10%
CART with MLR	1,318%	322%
CART with ANN	-4%	-11%
SVM with MLR	20%	-11%
SVM with ANN	10%	-1%
Stepwise Regression with MLR	507%	103%
Stepwise Regression with ANN	9%	-7%
Genetic Algorithm with MLR *	**-57%**	**-60%**
Genetic Algorithm with ANN	0	-9%
Chi-Square with MLR ***	**-52%**	**-51%**
Chi-Square with ANN	0%	-10%
Information Gain with MLR ***	**-52%**	**-51%**
Information Gain with ANN	0%	-10%

จากตารางที่ 4.7 ซึ่งได้แสดงค่า Error Reduction Rate ของวิธีการคัดเลือกคุณลักษณะทั้ง 7 ที่ทำงานร่วมกับอัลกอริทึมการเรียนรู้เพื่อสร้างโมเดลการคาดการณ์ผลผลิตทั้ง 2 ชนิด เมื่อเปรียบเทียบกับวิธีการดั้งเดิมนั้น จะทำให้ได้วิธีการทั้งหมด 14 วิธี โดยจะมี 5 วิธีที่ค่า Error Reduction $Rate_{(RMSE)}$ ที่มีค่าติดลบ ซึ่งหมายความว่าประสิทธิภาพของการคาดการณ์ผลผลิตที่นำเสนอนั้นดีกว่าวิธีการดั้งเดิม ได้แก่ วิธีการ C5 ทำงานร่วมกับ ANN, วิธีการ CART ทำงานร่วมกับ ANN, Genetic Algorithm ทำงานร่วมกับ MLR, วิธีการ Chi-Square ทำงานร่วมกับ MLR และ Information Gain ทำงานร่วมกับ MLR ค่าที่ดีที่สุดอยู่ที่ -57%

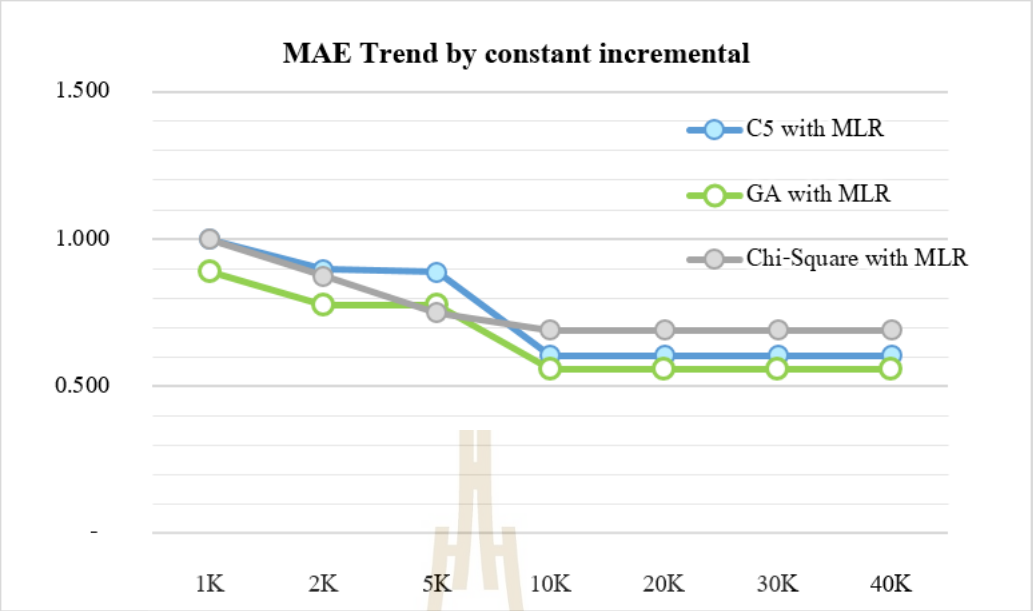
ในส่วนของค่า Error Reduction Rate$_{(MAE)}$ นั้น มีวิธีการจำนวนถึง 12 จาก 14 วิธีการที่มีค่าติดลบ ค่าที่ Error Reduction Rate$_{(MAE)}$ ดีที่สุด ได้แก่ -60% จากวิธีการ Genetic Algorithm ทำงานร่วมกับ MLR โดยจะมีเพียง 2 วิธีการที่ให้ผลลัพธ์เป็นบวกนั่น คือ วิธีการ CART ทำงานร่วมกับ MLR และวิธีการ Stepwise Regression ทำงานร่วมกับ MLR ซึ่งให้ค่า Error reduction rate$_{(MAE)}$ 322% และ 103% ตามลำดับ

4.4 ผลการทดลองเพิ่มเติมในการเพิ่มจำนวนค่าคงที่ของข้อมูลในการรวมกลุ่ม

ขั้นตอนนี้จะเป็นการทดลองเพิ่มเติมในส่วนของการเปลี่ยนค่าคงที่ของการจัดกลุ่มข้อมูลและศึกษาแนวโน้มของการเปลี่ยนแปลงค่าคงที่ว่าจะส่งผลอย่างไรต่อประสิทธิภาพการคาดการณ์ผลผลิต โดยจะนำวิธีการที่ดีที่สุดทั้งสามวิธีการมาใช้ในการทดลองครั้งนี้ ได้แก่ วิธีการคุณลักษณะจากคัดเลือกด้วย C5, Genetic Algorithm และ Information Gain ทำงานร่วมกับการสร้างโมเดลการคาดการณ์ผลผลิตด้วยอัลกอริทึม MLR ซึ่งการเพิ่มขึ้นของการรวมกลุ่มข้อมูลนั้นจะทำครั้งละ 10,000 ข้อมูล โดยเริ่มที่ 10,000 (จากการทดลองที่แล้ว) ไปจนถึง 40,000 นอกจากนั้นแล้วทางผู้วิจัยได้ทำการทดลองเพิ่มเติมในการใช้ค่าคงที่ที่ 1,000 2,000 และ 5,000 เพิ่มเข้าไปอีกด้วย ซึ่งผลการทดลองที่ได้นั้นแสดงไว้ในรูปที่ 4.1 รูปที่ 4.2 และตารางที่ 4.8



รูปที่ 4.1 กราฟแสดงแนวโน้มของความสัมพันธ์ระหว่างประสิทธิภาพในการคาดการณ์ผลผลิตและค่าคงที่ของจำนวนในการรวมกลุ่มข้อมูลโดยใช้ตัววัดประสิทธิภาพเป็น RMSE



รูปที่ 4.2 กราฟแสดงแนวโน้มของความสัมพันธ์ระหว่างประสิทธิภาพในการคาดการณ์ผลผลิต และค่าคงที่ของจำนวนในการรวมกลุ่มข้อมูลโดยใช้ตัววัดประสิทธิภาพเป็น MAE

ตารางที่ 4.8 แสดงค่า RMSE และ MAE เมื่อมีการเพิ่มจำนวนของข้อมูลในการรวมกลุ่ม จาก 1,000 ถึง 40,000 ข้อมูลต่อแถว

	RMSE						
Method	1K	2K	5K	10K	20K	30K	40K
C5 with MLR	1.430	1.285	1.271	0.866	0.866	0.866	0.866
GA with MLR	1.163	1.017	1.017	0.732	0.732	0.732	0.732
Chi-Square with MLR	1.189	1.041	0.892	0.821	0.821	0.821	0.821

	MAE						
Method	1K	2K	5K	10K	30K	35K	40K
C5 with MLR	0.999	0.898	0.888	0.605	0.605	0.605	0.605
GA with MLR	0.888	0.777	0.777	0.559	0.559	0.559	0.559
Chi-Square with MLR	0.999	0.875	0.750	0.690	0.690	0.690	0.690

4.5 อภิปรายผลการทดลอง

จากผลการทดลองเพิ่มประสิทธิภาพการคาดการณ์ผลผลิต ด้วยการคัดเลือกคุณลักษณะที่สำคัญต่อการจำแนกข้อมูลในคลาสเป้าหมาย (สถานะการทดสอบผลิตภัณฑ์ STATUS = Pass หรือ Fail) มาเพื่อสร้างโมเดลการคาดการณ์ผลผลิตนั้น สามารถนำมาสรุปและอภิปรายผลได้ดังต่อไปนี้

1. ในการคัดเลือกคุณลักษณะที่มีความสำคัญต่อการจำแนกคลาสเป้าหมายนั้นมีทั้งหมด 7 ชนิด โดยทั้ง 5 ชนิดแรกที่เป็นตัวแทนของอัลกอริทึมการเรียนรู้อันได้แก่ C5, CART, SVM, Stepwise Regression และ GA ได้มีการคัดเลือกคุณลักษณะที่หลากหลายแตกต่างกันออกไปซึ่งจะมีทั้งคุณลักษณะที่สภาพของชิ้นส่วนต่าง ๆ ที่นำมาประกอบเป็นฮาร์ดดิสก์ไดรฟ์ ไลน์การผลิตฮาร์ดดิสก์ไดรฟ์ ไลน์การผลิตชิ้นส่วน ไลน์การประกอบชิ้นส่วนย่อยเข้าด้วยกัน รวมไปถึงรุ่นของเครื่องจักร และโปรแกรมที่ใช้ในการทดสอบผลิตภัณฑ์ในขณะที่วิธีการคัดเลือกคุณลักษณะแบบ Chi-Square และ Information Gain นั้นที่เลือกคุณลักษณะด้วยชุดเดียวกันทั้งหมด ซึ่งทั้งสองวิธีนี้ต่างเป็นวิธีการที่เป็นตัวแทนการคัดเลือกคุณลักษณะด้วยวิธีทางสถิติเหมือนกัน โดยเมื่อนำคุณลักษณะที่ได้จากการคัดเลือกในการทดลองนี้มาเปรียบเทียบกับคุณลักษณะที่ได้จากประสบการณ์ของวิศวกรผู้ชำนาญการนั้นจะพบว่า คุณลักษณะที่ได้จากการคัดเลือกในการทดลองนี้มีความหลากหลายมากกว่าดังที่ได้แสดงไว้ในตารางที่ 4.3 ซึ่งแตกต่างจากคุณลักษณะที่ได้จากประสบการณ์ของวิศวกรผู้ชำนาญการที่จะมุ่งเน้นไปที่สภาพของชิ้นส่วนต่าง ๆ ที่นำมาประกอบเป็นฮาร์ดดิสก์ไดรฟ์

2. ค่าความแม่นยำในการคัดเลือกคุณลักษณะของอัลกอริทึมที่มีความซับซ้อนไม่มากอย่างเช่น ตระกูลต้นไม้ตัดสินใจอันได้แก่ C5 และ CART จะให้ค่าความแม่นยำที่ค่อนข้างสูงรวมถึงใช้เวลาในการสร้างโมเดลการเรียนรู้ที่ไม่มากนัก ในขณะที่อัลกอริทึมที่มีความซับซ้อนสูงเช่น Genetic Algorithm จะใช้เวลาในการสร้างโมเดลค่อนข้างมาก และในส่วนของการคัดเลือกคุณลักษณะด้วยวิธีทางสถิติอันได้แก่ Chi-Square และ Information Gain นั้นจะใช้เวลาในการประมวลผลเพียง 10 วินาทีซึ่งนับได้ว่าใช้เวลาน้อยกว่าวิธีการอื่นมาก

3. การรวมกลุ่มข้อมูลด้วยแนวคิดใหม่ที่ได้นำเสนอในงานวิจัยชิ้นนี้นั้นแสดงให้เห็นถึงการลดความแปรปรวนของจำนวนข้อมูลในแต่ละแถว ซึ่งตรงจุดนี้เองจะส่งผลกระทบต่อการคำนวณและคาดการณ์ผลผลิตในขั้นตอนถัดไป

4. จากผลการทดลองในการใช้คุณลักษณะที่ถูกคัดเลือกมาจากวิธีการคัดเลือกคุณลักษณะทั้ง 7 ชนิด นั้นการวัดประสิทธิภาพแบบ RMSE พบว่าการทำงานร่วมกับอัลกอริทึมการเรียนรู้แบบ ANN ให้ผลลัพธ์ของการคาดการณ์ผลผลิตที่ไม่แตกต่างจากวิธีการดั้งเดิมมากนัก ในขณะที่การใช้

MLR เป็นอัลกอริทึมในการเรียนรู้การสร้างโมเดลการคาดการณ์ผลผลิตจะให้ผลลัพธ์ที่ดีกว่าแบบดั้งเดิมถึง 4 วิธี ซึ่งค่า RMSE ที่ดีที่สุดเป็นการทำงานร่วมกันระหว่าง Genetic Algorithm และ MLR

5. โมเดลการคาดการณ์ที่สร้างจากอัลกอริทึม ANN นั้น ถึงแม้จะไม่ได้เป็นวิธีการที่ให้ประสิทธิภาพของการคาดการณ์ผลผลิตที่ดีที่สุด ดังที่แสดงไว้ในตารางที่ 4.6 แต่เมื่อทางผู้วิจัยได้ทำตารางสรุปการเปรียบเทียบประสิทธิภาพแล้ว พบว่า ANN นั้นไม่ว่าจะจับคู่กับการคัดเลือกคุณลักษณะด้วยวิธีใดก็จะให้ผลลัพธ์ที่ดีกว่าหรือเท่ากับวิธีการคาดการณ์ผลผลิตแบบดั้งเดิมตามที่ได้แสดงไว้ในตารางที่ 4.9 ซึ่งผลลัพธ์นี้ทางผู้วิจัยคาดว่ามาจากการที่ ANN เป็นอัลกอริทึมที่มีความซับซ้อนสูงและใช้งานข้อมูลโดยวิธีการสร้างโหนดแล้วเชื่อมโยงเข้าหากันในหลายระดับชั้น จึงทำให้การทำงานในการสร้างโมเดลการเรียนรู้นั้นมีความยืดหยุ่นสูง โดยไม่ว่าจะทำงานกับจำนวนข้อมูลมากหรือน้อยก็จะทำงานได้ค่อนข้างดี ด้วยเหตุนี้เองจึงทำให้ ANN อาจจะเป็นทางเลือกที่ดีในลักษณะข้อมูลที่มีความแปรปรวนสูงโดยเป็นอีกตัวเลือกที่เหมาะสมกับการนำไปใช้กรณีที่ต้องการการคาดการณ์ผลผลิตที่ดีกว่าเดิม ที่ไม่เน้นว่าจะต้องมีความแม่นยำของการคาดการณ์ที่สูงที่สุด

ตารางที่ 4.9 แสดงการเปรียบเทียบประสิทธิภาพของวิธีการที่นำเสนอเปรียบเทียบกับวิธีการดั้งเดิม

		Test Data	
		RMSE	**MAE**
MLR	C5	Better	Better
	CART	Worse	Worse
	SVM	Worse	Better
	Stepwise Regression	Worse	Worse
	Genetic Algorithm	Best of RMSE	Best of MAE
	Chi-Square	Better	Better
	Information Gain	Better	Better
ANN	C5	Comparable	Better
	CART	Comparable	Better
	SVM	Comparable	Comparable
	Stepwise Regression	Comparable	Comparable
	Genetic Algorithm	Comparable	Comparable
	Chi-Square	Comparable	Better
	Information Gain	Comparable	Better

บทที่ 5
บทสรุป

ในอุตสาหกรรมการผลิตฮาร์ดดิสก์ไดรฟ์นั้นมีขั้นตอนการประกอบชิ้นส่วนต่าง ๆ ที่มากมาย รวมไปถึงการทดสอบผลิตภัณฑ์ที่ซับซ้อนและยาวนาน การคาดการณ์ผลผลิตของการทดสอบผลิตภัณฑ์นั้นนับได้ว่าเป็นงานที่มีความสำคัญที่สุดอย่างหนึ่ง โดยมีวัตถุประสงค์ที่จะนำผลลัพธ์ของการคาดการณ์นั้นไปใช้งานในหลายวัตถุประสงค์ เช่น การวางแผนการใช้งานเครื่องจักรในการผลิต การวางแผนงานด้านการสั่งซื้อวัตถุดิบ การวางแผนงานในการใช้งานเครื่องจักรเพื่อทดสอบผลิตภัณฑ์ การวางแผนส่งมอบผลิตภัณฑ์แก่ลูกค้า และการตั้งราคาสินค้าเพื่อจัดจำหน่าย เป็นต้น

ในปัจจุบันมีนักวิจัยรวมไปถึงวิศวกรที่ทำงานเกี่ยวข้องกับการผลิตในอุตสาหกรรมเป็นจำนวนมากได้พยายามสร้างวิธีการเพื่อให้ได้การคาดการณ์ผลผลิตที่มีประสิทธิภาพในด้านความแม่นยำให้ได้มากที่สุด ซึ่งส่วนใหญ่จะมุ่งเน้นไปที่การหาแอททริบิวต์ หรือคุณลักษณะที่ส่งผลต่อการคำนวณยีลด์ (ผลผลิตของกระบวนการทดสอบผลิตภัณฑ์) แล้วจึงนำอัตราส่วนของคุณลักษณะเหล่านั้นไปใช้ในการคาดการณ์ผลผลิตในแต่ละช่วงสัปดาห์ของการผลิตฮาร์ดดิสก์ไดรฟ์ โดยคุณลักษณะเหล่านั้นจะถูกนำไปประยุกต์ใช้กับทุกรุ่นของของฮาร์ดดิสก์ไดรฟ์ ซึ่งการทำงานในลักษณะนี้จะเป็นการอาศัยความรู้และประสบการณ์ที่สืบทอดมาจากวิศวกรรุ่นก่อนหน้า

ดังนั้นวัตถุประสงค์ของงานวิจัยนี้ คือการเสนอเทคนิคและวิธีการเพิ่มประสิทธิภาพในการทำงานทางด้านการคาดการณ์ผลผลิตในอุตสาหกรรมการผลิตฮาร์ดดิสก์ไดรฟ์ โดยการนำความรู้ทางด้านการเรียนรู้ของเครื่อง และอัลกอริทึมในการสร้างโมเดลการเรียนรู้มาประยุกต์ใช้ นอกเหนือจากนั้นแล้วในงานวิจัยนี้ได้นำเสนอแนวคิดใหม่ในการรวมกลุ่มข้อมูลด้วยค่าคงที่ แทนที่การรวมกลุ่มของข้อมูลแบบดั้งเดิมที่จะอ้างอิงจากระยะเวลาตามปฏิทิน ได้แก่ รายวัน รายสัปดาห์ และรายเดือน

ในส่วนของชุดข้อมูลที่นำมาใช้ในงานวิจัยนั้น ทางผู้วิจัยได้เลือกใช้ชุดข้อมูลจริงจากการผลิตฮาร์ดดิสก์ไดรฟ์ โดยเป็นชุดข้อมูลที่ครอบคลุมระยะเวลาในการผลิตเป็นระยะเวลาถึง 3 ปี โดยมีจำนวนข้อมูล 10,000,000 แถว ซึ่งเป็นการแสดงให้เห็นว่าวิธีการที่นำเสนอในงานวิจัยนี้สามารถนำไปใช้งานได้จริง

5.1 สรุปผลการวิจัย

จากผลการดำเนินงานวิจัยในการทดลองเพิ่มประสิทธิภาพการคาดการณ์ผลผลิตในอุตสาหกรรมการผลิตฮาร์ดดิสก์ไดรฟ์นั้น สามารถสรุปได้เป็นขั้นตอนดังที่แสดงไว้ในตารางที่ 5.1

ตารางที่ 5.1 สรุปวิธีการตามขั้นตอนการดำเนินงานวิจัย

ลำดับ	ขั้นตอนและวิธีการ	ผลลัพธ์ที่ได้
1	ขั้นตอนการปรับสมดุลข้อมูลโดยใช้อัลกอริทึม k-Means Clustering, k-NN และ Resampling มาประยุกต์ใช้งานเป็นวิธีการใหม่ชื่อว่า DBC-2KAR	อัตราส่วนของความไม่สมดุลลดลงจาก 28: 1 เป็น 1:1 ซึ่งส่งผลให้โมเดลการเรียนรู้สามารถจำแนกคลาสเป้าหมายได้และสามารถหาคุณลักษณะที่สำคัญต่อคลาสเป้าหมายได้
2	ขั้นตอนการคัดเลือกคุณลักษณะ วิธีการที่ใช้ในการคัดเลือก CART, SVM, Stepwise Regression, Genetic Algorithm, Chi-Square และ Information Gain	คุณลักษณะที่ส่งผลต่อการทำนายสถานะการทดสอบผลิตภัณฑ์ที่ได้รับจากวิธีการคัดเลือกคุณลักษณะ
3	การรวมกลุ่มของข้อมูลด้วยค่าคงที่ 10,000 ข้อมูลต่อแถว	ชุดข้อมูลใหม่ที่เกิดจากการรวมกลุ่มของข้อมูลเดิม 10,000,000 ข้อมูล จากเดิม 157 แถว (รวมกลุ่มข้อมูลรายสัปดาห์) เป็น 1,000 แถว (รวมกลุ่มข้อมูลด้วยค่าคงที่ 10,000 ข้อมูลต่อแถว)
4	ขั้นตอนการทดสอบวัดประสิทธิภาพเปรียบเทียบกับวิธีการดั้งเดิม อัลกอริทึมที่ใช้ในการสร้างโมเดลการเรียนรู้ ได้แก่ Multiple Linear Regression และ Artificial Neural Networks	[A] ค่าจากการวัดประสิทธิภาพด้วยตัววัดประสิทธิภาพ Root Means Square Error และ Mean Absolute Error [B] การเปรียบเทียบด้วย Error Reduction Rate (อัตราส่วนการทำนายผิดพลาดที่ลดลง)

ตารางที่ 5.1 สรุปวิธีการตามขั้นตอนการดำเนินงานวิจัย (ต่อ)

ลำดับ	ขั้นตอนและวิธีการ	ผลลัพธ์ที่ได้
5	ขั้นตอนการทดสอบความสัมพันธ์ของแนวโน้มประสิทธิภาพกับ การเพิ่มขึ้นของค่าคงที่ที่ใช้ในการรวมกลุ่มข้อมูล การรวมกลุ่มข้อมูล แถวละ 1,000 2,000 5,000 10,000 20,000 30,000 40,000	แนวโน้มของประสิทธิภาพที่วัดด้วย RMSE และ MAE เมื่อเพิ่มค่าคงที่ของการรวมกลุ่มข้อมูล

จากตารางข้างต้นสามารถอธิบายได้ดังนี้

1) ขั้นตอนการปรับสมดุลข้อมูล ในขั้นตอนนี้ได้ทำการทดลองโดยการให้ความสำคัญกับคุณลักษณะที่ชื่อสถานะการทดสอบ (STATUS) โดยคุณลักษณะนี้เป็นตัวบ่งชี้ว่าฮาร์ดดิสก์ไดรฟ์นั้น ๆ ผ่านการทดสอบ (Pass) หรือไม่ผ่านการทดสอบ (Fail) ผู้วิจัยได้เลือกใช้อัลกอริทึมที่หลากหลายมาประยุกต์ใช้ในการทำสมดุลข้อมูล ได้แก่ k-Means Clustering, k-NN และ Resampling นำไปสู่วิธีการทำสมดุลข้อมูลที่มีชื่อว่า DBC-2KAR ซึ่งส่งผลลัพธ์ในการเพิ่มประสิทธิภาพของการนำเอาโมเดลการเรียนรู้ด้านการจำแนกมาประยุกต์ใช้ในการคัดเลือกคุณลักษณะเนื่องจากสามารถทำให้ชุดข้อมูลนี้มีอัตราส่วนความไม่สมดุลลดลงจาก 28:1 เป็น 1:1

2) ผลลัพธ์จากการทดลองในขั้นตอนการคัดเลือกคุณลักษณะนั้นมีวิธีการคัดเลือกคุณลักษณะทั้งสิ้น 7 ชนิด ทางผู้วิจัยได้นำเอาวิธีการที่เป็นตัวแทนของอัลกอริทึมการเรียนรู้ของเครื่อง 5 ชนิดได้แก่ CART, SVM, Stepwise Regression และ Genetic Algorithm ในส่วนของวิธีการที่เป็นตัวแทนของเทคนิคด้านสถิติมี 2 ชนิด ได้แก่ Chi-Square และ Information Gain ซึ่งคุณลักษณะที่ได้รับการคัดเลือกมาจากวิธีการทั้ง 7 ชนิดนั้น จะมีความหลากหลายมากกว่าเมื่อเทียบกับคุณลักษณะที่ใช้ในกระบวนการดั้งเดิมของ Human Expert ที่จะมีเพียงคุณลักษณะในหมวดหมู่ของสภาพชิ้นส่วนหลักเท่านั้น ด้วยเหตุนี้เองคุณลักษณะที่จะถูกนำไปใช้ในการคำนวณยีลด์ และการคาดการณ์ยีลด์นั้นจะมีรายละเอียดที่มากกว่าซึ่งนำไปสู่ผลลัพธ์ที่มีประสิทธิภาพและประสิทธิผลมากกว่า

3) จากผลการทดลองในขั้นตอนการทดสอบเปรียบเทียบประสิทธิภาพนั้น จะมีสองอัลกอริทึมในการสร้างโมเดลการเรียนรู้การคาดการณ์ยีลด์ ได้แก่ ANN และ MLR โดยมีตัววัดประสิทธิภาพสองชนิด ได้แก่ RMSE และ MAE ซึ่งผลลัพธ์ที่ได้จากตัววัดประสิทธิภาพทั้งสองจะมีแนวโน้มไปในทิศทางเดียวกัน คือ การใช้ MLR ในการสร้างโมเดลการเรียนรู้ทำงานร่วมกับ

คุณลักษณะที่ได้รับจากการคัดเลือกด้วยวิธีการ Genetic Algorithm นั้นจะให้ผลลัพธ์ที่ดีที่สุด โดยรองลงมาจะเป็นการใช้ MLR ในการสร้างโมเดลการเรียนรู้ทำงานร่วมกับการคัดเลือกคุณลักษณะด้วยวิธีการต้นไม้ตัดสินใจโดยใช้อัลกอริทึม C5

ในส่วนของการคัดเลือกคุณลักษณะด้วยวิธีการ Information Gain และ Chi-square นั้นจะได้ประสิทธิภาพที่เท่ากันเนื่องมาจากได้คุณลักษณะที่เป็นชุดเดียวกันทั้ง 10 คุณลักษณะ

นอกเหนือจากนั้นแล้วการทดลองในขั้นตอนนี้ยังทำให้ค้นพบว่า โมเดลการเรียนรู้ที่สร้างจากอัลกอริทึม ANN นั้นจะให้ประสิทธิภาพที่ดีกว่าวิธีการดั้งเดิมเสมอ ไม่ว่าทำงานร่วมกับวิธีการคัดเลือกคุณลักษณะแบบใดก็ตาม

4) การทดสอบวัดประสิทธิภาพ เปรียบเทียบการเพิ่มขึ้นของค่าคงที่ที่ใช้ในการรวมกลุ่มข้อมูลในขั้นตอนนี้ผู้วิจัยได้นำวิธีการที่ดีที่สุดทั้งสามวิธีการ ได้แก่ วิธีการที่นำคุณลักษณะจากคัดเลือกด้วย C5, Genetic Algorithm ทำงานร่วมกับการสร้างโมเดลการคาดการณ์ยีลด์ด้วยอัลกอริทึม MLR มาใช้ในการทดสอบหาความสัมพันธ์ระหว่างแนวโน้มของประสิทธิภาพที่ได้รับกับค่าคงที่ของจำนวนข้อมูลที่ใช้ในการรวมกลุ่มที่เพิ่มขึ้น ซึ่งผลลัพธ์ที่ได้นั้นแสดงให้เห็นว่า ทั้งค่า RMSE และ MAE นั้นมีแนวโน้มที่ลดลงอย่างมีนัยยะสำคัญเมื่อค่าคงที่ของจำนวนข้อมูลเพิ่มขึ้นจาก 1,000 ไปเป็น 2,000 5,000 และ 10,000 ตามลำดับ โดยค่า RMSE และ MAE จะเท่าเดิมกับการใช้ค่าคงที่ของจำนวนการรวมกลุ่มข้อมูลที่ 10,000 แม้ว่าจะเพิ่มจำนวนการรวมกลุ่มข้อมูลไปที่ 20,000 30,000 และ 40,000 ซึ่งเป็นการแสดงให้เห็นว่าแนวโน้มประสิทธิภาพของการคาดการณ์นั้นจะมีประสิทธิภาพที่ดีขึ้นเรื่อย ๆ ไปจนถึงการรวมกลุ่มข้อมูลด้วยค่าคงที่ที่จำนวนข้อมูล 10,000 แล้วจากนั้นจะประสิทธิภาพที่ได้จะคงที่

5.2 ข้อเสนอแนะ

จากผลการศึกษาที่ได้รับจากงานวิจัยชิ้นนี้ สามารถสรุปได้เป็นประเด็นเพื่อการเสนอแนะการนำไปประยุกต์ใช้งาน และนำไปต่อยอดให้มีประสิทธิภาพที่ดียิ่งขึ้นไปตามหัวข้อดังต่อไปนี้

1) ในการใช้งานอัลกอริทึม ANN เพื่อสร้างโมเดลการเรียนรู้สำหรับการคาดการณ์ผลผลิตนั้น แม้ว่าจะไม่ได้ให้ผลผลัพธ์ที่ดีที่สุดสำหรับการทดลองในงานวิจัยชิ้นนี้ แต่พบว่าไม่ว่าอัลกอริทึม ANN จะทำงานร่วมกับการคัดเลือกคุณลักษณะด้วยวิธีการใดก็ตาม จะให้ผลลัพธ์ที่คงและดีขึ้นกว่าวิธีการดั้งเดิม ซึ่งทางผู้วิจัยมีความเห็นว่าเหมาะกับการนำไปประยุกต์ใช้งานในส่วนที่ต้องการเพิ่มประสิทธิภาพให้สูงขึ้น โดยไม่ต้องการที่จะมีความเสี่ยงที่จะทำให้การคาดการณ์ลดลงจากวิธีการเดิม

นอกเหนือจากนั้นแล้วการทดลองเพิ่มประสิทธิภาพของในการคาดการณ์ผลผลิตด้วยอัลกอริทึม ANN คาดหมายว่ายังมีโอกาสที่จะพัฒนาต่อไปได้อีก ไม่ว่าจะเป็นการคัดเลือกคุณลักษณะด้วยวิธีการอื่น ๆ หรือการนำเทคนิคใหม่ในอนาคตมาประยุกต์ใช้งาน

2) ข้อมูลที่นำมาใช้ในงานวิจัยชิ้นนี้นั้นมาจากชุดข้อมูลจริงในอุตสาหกรรมการผลิตและประกอบฮาร์ดดิสก์ไดรฟ์ โดยข้อมูลที่นำมาใช้นั้นมาจากรุ่นของผลิตภัณฑ์ที่อยู่ในช่วงเวลาของ Product Maturity ซึ่งหมายถึงการที่มีข้อมูลจำนวนมหาศาลและยีลด์ของการทดสอบผลิตภัณฑ์จะค่อนข้างสูง จึงทำให้เห็นความแตกต่างของการเพิ่มประสิทธิภาพด้วยเทคนิคที่นำเสนอได้ไม่มากนัก ดังนั้นหากมีโอกาสที่จะทดสอบเพื่อเพิ่มประสิทธิภาพในการคาดการณ์ผลผลิตด้วยชุดข้อมูลที่มาจากรุ่นของผลิตภัณฑ์ที่อยู่ในช่วงเวลา New Product Launch ที่โดยปกติแล้วจะมียีลด์ของการทดสอบผลิตภัณฑ์จะค่อนข้างต่ำและมีความแปรปรวนสูง ซึ่งอาจจะทำให้การทดสอบเพิ่มประสิทธิภาพนั้นสามารถแสดงผลลัพธ์ได้เด่นชัดขึ้น

แต่อย่างไรก็ตามการใช้ข้อมูลที่เก็บมาจากช่วงระยะเวลาของ New Product Launch นั้นจะมีจำนวนค่อนข้างน้อย โดยปกติแล้วจะมีข้อมูลเพียงจำนวนหลักพัน หรือหลักหมื่นต่อไตรมาส ซึ่งสิ่งนี้เองอาจจะทำให้การทดลองประสบกับปัญหาการที่มีข้อมูลน้อยจนเกินไป โดยการต่อยอดนั้นอาจจะต้องคำนึงถึงการเพิ่มจำนวนข้อมูลด้วยวิธีการ Over-Sampling ในรูปแบบอื่นร่วมด้วย

3) ถึงแม้ว่าผลลัพธ์จากการทดลองประยุกต์ใช้เทคนิคการรวมกลุ่มข้อมูลด้วยค่าคงที่นั้นจะเพิ่มประสิทธิภาพการคาดการณ์ผลผลิตในอุตสาหกรรมการผลิตฮาร์ดดิสก์ไดรฟ์ได้เป็นที่น่าพอใจ แต่ทางผู้วิจัยได้เล็งเห็นถึงการนำเทคนิคที่นำเสนอนี้ไปต่อยอดกับอุตสาหกรรมอื่น ๆ เช่น อุตสาหกรรมการผลิตชิ้นส่วนอิเล็กทรอนิกส์ อุตสาหกรรมการผลิตเครื่องจักร อุตสาหกรรมการผลิตและประกอบชิ้นส่วนยานยนต์ หรือแม้แต่อุตสาหกรรมการผลิตภาคเกษตรกรรม ซึ่งโดยส่วนใหญ่จะมีการรายงานผลผลิตเป็นรายวัน รายสัปดาห์ หรือรายเดือน ซึ่งมีโอกาสที่จะใช้เทคนิค Data Aggregation นี้ไปประยุกต์ใช้เพื่อทดสอบเปรียบเทียบประสิทธิภาพได้ในอนาคต

รายการอ้างอิง

กีระชาติ สุขสุทธิ์. (2559). การจำแนกข้อมูลไม่สมดุลโดยใช้การปรับปรุงข้อมูลร่วมกับการหาค่าพารามิเตอร์ที่เหมาะสมด้วยขั้นตอนวิธีเชิงพันธุกรรมที่มีการเริ่มต้นใหม่. **นครราชสีมา : มหาวิทยาลัยเทคโนโลยีสุรนารี.**

มนต์ชัย เทียนทอง. (2548). สถิติและวิธีการวิจัยทางเทคโนโลยีสารสนเทศ. **กรุงเทพฯ: สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ.**

รติพร จันทร์กลั่น. (2560). การสร้างแบบจำลองด้วยเทคนิคการเรียนรู้ของเครื่องเพื่อคาดการณ์ปริมาณน้ำท่า. **นครราชสีมา : มหาวิทยาลัยเทคโนโลยีสุรนารี.**

Aiken, L. S., West, S. G., Pitts, S. C., Baraldi, A. N., & Wurpts, I. C. (2012). Multiple linear regression. **Handbook of Psychology, Second Edition**, 2.

Althuwaynee, O. F., Pradhan, B., Park, H.-J., & Lee, J. H. (2014). A novel ensemble decision tree-based CHi-squared Automatic Interaction Detection (CHAID) and multivariate logistic regression models in landslide susceptibility mapping. **Landslides**, 11(6), 1063–1078.

An, D., Ko, H.-H., Gulambar, T., Kim, J., Baek, J.-G., & Kim, S.-S. (2009). A semiconductor yields prediction using stepwise support vector machine. **Proceeding of 2009 IEEE International Symposium on Assembly and Manufacturing**, 130–136.

Arefnezhad, S., Samiee, S., Eichberger, A., & Nahvi, A. (2019). Driver drowsiness detection based on steering wheel data applying adaptive neuro-fuzzy feature selection. **Sensors**, 19(4), 943.

Baldi, P., Brunak, S., Chauvin, Y., Andersen, C. A. F., & Nielsen, H. (2000). Assessing the accuracy of prediction algorithms for classification: an overview. **Bioinformatics**, 16(5), 412–424.

Banzhaf, W., Nordin, P., Keller, R. E., & Francone, F. D. (1998). Genetic programming. Springer.

Barandela, R., Valdovinos, R. M., Sánchez, J. S., & Ferri, F. J. (2004). The imbalanced training sample problem: Under or over sampling? **Proceedings of Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)**, 806–814.

Basu, S., Davidson, I., & Wagstaff, K. (2008). Constrained clustering: Advances in algorithms, theory, and applications. CRC Press.

Berger, P. D., Maurer, R. E., & Celli, G. B. (2018). Introduction to simple regression. In *Experimental Design* (pp. 483–503). Springer.

Bertinetto, L., Henriques, J. F., Valmadre, J., Torr, P., & Vedaldi, A. (2016). Learning feed-forward one-shot learners. **Advances in Neural Information Processing Systems**, 29, 523–531.

Bies, R. R., Muldoon, M. F., Pollock, B. G., Manuck, S., Smith, G., & Sale, M. E. (2006). A genetic algorithm-based, hybrid machine learning approach to model selection. **Journal of Pharmacokinetics and Pharmacodynamics**, 33(2), 195–221.

Breiman, L., Friedman, J. H., Olshen, R. A., & others. (2017). *Classification and regression trees Routledge*.

Chae, Y. T., Horesh, R., Hwang, Y., & Lee, Y. M. (2016). Artificial neural network model for forecasting sub-hourly electricity usage in commercial buildings. **Energy and Buildings**, 111, 184–194.

Chai, T., & Draxler, R. R. (2014). Root mean square error (RMSE) or mean absolute error (MAE)? **GMDD**, 7(1), 1525–1534.

Chatterjee, S., & Hadi, A. S. (2015). Regression analysis by example. John Wiley & Sons.

Chawla, N. V. (2009). Data mining for imbalanced datasets: An overview. In *Data mining and knowledge discovery handbook* (pp. 875–886). Springer.

Chawla, N. V, Japkowicz, N., & Kotcz, A. (2004). Special issue on learning from imbalanced data sets. **ACM SIGKDD Explorations Newsletter**, 6(1), 1–6.

Chen, T. (2017). An ANN approach for modeling the multisource yield learning process with semiconductor manufacturing as an example. **Computers & Industrial Engineering**, 103, 98–104.

Cheng, Y., Chen, K., Sun, H., Zhang, Y., & Tao, F. (2018). Data and knowledge mining with big data towards smart production. **Journal of Industrial Information Integration**, 9, 1–13.

Darwin, C., & Bynum, W. F. (2009). The origin of species by means of natural selection: or, the preservation of favored races in the struggle for life. AL Burt New York.

Davis, J., & Goadrich, M. (2006). The relationship between Precision-Recall and ROC curves. **Proceedings of the 23rd International Conference on Machine Learning**, 233–240.

Elkan, C. (2003). Using the triangle inequality to accelerate k-means. **Proceedings of the 20th International Conference on Machine Learning (ICML-03)**, 147–153.

Fawcett, T. (2006). An introduction to ROC analysis. **Pattern Recognition Letters**, 27(8), 861–874.

Fox, J. (2015). Applied regression analysis and generalized linear models. Sage Publications.

Fukunaga, K., & Narendra, P. M. (1975). A branch and bound algorithm for computing k-nearest neighbors. **IEEE Transactions on Computers**, 100(7), 750–753.

Furey, T. S., Cristianini, N., Duffy, N., Bednarski, D. W., Schummer, M., & Haussler, D. (2000). Support vector machine classification and validation of cancer tissue samples using microarray expression data. **Bioinformatics**, 16(10), 906–914.

Gardner, M. W., & Dorling, S. R. (1998). Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. **Atmospheric Environment**, 32(14–15), 2627–2636.

Guenther, N., & Schonlau, M. (2016). Support vector machines. **The Stata Journal**, 16(4), 917–937.

Han, H., Wang, W.-Y., & Mao, B.-H. (2005). Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. **Proceeding of International Conference on Intelligent Computing**, 878–887.

Harrell Jr, F. E. (2015). Regression modeling strategies: with applications to linear models, logistic and ordinal regression, and survival analysis. Springer.

He, H., & Garcia, E. A. (2009). Learning from Imbalanced Data IEEE Transactions on Knowledge and Data Engineering v. 21 n. 9. September.

Hirunyawanakul, A., Kerdprasop, N., & Kerdprasop, K. (2020). Efficient Machine Learning Methods for Hard Disk Drive Yield Prediction Improvement. **International Journal of Machine Learning and Computing,** 10(2), 240–246.

Hsu, C.-W., Chang, C.-C., Lin, C.-J., & others. (2003). A practical guide to support vector classification. Taipei.

Hsu, C.-W., & Lin, C.-J. (2002). A comparison of methods for multiclass support vector machines. **IEEE Transactions on Neural Networks**, 13(2), 415–425.

Jadhav, S., He, H., & Jenkins, K. (2018). Information gain directed genetic algorithm wrapper feature selection for credit rating. **Applied Soft Computing**, 69, 541–553.

Jain, A. K., Mao, J., & Mohiuddin, K. M. (1996). Artificial neural networks: A tutorial. **Computer**, 29(3), 31–44.

Janes, J. (2001). Categorical relationships: chi-square. **Library Hi Tech**.

Japkowicz, N., & others. (2000). Learning from imbalanced data sets: a comparison of various strategies. **Proceedings of AAAI Workshop on Learning from Imbalanced Data Sets**, 68, 10–15.

Jin, X., Xu, A., Bie, R., & Guo, P. (2006). Machine learning techniques and chi-square feature selection for cancer classification using SAGE gene expression profiles. **Proceedings of International Workshop on Data Mining for Biomedical Applications**, 106–115.

Kanungo, T., Mount, D. M., Netanyahu, N. S., Piatko, C. D., Silverman, R., & Wu, A. Y. (2002). An efficient k-means clustering algorithm: Analysis and implementation. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, 24(7), 881–892.

Kapil, S., & Chawla, M. (2016). Performance evaluation of k-means clustering algorithm with various distance metrics. **Proceedings of 2016 IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES)**, 1–4.

Kent, J. T. (1983). Information gain and a general measure of correlation. **Biometrika**, 70(1), 163–173.

Khalid, S., Khalil, T., & Nasreen, S. (2014). A survey of feature selection and feature extraction techniques in machine learning. **Preceeding of 2014 Science and Information Conference**, 372–378.

Kiatwanidvilai, S., & Praserttaweelap, R. (2018). Neurofuzzy c-Means Network-Based SCARA Robot for Head Gimbal Assembly (HGA) Circuit Inspection. **Computational Intelligence and Neuroscience**, 2018.

Köksal, G., Batmaz, I., & Testik, M. C. (2011). A review of data mining applications for quality improvement in manufacturing industry. **Expert Systems with Applications**, 38(10), 13448–13467.

Kotsiantis, S., Kanellopoulos, D., Pintelas, P., & others. (2006). Handling imbalanced datasets: A review. **GESTS International Transactions on Computer Science and Engineering**, 30(1), 25–36.

Krawczyk, B. (2016). Learning from imbalanced data: open challenges and future directions. **Progress in Artificial Intelligence**, 5(4), 221–232.

Kutner, M. H., Nachtsheim, C. J., Neter, J., Li, W., & others. (2005). Applied linear statistical models (Vol. 5). McGraw-Hill Irwin New York.

Lamichhaney, S., Berglund, J., Almén, M. S., Maqbool, K., Grabherr, M., Martinez-Barrio, A., Promerová, M., Rubin, C.-J., Wang, C., Zamani, N., & others. (2015). Evolution of Darwin's finches and their beaks revealed by genome sequencing. **Nature**, 518(7539), 371–375.

Le, Q. V. (2013). Building high-level features using large scale unsupervised learning. **Proceedings of 2013 IEEE International Conference on Acoustics, Speech and Signal Processing**, 8595–8598.

Lee, C., & Lee, G. G. (2006). Information gain and divergence-based feature selection for machine learning-based text categorization. **Information Processing & Management**, 42(1), 155–165.

Lee, C.-Y., & Tsai, T.-L. (2019). Data science framework for variable selection, metrology prediction, and process control in TFT-LCD manufacturing. **Robotics and Computer-Integrated Manufacturing**, 55, 76–87.

Lee, H., Kim, C. O., Ko, H. H., & Kim, M.-K. (2015). Yield prediction through the event sequence analysis of the die attach process. **IEEE Transactions on Semiconductor Manufacturing**, 28(4), 563–570.

Lee, H. K., & Kim, S. B. (2018). An overlap-sensitive margin classifier for imbalanced and overlapping data. **Expert Systems with Applications**, 98, 72–83.

Li, J., Ji, X., Jia, Y., Zhu, B., Wang, G., Li, Z., & Liu, X. (2014). Hard drive failure prediction using classification and regression trees. **Proceedings of 2014 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks**, 383–394.

Li, J., Stones, R. J., Wang, G., Liu, X., Li, Z., & Xu, M. (2017). Hard drive failure prediction using decision trees. **Reliability Engineering & System Safety**, 164, 55–65.

Liang, J., Bai, L., Dang, C., & Cao, F. (2012). The k-means-type algorithms versus imbalanced data distributions. **IEEE Transactions on Fuzzy Systems**, 20(4), 728–745.

Lin, W.-C., Tsai, C.-F., Hu, Y.-H., & Jhang, J.-S. (2017). Clustering-based undersampling in class-imbalanced data. **Information Sciences**, 409, 17–26.

Liu, H., & Zhou, M. (2017). Decision tree rule-based feature selection for large-scale imbalanced data. **Proceedings of Wireless and Optical Communication Conference (WOCC), 2017 26th**, 1–6.

Liu, X.-Y., Wu, J., & Zhou, Z.-H. (2008). Exploratory undersampling for class-imbalance learning. **IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)**, 39(2), 539–550.

Liu, X.-Y., & Zhou, Z.-H. (2006). The influence of class imbalance on cost-sensitive learning: An empirical study. **Proceeding of Data Mining, 2006. ICDM'06. Sixth International Conference On**, 970–974.

Loh, W.-Y. (2011). Classification and regression trees. **Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery**, 1(1), 14–23.

López, V., Fernández, A., Moreno-Torres, J. G., & Herrera, F. (2012). Analysis of preprocessing vs. cost-sensitive learning for imbalanced classification. Open problems on intrinsic data characteristics. **Expert Systems with Applications**, 39(7), 6585–6608.

Marsh, G. (2015). Darwin's iconic finches join genome club. **Nature News**, 518(7538), 147.

Martin-Diaz, I., Morinigo-Sotelo, D., Duque-Perez, O., & Romero-Troncoso, R. de J. (2016). Early fault detection in induction motors using AdaBoost with imbalanced small data and optimized sampling. **IEEE Transactions on Industry Applications**, 53(3), 3066–3075.

McHugh, M. L. (2013). The chi-square test of independence. **Biochemia Medica: Biochemia Medica**, 23(2), 143–149.

Meyer, D., & Wien, F. H. T. (2015). Support vector machines. **The Interface to Libsvm in Package E1071**, 28.

Moghaddam, V. H., & Hamidzadeh, J. (2016). New Hermite orthogonal polynomial kernel and combined kernels in support vector machine classifier. **Pattern Recognition**, 60, 921–935.

Montgomery, D. C., Peck, E. A., & Vining, G. G. (2012). Introduction to linear regression analysis (Vol. 821). John Wiley & Sons.

Myreaders. (2010). Genetic Algorithm (Online). Available: http://www.myreaders.info/09_Genetic_Algorithms.pdf [2020, November 8]

Nowozin, S. (2012). Improved information gain estimates for decision tree induction. **ArXiv Preprint ArXiv:1206.4620**.

Nwosu, H. U., Obieke, C. C., & Ameh, A. J. (2016). Failure analysis and shock protection of external hard disk drive. **Nigerian Journal of Technology**, 35(4), 855–865.

Peng, L., Zhang, H., Yang, B., & Chen, Y. (2014). A new approach for imbalanced data classification based on data gravitation. **Information Sciences**, 288, 347–373.

Podos, J., & Nowicki, S. (2004). Beaks, adaptation, and vocal evolution in Darwin's finches. **Bioscience**, 54(6), 501–510.

Powers, D. M. W. (2020). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. **ArXiv Preprint ArXiv:2010.16061**.

Quinlan, J. R. (1986). Induction of decision trees. **Machine Learning**, 1(1), 81–106.

Ramentol, E., Caballero, Y., Bello, R., & Herrera, F. (2012). SMOTE-RSB*: a hybrid preprocessing approach based on oversampling and undersampling for high imbalanced data-sets using SMOTE and rough sets theory. **Knowledge and Information Systems**, 33(2), 245–265.

Roobaert, D., Karakoulas, G., & Chawla, N. V. (2006). Information gain, correlation and support vector machines. In *Feature extraction* (pp. 463–470). Springer.

Ruangthong, P., & Jaiyen, S. (2016). Hybrid ensembles of decision trees and Bayesian network for class imbalance problem. **Proceedings of 2016 8th International Conference on Knowledge and Smart Technology (KST)**, 39–42.

Russell, S. J., & Norvig, P. (2016). Artificial intelligence: a modern approach. Malaysia. Pearson Education Limited.

Russell, S., & Norvig, P. (2002). *Artificial intelligence: a modern approach*.

Sadrawi, M., Sun, W.-Z., Ma, M. H.-M., Yeh, Y.-T., Abbod, M. F., & Shieh, J.-S. (2018). Ensemble genetic fuzzy neuro model applied for the emergency medical service via unbalanced data evaluation. **Symmetry**, 10(3), 71.

Safavian, S. R., & Landgrebe, D. (1991). A survey of decision tree classifier methodology. **IEEE Transactions on Systems, Man, and Cybernetics**, 21(3), 660–674.

Salimans, T., & Kingma, D. P. (2016). Weight normalization: A simple reparameterization to accelerate training of deep neural networks. **Proceedings of the 30th International Conference on Neural Information Processing Systems**, 901–909.

Samattapapong, N., & Afzulpurkar, N. (2016). A production throughput forecasting system in an automated hard disk drive test operation using GRNN. **Journal of Industrial Engineering and Management**, 9(2), 330–358.

Sankar, S., Shaw, M., Vaid, K., & Gurumurthi, S. (2013). Datacenter scale evaluation of the impact of temperature on hard disk drive failures. **ACM Transactions on Storage (TOS)**, 9(2), 1–24.

Satorra, A., & Bentler, P. M. (2001). A scaled difference chi-square test statistic for moment structure analysis. **Psychometrika**, 66(4), 507–514.

Schölkopf, B., Smola, A. J., Bach, F., & others. (2002). Learning with kernels: support vector machines, regularization, optimization, and beyond. MIT press.

Seber, G. A. F., & Lee, A. J. (2012). Linear regression analysis (Vol. 329). John Wiley & Sons.

Shannon, C. E. (2001). A mathematical theory of communication. **ACM SIGMOBILE Mobile Computing and Communications Review**, 5(1), 3–55.

Simon, T. R., Cong, L., Zhai, Y., Zhu, Y., & Zhao, F. (2018). A Semi-automatic System for Efficient Recovery of Rare Earth Permanent Magnets from Hard Disk Drives. **Procedia CIRP**, 69, 916–920.

Singh, A., Yadav, A., & Rana, A. (2013). K-means with Three different Distance Metrics. **International Journal of Computer Applications**, 67(10).

Song, W., Ovcharenko, A., Knigge, B., Yang, M., & Talke, F. E. (2012). Effect of contact conditions during thermo-mechanical contact between a thermal flying height control slider and a disk asperity. **Tribology International**, 55, 100–107.

Song, Y.-Y., & Ying, L. U. (2015). Decision tree methods: applications for classification and prediction. **Shanghai Archives of Psychiatry**, 27(2), 130.

Soons, J., Herrel, A., Genbrugge, A., Aerts, P., Podos, J., Adriaens, D., De Witte, Y., Jacobs, P., & Dirckx, J. (2010). Mechanical stress, fracture risk and beak evolution in Darwin's ground finches (Geospiza). **Philosophical Transactions of the Royal Society B: Biological Sciences**, 365(1543), 1093–1098.

Srivastava, A., Han, E.-H., Kumar, V., & Singh, V. (1999). Parallel formulations of decision-tree classification algorithms. In *High Performance Data Mining* (pp. 237–261). Springer.

Story, M., & Congalton, R. G. (1986). Accuracy assessment: a user's perspective. **Photogrammetric Engineering and Remote Sensing**, 52(3), 397–399.

Su, M.-C., & Chou, C.-H. (2001). A modified version of the K-means algorithm with a distance based on cluster symmetry. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, 23(6), 674–680.

Sun, Y., Kamel, M. S., Wong, A. K. C., & Wang, Y. (2007). Cost-sensitive boosting for classification of imbalanced data. **Pattern Recognition**, 40(12), 3358–3378.

Sun, Y., Wong, A. K. C., & Kamel, M. S. (2009). Classification of imbalanced data: A review. **International Journal of Pattern Recognition and Artificial Intelligence**, 23(04), 687–719.

Sun, Z., Song, Q., Zhu, X., Sun, H., Xu, B., & Zhou, Y. (2015). A novel ensemble method for classifying imbalanced data. **Pattern Recognition**, 48(5), 1623–1637.

Suppers, A., Gool, A. J. van, & Wessels, H. J. C. T. (2018). Integrated chemometrics and statistics to drive successful proteomics biomarker discovery. **Proteomes**, 6(2), 20.

Suykens, J. A. K., & Vandewalle, J. (1999). Least squares support vector machine classifiers. **Neural Processing Letters**, 9(3), 293–300.

Taetragool, U., & Achalakul, T. (2011). Method for failure pattern analysis in disk drive manufacturing. **International Journal of Computer Integrated Manufacturing**, 24(9), 834–846.

Tang, J., Alelyani, S., & Liu, H. (2014). Data classification: algorithms and applications. **Data Mining and Knowledge Discovery Series, CRC Press**, 37–64.

Thai-Nghe, N., Gantner, Z., & Schmidt-Thieme, L. (2010). Cost-sensitive learning methods for imbalanced data. **Proceeding of The 2010 International Joint Conference on Neural Networks (IJCNN)**, 1–8.

Thakare, Y. S., & Bagal, S. B. (2015). Performance evaluation of K-means clustering algorithm with various distance metrics. **International Journal of Computer Applications**, 110(11), 12–16.

Thaseen, I. S., & Kumar, C. A. (2017). Intrusion detection model using fusion of chi-square feature selection and multi class SVM. **Journal of King Saud University-Computer and Information Sciences**, 29(4), 462–472.

Tong, S., & Koller, D. (2001). Support vector machine active learning with applications to text classification. **Journal of Machine Learning Research**, 2(Nov), 45–66.

Turabieh, H., Mafarja, M., & Li, X. (2019). Iterated feature selection algorithms with layered recurrent neural network for software fault prediction. **Expert Systems with Applications**, 122, 27–42.

Visa, S., Ramsay, B., Ralescu, A. L., & Van Der Knaap, E. (2011). Confusion Matrix-based Feature Selection. **MAICS**, 710, 120–127.

Weiss, G. M., McCarthy, K., & Zabar, B. (2007). Cost-sensitive learning vs. sampling: Which is best for handling unbalanced classes with unequal error costs? **DMIN**, 7, 35–41.

Willmott, C. J., & Matsuura, K. (2005). Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. **Climate Research**, 30(1), 79–82.

Wright, A. H. (1991). Genetic algorithms for real parameter optimization. In *Foundations of genetic algorithms* (Vol. 1, pp. 205–218). Elsevier.

Wu, J. (2012). Advances in K-means clustering: a data mining thinking. Springer Science & Business Media.

Xu, H., Zhang, J., Lv, Y., & Zheng, P. (2020). Hybrid Feature Selection for Wafer Acceptance Test Parameters in Semiconductor Manufacturing. **IEEE Access**, 8, 17320–17330.

Yaeger, L. (2008). Intro to Genetic Algorithms. **Artificial Life as an Approach to Artificial Intelligence Professor of Informatics, Indiana University**.

Yang, Y. Y., Mahfouf, M., Panoutsos, G., Zhang, Q., & Thornton, S. (2011). Adaptive neural-fuzzy inference system for classification of rail quality data with bootstrapping-based over-sampling. **Proceeding of 2011 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2011)**, 2205–2212.

Yao, X. (1999). Evolving artificial neural networks. **Proceedings of the IEEE**, 87(9), 1423–1447.

Ye, Z.-S., Xie, M., & Tang, L.-C. (2013). Reliability evaluation of hard disk drive failures based on counting processes. **Reliability Engineering & System Safety**, 109, 110–118.

Yuan, T., Ramadan, S. Z., & Bae, S. J. (2011). Yield prediction for integrated circuits manufacturing through hierarchical Bayesian modeling of spatial defects. **IEEE Transactions on Reliability**, 60(4), 729–741.

Zhang, G., Patuwo, B. E., & Hu, M. Y. (1998). Forecasting with artificial neural networks:: The state of the art. **International Journal of Forecasting**, 14(1), 35–62.

Zhang, M.-L., & Zhou, Z.-H. (2005). A k-nearest neighbor based algorithm for multi-label classification. **Proceeding of 2005 IEEE International Conference on Granular Computing**, 2, 718–721.

Zhang, X., Li, Y., Kotagiri, R., Wu, L., Tari, Z., & Cheriet, M. (2017). KRNN: k Rare-class Nearest Neighbour classification. **Pattern Recognition**, 62, 33–44.

Zhou, H., Yu, K.-M., Chen, Y.-C., & Hsu, H.-P. (2021). A Hybrid Feature Selection Method RFSTL for Manufacturing Quality Prediction Based on a High Dimensional Imbalanced Dataset. **IEEE Access**, 9, 29719–29735.

Zhu, A. (2017). Artificial Neural Networks. The International Encyclopedia of Geography. **Wiley. Azuma, R., Baillot, Y., Behringer, R., Feiner, S., Julier, S., & MacIntyre, B.(2001). Recent Advances in Augmented Reality. IEEE Computer Graphics and Applications**, 21(6), 34–47.

Zong, W., Huang, G.-B., & Chen, Y. (2013). Weighted extreme learning machine for imbalance learning. **Neurocomputing**, 101, 229–242.

ภาคผนวก ก

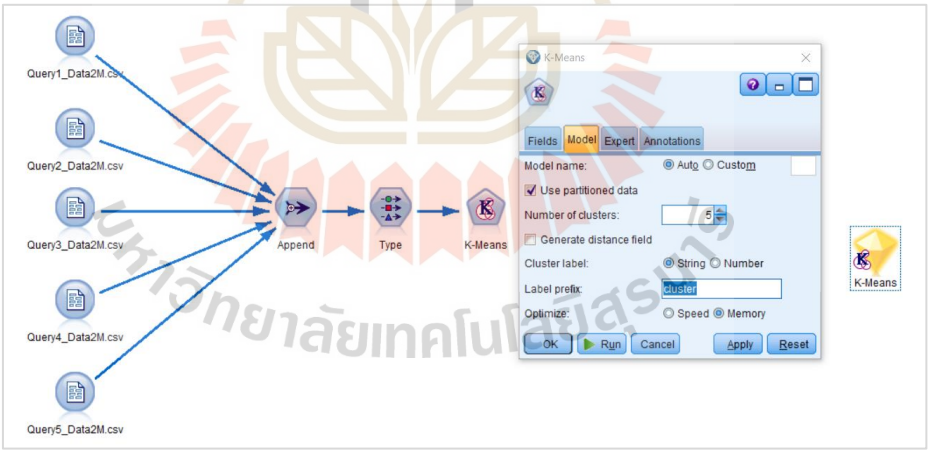
การใช้งานโปรแกรมและแสดงรหัสต้นฉบับ

การใช้งานโปรแกรมและแสดงรหัสต้นฉบับ

เนื้อหาส่วนนี้แสดงการใช้งานโปรแกรม IBM SPSS Modeler และ Microsoft Excel เพื่อใช้ในการทำสมดุลข้อมูล และแสดงรหัสต้นฉบับในการใช้ภาษา R ของทั้ง 7 วิธีการที่ใช้สำหรับการคัดเลือกคุณลักษณะ รวมไปจนถึงการคาดการณ์ผลลัพธ์ด้วยอัลกอริทึม MLR และ ANN ซึ่งมีรายละเอียดดังต่อไปนี้

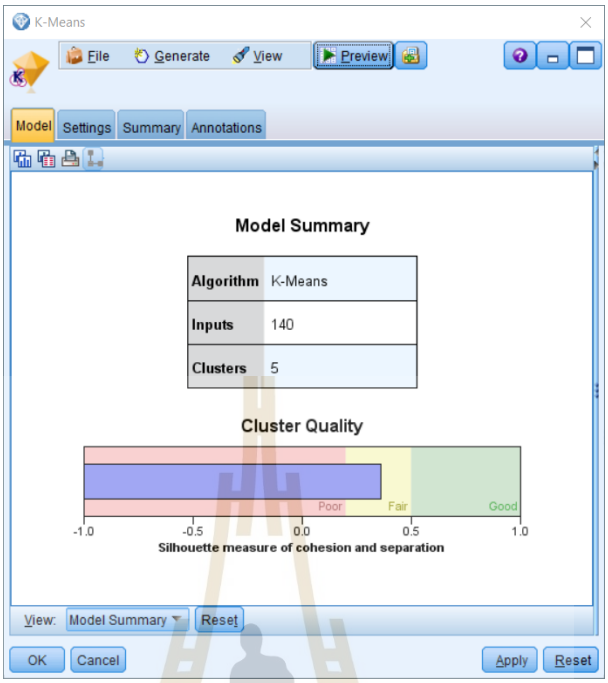
1. การทำสมดุลข้อมูลด้วยวิธี DBC-2KAR (Data Balancing by k-Means Clustering k-NN and Re-sampling)

การนำข้อมูลเข้าสู่โปรแกรม IBM SPSS Modeler ผู้วิจัยเลือกใช้ไฟล์ .csv จำนวน 5 ไฟล์ โดยในแต่ละไฟล์มีจำนวนข้อมูล 2,000,000 แถว จากนั้นจึงใช้โหนด Append ในการรวมข้อมูลเพื่อให้ได้ข้อมูลจำนวน 10,000,000 แถว หลังจากนั้นใช้โหนด Type เพื่อกำหนดค่าและเช็คค่าเบื้องต้นของคุณลักษณะต่าง ๆ ก่อนที่จะเข้าสู่โหนด K-Means (อัลกอริทึม k-Mean Clustering) ซึ่งผู้วิจัยกำหนดจำนวนของคลัสเตอร์ในโหนดนี้ให้มีค่าเท่ากับ 5 เมื่อทำการประมวลผลโปรแกรมจะได้ผลลัพธ์เป็นรูปเพชรสีทอง ซึ่งรายละเอียดที่กล่าวมาข้างต้นแสดงไว้ดังรูปที่ ก.1

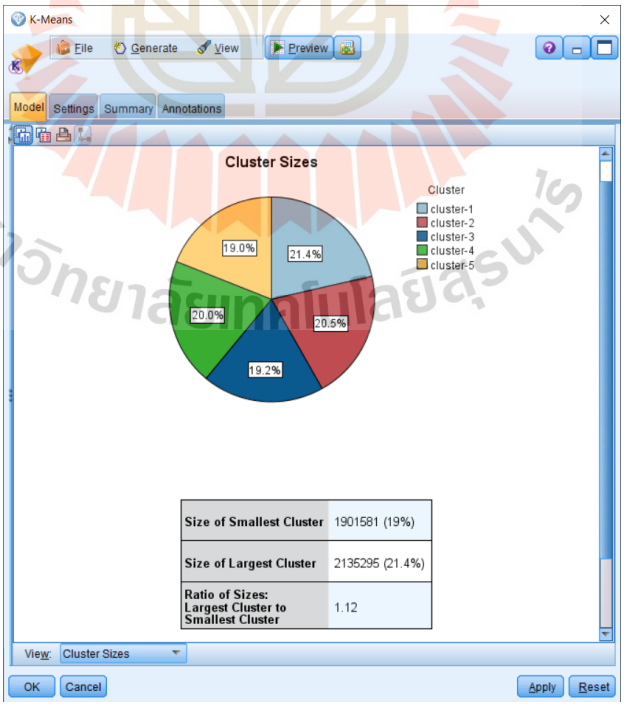


รูปที่ ก.1 การนำเข้าข้อมูลเพื่อใช้อัลกอริทึม K-Means ในโปรแกรม IBM SPSS Modeler

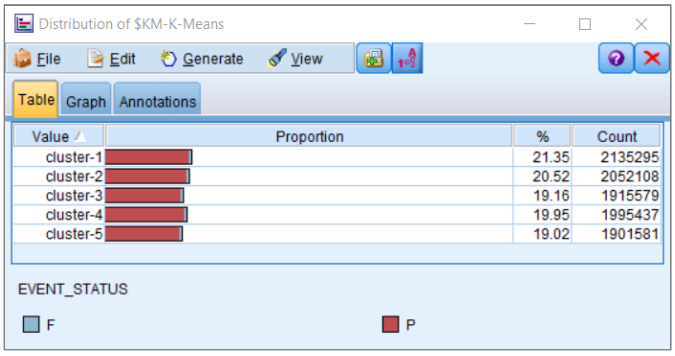
เมื่อพิจารณาผลลัพธ์ที่ได้จากอัลกอริทึม k-Means Clustering พบว่ามีคุณลักษณะที่อัลกอริทึมนี้นำมาพิจารณา 140 คุณลักษณะและถือว่าผลลัพธ์ในการจัดกลุ่มอยู่ในเกณฑ์พอใช้ (Fair) ซึ่งแสดงดังรูปที่ ก.2 และในส่วนของจำนวนข้อมูลในแต่คลัสเตอร์แสดงดังรูปที่ ก.3 และ ก.4 ตามลำดับ



รูปที่ ก.2 แสดงผลลัพธ์ภาพรวมจาก k-Means Clustering

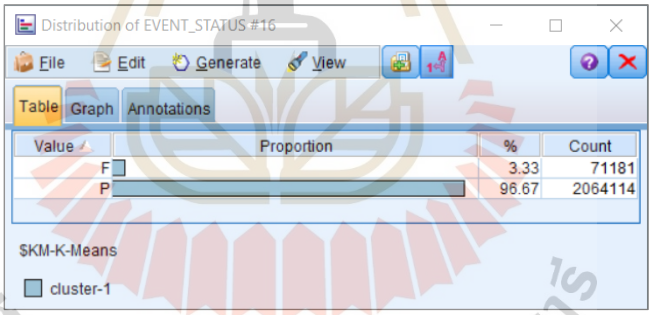


รูปที่ ก.3 ผลลัพธ์จาก k-Means Clustering แสดงเปอร์เซนต์ข้อมูลในแต่คลัสเตอร์

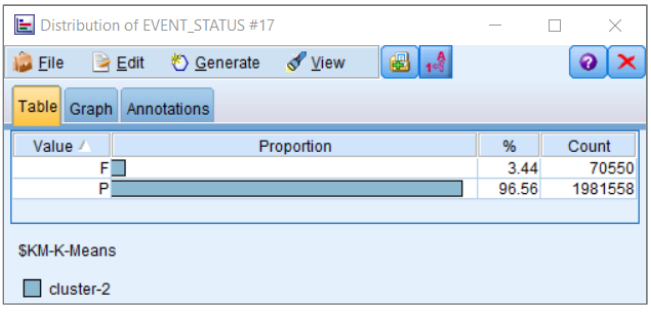


รูปที่ ก.4 แสดงจำนวนข้อมูลในแต่คลัสเตอร์ ซึ่งได้มาจาก k-Means Clustering

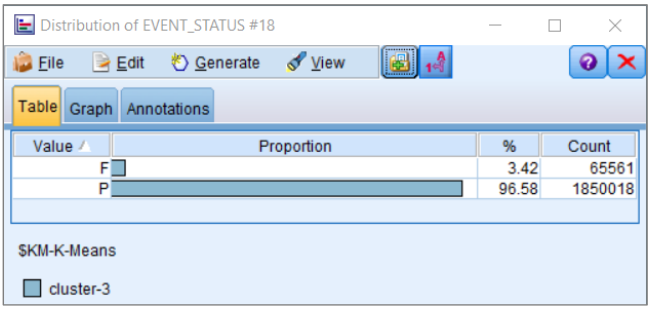
เมื่อพิจารณาในแต่ละคลัสเตอร์จะพบว่าทุก ๆ คลัสเตอร์มีลักษณะสัดส่วนข้อมูลเป็นไปในทิศทางเดียวกันนั่นคือมีจำนวน Passed unit มากกว่าจำนวนของ Failed unit ซึ่งแสดงรายละเอียดไว้ดังรูปที่ ก.5, ก.6, ก.7, ก.8 และ ก.9



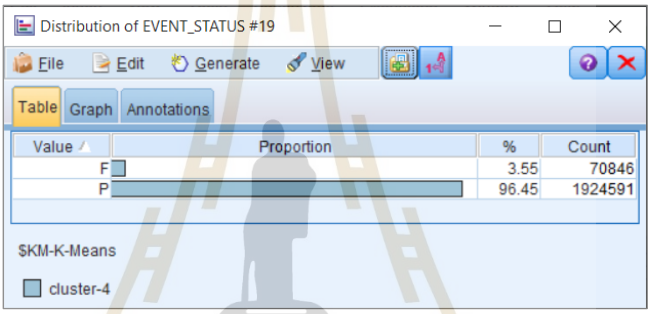
รูปที่ ก.5 แสดงสัดส่วนข้อมูลของคลัสเตอร์ที่ 1



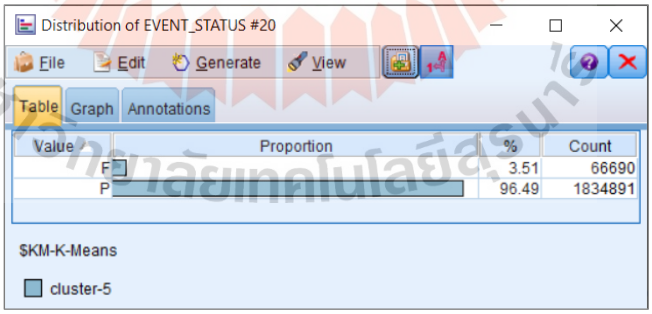
รูปที่ ก.6 แสดงสัดส่วนข้อมูลของคลัสเตอร์ที่ 2



รูปที่ ก.6 แสดงสัดส่วนข้อมูลของคลัสเตอร์ที่ 3



รูปที่ ก.6 แสดงสัดส่วนข้อมูลของคลัสเตอร์ที่ 4



รูปที่ ก.6 แสดงสัดส่วนข้อมูลของคลัสเตอร์ที่ 5

เมื่อทั้ง 5 คลัสเตอร์มีข้อมูล Passed unit มากกว่าจำนวนของ Failed unit ซึ่งถือว่าอยู่ในการจัดการข้อมูลตามกรณีที่ 1 (ตารางที่ 3.1 หน้า 85) โดยกำหนดว่าจำนวน Failed unit ให้คงไว้เพราะถือเป็นคลาสน้อย ให้ไปพิจารณาเพื่อลดจำนวนข้อมูลในคลาสส่วนมากแทน

โดยจะเริ่มจากทำการสุ่มเลือกข้อมูล Passed unit มาหนึ่งข้อมูล และหาข้อมูลที่ใกล้ที่สุดมาเท่ากับจำนวนของ Failed unit – 1 ซึ่งตัวอย่างการคำนวณแสดงดังรูปที่ ก.7 และ ก.8

DRIVE SN	SN0000001	SN0000002	SN0000003	SN0000004	SN0000005	SN0000006	SN0000007	SN0000008	SN0000009	SN0000010	SN0000011
EVENT_STATUS	P	P	P	P	P	P	P	P	P	P	P
DRV_COMP_TRK	PPPPP	PPPPP	PPPRP	PPPPP	PPPPP	RRRPP	PPPPP	RRRRR	PRRRP	PPPPP	RRRRP
DRV6	P	P	P	P	P	P	P	O	P	P	P
HGSA_Prime	Y	Y	Y	Y	Y	N	Y	N	Y	Y	N
HGSA_RCY	N	N	N	N	N	N	N	N	N	N	N
HGSA_RWK	N	N	N	N	N	N	N	N	N	N	N
Media_Prime	Y	Y	Y	Y	Y	N	Y	N	N	Y	N
Media_RCY	N	N	N	N	N	Y	N	Y	Y	N	Y
Media_RWK	N	N	N	N	N	N	N	N	N	N	N
LVCM	Y	Y	N	Y	Y	Y	Y	N	N	Y	N
UVCM	Y	Y	N	Y	Y	Y	Y	N	N	Y	N
BSNS_SEGMENT	NL2	NL2	NL3	NL3	NL3	NL3	NL2	NL3	NL2	NL2	NL2
CERT_TEMP	55	53	55	54	54	54	56	54	56	54	54
CLAMP_VEN_ID	U	E	E	U	U	U	E	0	E	E	E
CRX_CNT	1	1	1	1	1	1	1	2	1	1	1
LINE_NUM	108	108	108	606	108	606	606	113	606	606	606
MTR_CYCLE_CNT	1	1	1	1	1	3	1	2	2	1	2
PRIME	Y	Y	N	N	Y	N	Y	N	N	Y	N
PWALINE	L606	108BPS7	L606	L606	113BPSA	113BPSA	L606	L606	113BPSA	L606	L606
TOP_COVER_VENDO	31	31	31	31	31	21	21	31	31	21	31
BUILD_GROUP	SLV	NEW	SLV	SLV	SLV	SLV	MRW	SLV	NEW	MRW	SLV
CELL_ID	108	108	108	606	108	606	606	113	606	606	606
RISER_TYPE	DDSAS	DDSATA	DDSAS	DDSAS	DDSAS	DDSAS	DDSATA	DDSAS	DDSAS	DDSAS	DDSAS
SHIFT_NUM	2	2	2	1	2	2	1	3	2	2	2
HD_STACK_ID	HSI081	HSI081	HSI081	HSI661	HSI081	HSI661	HSI661	HSI131	HSI661	HSI661	HSI661
HSA_CODE	E91	E91	E91	Q91	E91	Q91	E91	E91	Q92	E91	Q91
HSA_COH	OXCXXXS	OXCXXXS	OXCMGXS	OXCXXXS	OXCXXXS	OXCXXXS	OXCXXXS	OXCXXXS	OXCXXXS	OXCXXXS	OXCXXXS
HSA_FW	20	20	20	21	22	22	22	23	23	23	24
HSA_PART_NUM	100833699	100833699	100833699	100833699	100833699	100833699	100845713	100833699	100833699	100833699	100833699
DISTANCE	0	5	7	9	4	15	10	18	16	8	15

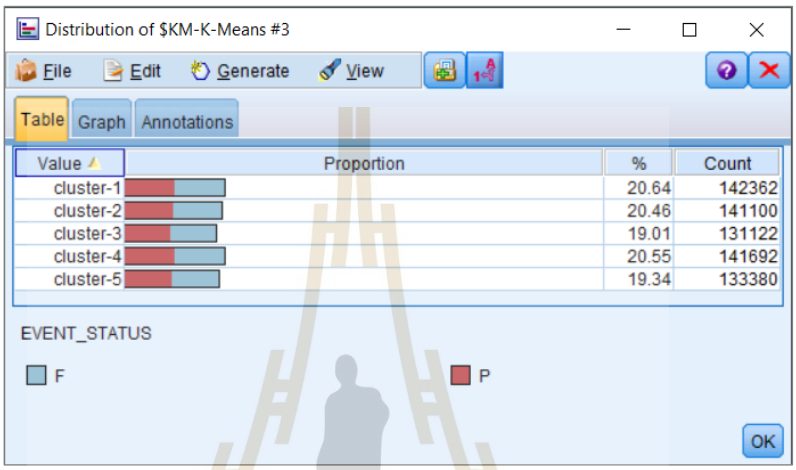
รูปที่ ก.7 ตัวอย่างข้อมูล Passed unit ซึ่งแสดงระยะห่างจากข้อมูล SN0000001 ซึ่งถูกสุ่มเลือก

DRIVE SN	SN0000001	SN0000002	SN0000003	SN0000004	SN0000005	SN0000006	SN0000007	SN0000008	SN0000009	SN0000010	SN0000011
CALCULATION											
DRV_COMP_TRK	0	0	0	0	0	0	0	0	0	0	0
DRV6	0	0	1	0	0	1	0	1	1	0	1
HGSA_Prime	0	0	0	0	0	0	0	1	0	0	0
HGSA_RCY	0	0	0	0	0	1	0	1	0	0	1
HGSA_RWK	0	0	0	0	0	0	0	0	0	0	0
Media_Prime	0	0	0	0	0	0	0	0	0	0	0
Media_RCY	0	0	0	0	0	1	0	1	1	0	1
Media_RWK	0	0	0	0	0	1	0	1	1	0	1
LVCM	0	0	0	0	0	0	0	0	0	0	0
UVCM	0	0	1	0	0	0	0	1	1	0	1
BSNS_SEGMENT	0	0	1	0	0	0	0	1	1	0	1
CERT_TEMP	0	0	1	1	1	1	0	1	0	0	0
CLAMP_VEN_ID	0	1	0	1	1	1	1	1	1	1	1
CRX_CNT	0	1	1	0	0	0	1	1	1	1	1
LINE_NUM	0	0	0	0	0	0	0	1	0	0	0
MTR_CYCLE_CNT	0	0	0	1	0	1	1	1	1	1	1
PRIME	0	0	0	0	0	1	0	1	1	0	1
PWALINE	0	0	1	1	0	1	0	1	1	0	1
TOP_COVER_VENDO	0	1	0	0	1	1	0	0	1	0	0
BUILD_GROUP	0	0	0	0	0	1	1	0	0	1	0
CELL_ID	0	1	0	0	0	0	1	0	1	1	0
RISER_TYPE	0	0	0	1	0	1	1	1	1	1	1
SHIFT_NUM	0	1	0	0	0	0	1	0	0	0	0
HD_STACK_ID	0	0	0	1	0	0	1	1	0	0	0
HSA_CODE	0	0	0	1	0	1	1	1	1	1	1
HSA_COH	0	0	0	1	0	1	0	0	1	0	1
HSA_FW	0	0	1	0	0	0	0	0	0	0	0
HSA_PART_NUM	0	0	0	1	1	1	1	1	1	1	1
DISTANCE	0	5	7	9	4	15	10	18	16	8	15

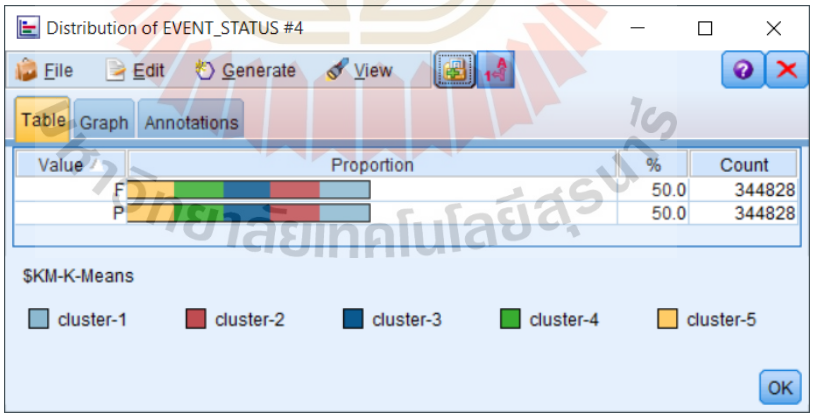
รูปที่ ก.8 แสดงการคำนวณระยะห่างของข้อมูล SN0000001 ซึ่งถูกสุ่มเลือก

หากข้อมูลเหมือนกันระยะห่างเป็น 0 ต่างกันระยะห่างเป็น 1

หลังจากดำเนินการตามวิธี DBC-2KAR ครบถ้วน จะทำให้ได้ข้อมูลในทุกคลัสเตอร์ที่สมดุลกันดังรูปที่ ก.9 และ ก.10 ซึ่งพร้อมสำหรับกระบวนการคัดเลือกคุณลักษณะในขั้นตอนถัดไป



รูปที่ ก.9 แสดงข้อมูลในทุกคลัสเตอร์หลังจากผ่านวิธีการ DBC-2KAR



รูปที่ ก.10 แสดงข้อมูลระหว่าง Failed unit และ Passed unit ซึ่งนำทุกคลัสเตอร์มารวมกัน

2. แสดงรหัสต้นฉบับการใช้ภาษา R ของ 7 วิธีการที่ใช้ในการคัดเลือกคุณลักษณะ

ในส่วนนี้จะเป็นการแสดงรหัสต้นฉบับการใช้ภาษา R ของทั้ง 7 วิธีการที่ใช้ในการคัดเลือกคุณลักษณะ ได้แก่ C5, CART, SVM, Stepwise, GA, Information Gain และ Chi-Square

```
install.packages("FSelector")

library(tidyverse)
library(caret)
library(doParallel)
library(FSelector)
library(mlbench)

# Install packages if missing
list.of.packages <- c("parallel", "doParallel", "caret", "randomForest", "funModeling", "tidyverse", "GA")
new.packages <- list.of.packages[!(list.of.packages %in% installed.packages()[,"Package"])]
if(length(new.packages)) install.packages(new.packages)

# Load libraries
library(randomForest)
library(funModeling)
library(GA)
library(dplyr)

list.files(path = "../input")
```

Feature Selection with C5.0

```
MyData <- read.csv(file="../input/research/data.csv", header=TRUE, sep=",")

set.seed(100)

cl <- makePSOCKcluster(4)
registerDoParallel(cl)

start_time <- Sys.time()

c5Mod <- train(Class ~ ., data=MyData, method="C5.0")

end_time <- Sys.time()

stopCluster(cl)

c5Imp <- varImp(c5Mod)
print(c5Imp)

plot(c5Imp, top = 20, main='Variable Importance')

exctime <- end_time-start_time
print(exctime)
```

Feature Selection with CART

```
MyData <- read.csv(file="../input/research/data.csv", header=TRUE, sep=",")

set.seed(100)

cl <- makePSOCKcluster(4)
registerDoParallel(cl)

start_time <- Sys.time()
cartMod <- train(Class ~ ., data=MyData, method="rpart")
end_time <- Sys.time()

stopCluster(cl)

cartImp <- varImp(cartMod)
print(cartImp)

plot(cartImp, top = 20, main='Variable Importance')

exctime <- end_time-start_time
print(exctime)
```

Feature Selection with SVM

```
MyData <- read.csv(file="../input/research/data.csv", header=TRUE, sep=",")

set.seed(100)

start_time <- Sys.time()

cl <- makePSOCKcluster(10)
registerDoParallel(cl)

svmMod <- train(Class~., data=MyData, method="svmRadial", metric="Accuracy")

stopCluster(cl)

end_time <- Sys.time()

svmImp <- varImp(svmMod)
print(svmImp)

end_time-start_time

plot(svmImp, top = 20, main='Variable Importance')
```

Feature Selection with Stepwise

```
trainData <- read.csv(file="../input/researcht1/data.csv", stringsAsFactors=F)

print(head(trainData))

# Step 1: Define base intercept only model
base.mod <- lm(Class ~ 1 , data=trainData)

# Step 2: Full model with all predictors
all.mod <- lm(Class ~ . , data= trainData)

start_time <- Sys.time()

cl <- makePSOCKcluster(10)
registerDoParallel(cl)

# Step 3: Perform step-wise algorithm. direction='both' implies both forward and backward stepwise
stepMod <- step(base.mod, scope = list(lower = base.mod, upper = all.mod), direction = "both", trace = 0, steps = 1000)

stopCluster(cl)

end_time <- Sys.time()

# Step 4: Get the shortlisted variable.
shortlistedVars <- names(unlist(stepMod[[1]]))
shortlistedVars <- shortlistedVars[!shortlistedVars %in% "(Intercept)"] # remove intercept

# Show
print(shortlistedVars)


exctime <- end_time-start_time
print(exctime)

y_pred = predict(stepMod, newdata = trainData)
```

Feature Selection with Chi-square

```
MyData <- read.csv(file="../input/research/data.csv", header=TRUE, sep=",")

start_time <- Sys.time()

weights <- chi.squared(Class~., MyData)

exctime <- end_time-start_time

print(weights)

print(exctime)
```

Feature Selection with Information Gain

```
MyData <- read.csv(file="../input/research/data.csv", header=TRUE, sep=",")

start_time <- Sys.time()

weights <- information.gain(Class~., MyData)

exctime <- end_time-start_time
print(exctime)

print(weights)
```

Feature Selection with GA

```
data <- read.csv(file="../input/research/data.csv", header=TRUE, sep=",")

# Install packages if missing
list.of.packages <- c("parallel", "doParallel", "caret", "randomForest", "funModeling", "tidyverse", "GA")
new.packages <- list.of.packages[!(list.of.packages %in% installed.packages()[,"Package"])]
if(length(new.packages)) install.packages(new.packages)

# Load libraries
library(caret)
library(randomForest)
library(funModeling)
library(tidyverse)
library(GA)
library(dplyr)

custom_fitness <- function(vars, data_x, data_y, p_sampling)
{
  # speeding up things with sampling
  ix=get_sample(data_x, percentage_tr_rows = p_sampling)
  data_2=data_x[ix,]
  data_y_smp=data_y[ix]

  # keep only vars from current solution
  names=colnames(data_2)
  names_2=names[vars==1]
  # get the columns of the current solution
  data_sol=data_2[, names_2]
  # get the roc value from the created model
  roc_value=get_roc_metric(data_sol, data_y_smp, names_2)

  # get the total number of vars for the current selection
  q_vars=sum(vars)

  # time for your magic
  fitness_value=roc_value/q_vars

  return(fitness_value)
}

get_roc_metric <- function(data_tr_sample, target, best_vars)
{
  # data_tr_sample=data_sol
  # target = target_var_s
  # best_vars=names_2

  fitControl <- trainControl(method = "cv",
                             number = 3,
                             summaryFunction = twoClassSummary,
                             classProbs = TRUE)

  data_model=select(data_tr_sample, one_of(best_vars))

  mtry = sqrt(ncol(data_model))
  tunegrid = expand.grid(.mtry=round(mtry))

  fit_model_1 = train(x=data_model,
                      y= target,
                      method = "rf",
                      trControl = fitControl,
                      metric = "ROC",
                      tuneGrid=tunegrid
                      )
  metric=fit_model_1$results["ROC"][1,1]

  return(metric)
}



get_accuracy_metric <- function(data_tr_sample, target, best_vars)
{
  data_model=select(data_tr_sample, one_of(best_vars))

  fitControl <- trainControl(method = "cv",
                             number = 3,
                             summaryFunction = twoClassSummary)

  data_model=select(data_tr_sample, one_of(best_vars))

  mtry = sqrt(ncol(data_model))
  tunegrid = expand.grid(mtry=round(mtry))

  fit_model_1 = train(x=data_model,
                      y= target,
                      method = "rf",
                      tuneGrid = tunegrid)



  metric=fit_model_1$results["Accuracy"][1,1]
  return(metric)
}
```

Feature Selection with GA

```
get_accuracy_metric <- function(data_tr_sample, target, best_vars)
{
  data_model=select(data_tr_sample, one_of(best_vars))

  fitControl <- trainControl(method = "cv",
                             number = 3,
                             summaryFunction = twoClassSummary)

  data_model=select(data_tr_sample, one_of(best_vars))

  mtry = sqrt(ncol(data_model))
  tunegrid = expand.grid(mtry=round(mtry))

  fit_model_1 = train(x=data_model,
                      y= target,
                      method = "rf",
                      tuneGrid = tunegrid)



  metric=fit_model_1$results["Accuracy"][1,1]
  return(metric)
}


# Data preparation
data2=na.omit(data) # <- use with care...

data_y=as.factor(data2$Class)
data_xx=select(data2,-Class)
data_x=as.data.frame(data_xx)

# GA parameters
param_nBits=ncol(data_x)
col_names=colnames(data_x)

# Executing the GA
# Executing the GA
start_time <- Sys.time()

ga_GA_1 = ga(fitness = function(vars) custom_fitness(vars = vars,
                                                     data_x =  data_x,
                                                     data_y = data_y,
                                                     p_sampling = 0.7), # custom fitness function
             type = "binary", # optimization data type
             crossover=gabin_uCrossover,  # cross-over method
             elitism = 3, # number of best ind. to pass to next iteration
             pmutation = 0.03, # mutation rate prob
             popSize = 50, # the number of indivduals/solutions
             nBits = param_nBits, # total number of variables
             names=col_names, # variable name
             run=5, # max iter without improvement (stopping criteria)
             maxiter = 50, # total runs or generations
             monitor=plot, # plot the result at each iteration
             keepBest = TRUE, # keep the best solution at the end
             parallel = T, # allow parallel procesing
             seed=84211 # for reproducibility purposes
)


# Checking the results
summary(ga_GA_1)

# Following line will return the variable names of the final and best solution
best_vars_ga=col_names[ga_GA_1@solution[1,]==1]

# Checking the variables of the best solution...
best_vars_ga

# Checking the accuracy
get_accuracy_metric(data_tr_sample = data_x, target = data_y, best_vars_ga)

end_time <- Sys.time()

exctime <- end_time-start_time
print(exctime)
```

3. แสดงรหัสต้นฉบับการใช้ภาษา R เพื่อการคาดการณ์ผลลัพธ์

ในส่วนนี้จะเป็นการแสดงรหัสต้นฉบับการใช้ภาษา R ในการคาดการณ์ผลลัพธ์ด้วยอัลกอริทึม MLR และ ANN

MLR

```
library(tidyverse)
library(caTools)
library(neuralnet)
library(keras)

set.seed(123)

list.files(path = "../input")

trainData <- read.csv(file="../input/research2/C5_training.csv", header=TRUE, sep=",")
testData <- read.csv(file="../input/research2/C5_testing.csv", header=TRUE, sep=",")

trainData <- trainData[,2:31]
testData <- testData[,2:31]

#trainData <- read.csv(file="../input/research2-cart/CART_training.csv", header=TRUE, sep=",")
#testData <- read.csv(file="../input/research2-cart/CART_testing.csv", header=TRUE, sep=",")

#trainData <- trainData[,2:40]
#testData <- testData[,2:40]

# Function that returns Root Mean Squared Error
rmse <- function(error)
{
    sqrt(mean(error^2))
}

# Function that returns Mean Absolute Error
mae <- function(error)
{
    mean(abs(error))
}

regressor = lm(formula = YIELD ~ .,
               data = trainData)

# Predicting the Test set results
y_pred = predict(regressor,testData)

# Calculate error
error <- c(testData[,1]) - c(y_pred)

# Example of invocation of functions
rmse(error)
mae(error)
```

ANN

```
n <- names(trainData)
f <- as.formula(paste("YIELD ~", paste(n[!n %in% "YIELD"], collapse = " + ")))
nn <- neuralnet(f,trainData,hidden=c(5,3),linear.output=T)

y_pred_nn = predict(nn, testData[,2:31])

# Calculate error
error <- c(testData[,1]) - c(y_pred_nn)

# Example of invocation of functions
rmse(error)
mae(error)
```

ภาคผนวก ข

บทความวิจัยที่ตีพิมพ์ระหว่างศึกษา

รายชื่อบทความวิชาการที่ได้รับการตีพิมพ์ในระหว่างศึกษา

Hirunyawanakul, A., Kerdprasop, N., & Kerdprasop, K. (2018). A novel heuristic method for misclassification cost tuning in imbalanced data. **International Journal of Machine Learning and Computing,** 8(6), 565–570.

Hirunyawanakul, A., Kerdprasop, N., & Kerdprasop, K. (2019). Misclassification cost tuning for root cause analysis on hard disk drive manufacturing. In H.A. Sulaiman et al. (Eds.), **Lecture Notes in Computer, Communication and Control Technology,** pp. 316-321, Malaysia Technical Scientist Association. (eISBN: 978-967-2348-08-5).

Hirunyawanakul, A., Kerdprasop, N., & Kerdprasop, K. (2020). Efficient machine learning methods for hard disk drive yield prediction improvement. **International Journal of Machine Learning and Computing,** 10(2), 240–246.

Hirunyawanakul, A., Kaoungku, N., Kerdprasop, N., & Kerdprasop, K. (2020). Feature selection to improve performance of yield prediction in hard disk drive manufacturing. **International Journal of Electrical and Electronic Engineering & Telecommunication,** 9(6), 420-428.

International Journal of Machine Learning and Computing, Vol. 8, No. 6, December 2018

A Novel Heuristic Method for Misclassification Cost Tuning in Imbalanced Data

Anusara Hirunyawanakul, Nittaya Kerdprasop, and Kittisak Kerdprasop

***Abstract*—Currently, one of the most challenging problem in machine learning and data mining is the data imbalance problem. Many techniques and methods are researched and proposed to solve this problem. Fundamental solution is data balancing with under-sampling and over-sampling techniques. However, these conventional methods might be suffered from the potential loss of useful information leading to the generation of useless patterns. Therefore, the techniques that avoid adjusting the sample size of data are more interesting. One of such technique is misclassification cost adjustment. This paper focuses on improving the performance of classification model built from the misclassification cost adjustment technique by proposing the novel heuristic method. Our proposed method uses a heuristic based on the experience of practitioner working on many manufacturing data. The heuristic employs the relation between misclassification cost, imbalance ratio and a constant factor "e" (Euler's number). The experiment has been operated on 56 real-world datasets with various number of attributes and different degrees of imbalance ratio. The results confirm that our novel heuristic method can help improving the performance of the classification model. On datasets with high imbalance ratio, our method shows the improvement rate of AUC up to 29%.**

***Index Terms*—Misclassification cost, imbalance data, classification, decision tree learning.**

I. Introduction

Data mining and machine learning are very popular and extensively used in several areas. The problem that has been reported as one of the most often found in this field is the imbalance ratio problem. Class imbalance data problem has been reported to occur in a wide variety of real world domains, such as facial age approximation [1], detecting oil spills from satellite images [2], anomaly detection [3], fraudulent credit card transactions detection [4], software error prediction [5], and pattern recognition on image annotation [6].

Most traditional algorithms, such as decision trees [7]–[9], k-nearest neighbors [10], [11], focus on generating the models that provide the highest overall accuracy and the minority data is always ignored [12]–[14]. However, in some cases the minority class instances may have so high significance and importance that they should not be ignored by the classification algorithms. Thus, data-preprocessing steps for balancing instances between classes are needed. One of the most popular methods for class rebalancing is data sampling [15]–[18]. However, under-sampling may eliminate the important data of the majority class. While over-sampling methods may alter the original class distribution. Moreover, increasing the minority class instances may generate the useless data and misleading the classification result. The cost-sensitive learning or misclassification cost adjustment seems to be the efficient way to solve the class imbalance problems [19]–[21].

The technique that we discovered in one field may show the good result in other fields and this paper is one of them. The technique that we introduce in this paper is extracted from the experience of researchers while had been working in the manufacturing companies and already proved with the datasets which are collected from production line database of Computer's component manufacturing. This method can help the expertise engineers to achieve the optimal of "true positive rate" in a shorter time.

This paper is used that novel method to apply on worldwide 56 datasets with the various fields like citizen data, wine quality data, card game data, medical/scientific experimental data etc. With these datasets, we separate them into 2 groups: the low imbalance ratio group and the high imbalance ratio group. The model performance can be significantly improved by our novel heuristic method especially in case of high imbalance ratio group. The remaining of this paper is organized as follows. Section II is Theory and Literature review. Section III is the Material and Method explaining the novel heuristic method and how to calculate the heuristic value. Section IV is the research workflow and research framework. Section V presents the experimental results. Section VI is the conclusion of this paper and the recommendation is presented in Section VII.

II. Background Theory and Literature Review

A. Decision Tree

Decision tree is a well-known and one of the most employed technique to generate classifier [22]. Decision tree has 3 important parts: a root node, leaf nodes, and branches to connect nodes. The root node is the origin node of the tree, and both root and other internal nodes consist of condition or criteria to be considered before selecting a branch to traverse. Each branch is a connection line between nodes. Leaf node is a final solution for a specific classification problem.

The tree building process starts with all the training data in the root node. A first split is made using a predictor variable to segment data into 2 or more child nodes, depending on the possible values of the predictor variable. The terminal node is the node that cannot be further split, and the predictions are made from the terminal nodes. To use a decision tree to make

Manuscript received August 20, 2018; revised October 12, 2018.

The authors are with School of Computer Engineering, Suranaree University of Technology, Nakhon Ratchasima 30000, Thailand (e-mail: Anusara.hi@gmail.com, nittaya@sut.ac.th, kerdpras@sut.ac.th).

doi: 10.18178/ijmlc.2018.8.6.746

a prediction, the split decisions are followed until a terminal node is reached.

Decision trees are always mentioned as popular tools for presenting a decision-making process [23], because they are easy for understanding with the clearly graphic. But building efficient decision trees from data is quite complicated. The classical method such as ID3, developed by Quinlan [24]–[26], takes a table of examples as input, where each example consists of a collection of attributes, together with a class. And then, induces a decision tree, where each node is a test on an attribute, each branch is the outcome of that test. The last branching step leads to one of the leaf nodes consisting of the class value to which the example, when following that path, belongs. With the continuous development and improvement, many algorithms such as C4.5 and C5.0 [27] are developed to focus on how to build a decision tree efficiently based on several criteria of consideration [28].

In this research, we use the C5.0 as an algorithm to build model because it has been shown the very satisfying performance compared to other algorithms. Besides the easy-to-understand which is the strongest point of the decision tree, the robustness is also another advantage that makes decision tree popular. It has the ability to be applied with many types of data, fast in prediction, and no need for the assumption on variable distribution [29].

B. Imbalanced Data

Data imbalance is often reported as a problem to reduce classification efficiency in traditional learning algorithms. In classification task, imbalanced data problem occurs when the samples size from the majority class is heavily higher than minority class, and the minority class is usually misclassified by such classification models [30], [31]. Thus, methods to balance the skewed data, such as under-sampling and over-sampling, have been used to tackle the problem. However, under-sampling may drop some potentially useful information, while over-sampling may be the cause of another problem like overfitting [32], [33]. Therefore, it is reasonable to develop the algorithm without conversion from imbalanced data into balanced ones by introducing extra information or removing the original information. The misclassification cost adjustment or cost-sensitive learning is the answer.

The cost-sensitive learning algorithm is developed based on the assumption that the positive minority class is expected to be more important than the majority negative class. Thus, instances in positive class have been weighted with more value than those in negative class. The weighting scheme is based on the misclassification cost adjustment occurred during the iterative model assessment process. The difficulty of this method is finding a proper value for misclassification cost that should be adjusted. The optimal goal is adjusting with the value that results in the highest classification performance on both classifying the minority and majority classes. Unfortunately, a suitable value of misclassification cost comes from many times of trial and run the model repeatedly to see the satisfied result.

C. Confusion Matrix

Confusion matrix [34] is a table that is normally used as a tool for computing performance of a classification model. The key function of this table is to present a comparison between "Predicted Labels" from model and "Actual Labels" from the ground truth. Fig. 1 shows the example of classification outcome of data instances from two groups: "Positive" and "Negative".

		Predicted Label: Positive	Predicted Label: Negative
Actual Label: Positive		TRUE POSITIVE TP	FALSE NEGATIVE FN
Actual Label: Negative		FALSE POSITIVE FP	TRUE NEGATIVE TN

Fig. 1. Example of confusion matrix.

- *True Positive (TP):* The number of instances that a model predicts correctly such that the "Actual Labels" is Positive and "Predicted Labels" is Positive as well.
- *True Negative (TN):* The number of instances that a model predicts correctly such that the "Actual Labels" is Negative and "Predicted Labels" is Negative as well.
- *False Positive (FP):* The number of instances that a model predicts incorrectly such that the "Actual Labels" is Negative but "Predicted Labels" is Positive.
- *False Negative (FN):* The number of instances that a model predicts incorrectly such that the "Actual Labels" is Positive but "Predicted Labels" is Negative.

True Positive Rate (TPR), or Sensitivity, measures the proportion of actual positive data instances that are correctly identified. The calculation of TPR is shown in equations 1 and 2.

$$TPR = TP / (TP + FN) \quad (1)$$

or

$$TPR = TP / (\text{All actual positive instances}) \quad (2)$$

False Positive Rate (FPR) is a metric for measuring the error of classification. It is calculate with the equations 3 and 4.

$$FPR = FP / (FP + TN) \quad (3)$$

or

$$FPR = FP / (\text{All actual negative instances}) \quad (4)$$

D. Performance Evaluation

In classification, there are various measurement methods for evaluating the performance of classification models. Receiver Operating Characteristic curve (ROC) is the visualization to represent the relation of the false positive rate (FPR) against the true positive rate (TPR) by plotting graphs with TPR on the Y-axis and FPR on the X-axis. The performance of a classifier is presented by ROC curve. If it lies in the upper left of the square that means good performance.

AUC or area under the ROC curve [35], [36] is the popular measure for evaluating the performance of a classification model with binary classes. AUC provides a value description for the performance of the ROC curve. AUC is a portion of

International Journal of Machine Learning and Computing, Vol. 8, No. 6, December 2018

the area inside the square of unit (Fig. 2). So, its value must be in the range of 0 and 1, and usually higher than 0.5.

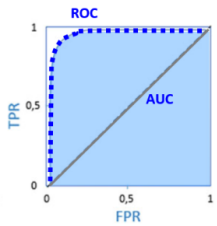


Fig. 2. Example of ROC Curve.

III. Materials and Method

A. Datasets of Research

TABLE I: The 34 Datasets of "Low Imbalance Ratio" Showing Numbers of Majority Class and Minority Class

Group Low Imbalance Ratio						
No.	Dataset	IR	# Attr.	# Ins.	# Major	# Minor
1	glass-0-1-6_vs_5	19.4	9	184	175	9
2	abalone9-18	16.4	8	731	689	42
3	page-blocks-1-3_vs_4	15.9	10	472	444	28
4	ecoli4	15.8	7	336	316	20
5	glass4	15.5	9	214	201	13
6	yeast-1_vs_7	14.3	7	459	429	30
7	shuttle-c0-vs-c4	13.9	9	1,829	1,706	123
8	ecoli-0-1-4-6_vs_5	13.0	6	280	260	20
9	cleveland-0_vs_4	12.6	13	177	164	13
10	ecoli-0-1-4-7_vs_5-6	12.3	6	332	307	25
11	glass2	11.6	9	214	197	17
12	glass-0-1-4-6_vs_2	11.1	9	205	188	17
13	ecoli-0-1_vs_5	11.0	6	240	220	20
14	glass-0-6_vs_5	11.0	9	108	99	9
15	led7digit-0-2-4-5-6-7-8-9_vs_1	11.0	7	443	406	37
16	ecoli-0-1-4-7_vs_2-3-5-6	10.6	7	336	307	29
17	glass-0-1-6_vs_2	10.3	9	192	175	17
18	ecoli-0-6-7_vs_5	10.0	6	220	200	20
19	vowel0	10.0	13	988	898	90
20	yeast-0-5-6-7-9_vs_4	9.4	8	528	477	51
21	ecoli-0-3-4-7_vs_5-6	9.3	7	257	232	25
22	ecoli-0-3-4-6_vs_5	9.3	7	205	185	20
23	glass-0-4_vs_5	9.2	9	92	83	9
24	ecoli-0-2-6-7_vs_3-5	9.2	7	224	202	22
25	ecoli-0-1_vs_2-3-5	9.2	7	244	220	24
26	ecoli-0-4-6_vs_5	9.2	6	203	183	20
27	yeast-0-2-5-6_vs_3-7-8-9	9.1	8	1,004	905	99
28	yeast-0-2-5-7-9_vs_3-6-8	9.1	8	1,004	905	99
29	yeast-0-3-5-9_vs_7-8	9.1	8	506	456	50
30	glass-0-1-5_vs_2	9.1	9	172	155	17
31	ecoli-0-2-3-4_vs_5	9.1	7	202	182	20
32	ecoli-0-6-7_vs_3-5	9.1	7	222	200	22
33	yeast-2_vs_4	9.1	8	514	463	51
34	ecoli-0-3-4_vs_5	9.0	7	200	180	20

The experimentation of this research is to demonstrate that our novel heuristic method can help improving the performance of classification model in various application areas with different imbalance ratios. So, all of 56 datasets are collected from 2 famous real-world dataset repositories, which are "KEEL" and "KDD-CUP". Then, we group them into two groups of imbalance ratio, that is, "low imbalance ratio" with a range of imbalance ratio from 9 to 20, and "high imbalance ratio" which imbalance ratio is over 20 and the maximum of imbalance ratio is 129. The Table I shows 34 datasets of "low imbalance ratio" and Table II show 24 datasets of "high imbalance ratio".

TABLE II: The 26 Datasets of "High Imbalance Ratio" Showing Numbers of Majority Class and Minority Class

Group High Imbalance Ratio						
No.	Dataset	IR	# Attr.	# Ins.	# Major	# Minor
1	abalone19	129.4	8	4,174	4,142	32
2	kddcup-rootkit-imap_vs_back	100.1	41	2,225	2,203	22
3	poker-8_vs_6	85.9	10	1,477	1,460	17
4	poker-8-9_vs_5	82.0	10	2,075	2,050	25
5	kr-vs-k-zero_vs_fifteen	80.2	6	2,193	2,166	27
6	kddcup-land_vs_satan	75.7	41	1,610	1,589	21
7	kddcup-buffer_overflow_vs_back	73.4	41	2,233	2,203	30
8	abalone-20_vs_8-9-10	72.7	8	1,916	1,890	26
9	winequality-red-3_vs_5	68.1	11	691	681	10
10	shuttle-2_vs_5	66.7	9	3,316	3,267	49
11	poker-8-9_vs_6	58.4	10	1,485	1,460	25
12	winequality-white-3-9_vs_5	58.3	11	1,482	1,457	25
13	kr-vs-k-zero_vs_eight	53.1	6	1,460	1,433	27
14	yeast6	41.4	8	1,484	1,449	35
15	ecoli-0-1-3-7_vs_2-6	39.1	7	281	274	7
16	yeast5	32.7	8	1,484	1,440	44
17	yeast-1-2-8-9_vs_7	30.6	8	947	917	30
18	yeast4	28.1	8	1,484	1,433	51
19	glass5	22.8	9	214	205	9
20	yeast-1-4-5-8_vs_7	22.1	8	693	663	30
21	yeast-2_vs_8	21.3	8	482	426	20
22	shuttle-c2-vs-c4	20.5	9	129	123	6

B. A Novel Heuristic Method

The novel heuristic method that we present in this paper is extracted from experience over 5 years of data mining expert engineers in the manufacturing field. The formula of this novel heuristic method is the relation between misclassification cost, imbalance ratio, and the constant e which is the "Euler's number" (~2.71828…). The computation of this heuristic is shown in equation 5.

$$\text{MCC} = \sqrt{\frac{\text{IR}^2}{e}} \tag{5}$$

where

MCC = misclassification cost or cost sensitive,
IR = imbalance ratio, and
e = Euler's number (constant number ~2.718…).

IR or imbalance ratio is defined by the calculation as shown in equation 6.

$$IR = \frac{\text{Number of majority class}}{\text{Number of minority class}} \tag{6}$$

We empirically validate this proposed method and have been found that it can improve classification performance in terms of the true positive rate in root cause analysis of computer's component manufacturing datasets with IR in the range of 4.1 to 1,245.7.

IV. Research Framework and Research Workflow

A. Research Framework

In this paper, we use 56 real-world datasets from serval areas such as medical/scientific experiment, wine quality, and many others. The minimum of imbalance ratio is 9 and the maximum is 129. These 56 datasets are classified into two groups: "low imbalance ratio" and "high imbalance ratio". The model that we use for classification in this paper is the state of art model in IBM SPSS Modeler, C5.0 model

(research framework is shown in Fig. 3). Then, we compare the model result between the traditional method and our novel heuristic method. The assumption of comparison in this paper focuses on 2 points:

1) The novel method should show better performance than the traditional method.
2) The high imbalance ratio group should show better of improvement rate than the low imbalance ratio group.

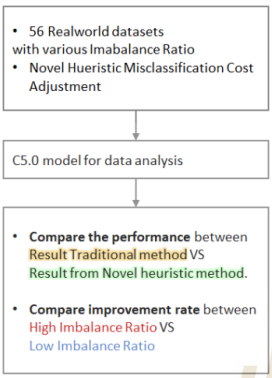


Fig. 3. Research framework.

B. Research Workflow

The research workflow of this research is shown in Fig. 4. Each of the 56 real-world datasets is used as input into the C5.0 model with 70% data instances for training the model and keep aside 30% of the rest for model validation. The same datasets are operated with two methods: "Traditional Method" and "Novel Heuristic Method Misclassification Cost Adjustment". After we run through this process we will obtain two classifiers from the two methods. Then the 30% of data that we set aside in earlier step will be used to test performance of classifiers. The final step is comparing the model performance in terms of AUC.

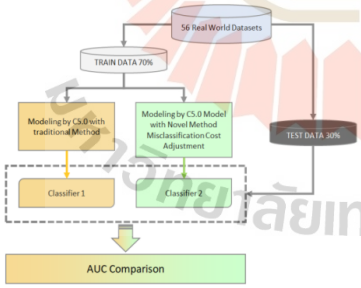


Fig. 4. Research workflow.

V. Experimentation and Results

This section is a demonstration of the experimentation results. The key point is a comparison between traditional method and novel heuristic method (or called proposed method). Table III is the experimentation results of "low imbalance ratio" group. There are 34 datasets in this group. The average imbalance ratio is 11.3 (minimum is 9 and maximum is 19.4), misclassification cost is averaged as 0.81.

In terms of AUC comparison, average AUC before adjusting misclassification cost (traditional methods) is 0.81 and after adjusting misclassification cost with the proposed method, AUC is 0.93. The proposed method shows the better AUC with the improvement rate of 18%. The top-3 of improvement rate are the dataset named "cleveland-0_ vs_4", "glass-0-1-6_vs_5" and "glass-0-1-6_vs_5" with the improvement rate of 97%, 82% and 57%, respectively.

The improvement rate is calculated by equation 7.

$$\text{Improvement Rate} = \frac{\text{Proposed AUC} - \text{Traditional AUC}}{\text{Traditional AUC}} \quad (7)$$

TABLE III: AUC Comparison between "Traditional Method" and "Propose Method" in Group "Low Imbalance Ratio"

				AUC : Area Under ROC Curve				
Group "Low Imbalance Ratio"				Traditional Method		Proposed Method		Improvement Rate
No.	Dataset	IR	MCC	Training	Testing	Training	Testing	
1	glass-0-1-6_vs_5	19.4	11.8	1.00	0.50	0.95	0.91	82%
2	abalone9-18	16.4	10.0	0.82	0.61	0.92	0.68	11%
3	page-blocks-1-3_vs_4	15.9	9.6	1.00	1.00	1.00	1.00	0%
4	ecoli4	15.8	9.6	0.88	0.74	0.97	0.88	18%
5	glass4	15.5	9.4	1.00	0.57	0.98	0.89	57%
6	yeast-1_vs_7	14.3	8.7	0.78	0.79	0.92	0.96	23%
7	shuttle-c0-vs-c4	13.9	8.4	1.00	1.00	1.00	1.00	0%
8	ecoli-0-1-4-6_vs_5	13.0	7.9	0.93	0.90	0.98	0.98	9%
9	cleveland-0_vs_4	12.6	7.7	0.88	0.46	0.97	0.90	97%
10	ecoli-0-1-4-7_vs_5-6	12.3	7.4	0.90	0.61	0.99	0.90	47%
11	glass2	11.6	7.0	0.97	0.92	0.96	0.92	0%
12	glass-0-1-4-6_vs_2	11.1	6.7	0.90	0.85	0.96	0.93	9%
13	ecoli-0-1_vs_5	11.0	6.7	0.99	0.75	0.97	0.93	24%
14	glass-0-6_vs_5	11.0	6.7	0.99	1.00	0.99	1.00	0%
15	led7digit-0-2-4-5-6-7-8-9_vs_1	11.0	6.7	0.96	0.94	0.97	0.94	0%
16	ecoli-0-1-4-7_vs_2-3-5-6	10.6	6.4	0.88	0.91	0.98	0.95	5%
17	glass-0-1-6_vs_2	10.3	6.2	0.50	0.50	0.94	0.79	57%
18	ecoli-0-6-7_vs_5	10.0	6.1	0.92	0.79	0.99	0.93	18%
19	vowel0	10.0	6.1	0.99	0.94	1.00	1.00	7%
20	yeast-0-5-6-7-9_vs_4	9.4	5.7	0.85	0.87	0.96	0.91	5%
21	ecoli-0-3-4-7_vs_5-6	9.3	5.6	0.84	0.90	0.90	1.00	11%
22	ecoli-0-3-4-6_vs_5	9.3	5.6	0.89	0.83	0.98	0.91	9%
23	glass-0-4_vs_5	9.2	5.6	0.99	1.00	0.99	1.00	0%
24	ecoli-0-2-6-7_vs_3-5	9.2	5.6	0.90	0.83	0.93	0.92	10%
25	ecoli-0-1_vs_2-3-5	9.2	5.6	0.99	0.77	0.99	0.90	16%
26	ecoli-0-4-6_vs_5	9.2	5.6	0.87	0.74	0.99	0.95	28%
27	yeast-0-2-5-6_vs_3-7-8-9	9.1	5.5	0.78	0.80	0.86	0.92	14%
28	yeast-0-2-5-7-9_vs_3-6-8	9.1	5.5	0.90	0.86	0.99	0.91	5%
29	yeast-0-3-5-9_vs_7-8	9.1	5.5	0.79	0.88	0.93	0.89	1%
30	glass-0-1-5_vs_2	9.1	5.5	0.84	0.71	0.95	0.92	30%
31	ecoli-0-2-3-4_vs_5	9.1	5.5	0.93	0.92	0.98	0.97	6%
32	ecoli-0-6-7_vs_3-5	9.1	5.5	0.85	1.00	0.98	1.00	0%
33	yeast-2_vs_4	9.1	5.5	1.00	0.96	0.98	0.98	3%
34	ecoli-0-3-4_vs_5	9.0	5.5	0.84	0.83	0.99	0.90	8%
	AVG	**11.3**	**6.8**	0.90	**0.81**	0.97	**0.93**	**18%**

Table IV is the experimental results of "high imbalance ratio" showing comparative AUC performance between "Traditional Method" and "Proposed Method". In this groups, there are 22 datasets. The average value of imbalance ratio is 57.39 (minimum is 20.5 and maximum is 129). Average of misclassification cost adjustment is 34.8. AUC of traditional method is 0.74 compared to 0.90 of the proposed method. There are many datasets showing better performance in terms of AUC with high improvement rate.

The top-5 datasets are "yeast-1-4-5-8_vs_7", winequality-red-3_vs_5", "poker-8-9_vs_6", "poker-8_vs _6" and "winequality-white-3-9_vs_5". The improvement

rates are 77%, 71%, 67%, 65% and 63%, respectively. The average improvement rate is as high as 29%. It is a significant gap when compared to "low imbalance ratio" (which has an improvement rate of 18%).

TABLE IV: AUC COMPARISON BETWEEN "TRADITIONAL METHOD" AND "PROPOSE METHOD" IN GROUP "HIGH IMBALANCE RATIO"

				AUC : Area Under ROC Curve				
Group "High Imbalance Ratio"				Traditional Method		Proposed Method		Improvement Rate
No.	Dataset	IR	MCC	Training	Testing	Training	Testing	
1	abalone19	129.4	78.5	0.50	0.50	0.91	0.64	28%
2	kddcup-rootkit-imap_vs_back	100.1	60.7	1	1	1	1	0%
3	poker-8_vs_6	85.9	52.1	0.5	0.5	0.806	0.826	65%
4	poker-8-9_vs_5	82.0	49.7	0.5	0.5	0.849	0.718	44%
5	kr-vs-k-zero_vs_fifteen	80.2	48.7	0.932	0.801	0.965	0.964	20%
6	kddcup-land_vs_satan	75.7	45.9	1	1	1	1	0%
7	kddcup-buffer_overflow_vs_back	73.4	44.5	1	1	1	1	0%
8	abalone-20_vs_8-9-10	72.7	44.1	0.839	0.739	0.941	0.869	18%
9	winequality-red-3_vs_5	68.1	41.3	0.5	0.5	0.925	0.857	71%
10	shuttle-2_vs_5	66.7	40.4	1	1	1	1	0%
11	poker-8-9_vs_6	58.4	35.4	0.5	0.5	0.857	0.837	67%
12	winequality-white-3-9_vs_5	58.3	35.3	0.648	0.578	0.916	0.945	63%
13	kr-vs-k-zero_vs_eight	53.1	32.2	0.988	1	0.982	0.984	-2%
14	yeast6	41.4	25.1	0.92	0.75	0.90	0.96	28%
15	ecoli-0-1-3-7_vs_2-6	39.1	23.7	0.80	0.99	0.98	0.99	0%
16	yeast5	32.7	19.9	0.99	0.91	0.99	0.98	8%
17	yeast-1-2-8-9_vs_7	30.6	18.5	0.66	0.49	0.93	0.70	43%
18	yeast4	28.1	17.0	0.86	0.80	0.96	0.87	8%
19	glass5	22.8	13.8	1.00	0.63	0.94	0.98	57%
20	yeast-1-4-5-8_vs_7	22.1	13.4	0.50	0.50	0.89	0.89	77%
21	yeast-2_vs_8	21.3	12.9	0.50	0.50	0.79	0.75	50%
22	shuttle-c2-vs-c4	20.5	12.4	1.00	1.00	1.00	1.00	0%
	AVG	**57.39**	**34.81**	**0.78**	**0.74**	**0.93**	**0.90**	**29%**

VI. Conclusion

In this paper, we presented the novel heuristic method to compute proper cost-sensitive value for classifying imbalanced data that have high imbalance ratio between the tremendous majority class as compared to the tiny minority class. The experimentation have been performed on the 56 real-world datasets to assess the improvement rate of AUC when compared to the traditional classification method. These datasets are from various domains and various imbalance ratios. The key proposals of this paper are based on the two assumptions:

- Novel method can improve the model performance when compared to traditional classification method.
- High imbalance ratio should show the better improvement rate than low imbalance ratio.

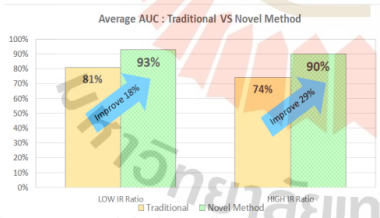


Fig. 5. Summary graph showing overall AUC comparisons improvement rate between "traditional method" and "propose method".

It turns out that the experimental results confirm our assumptions. From overall data, we can see the improvement rate at the satisfying level. For the 34 datasets of low imbalance group, with the imbalance ratio ranging from 9 to 20, the improvement rate is about 18%. For the 22 datasets of high imbalance ratio (with imbalance ratio over 20), the improvement rate is 29% on average. A graph of overall AUC comparisons is shown in Fig. 5. From this result, we can conclude that our novel heuristic method is suitable for classifying data with high imbalance ratio.

VII. Recommendation

On standard datasets obtained from the worldwide repositories, we observe that imbalance ratios in these data are not so high (11.3 to 57.39 on average). This is unlike real production data of manufacturing fields in which the imbalance ratio can be as high as 1: 1,000 or over. Based on the experimental results that reveal significant classification improvement when the imbalance ratio is very high, we thus expect that the proposed novel heuristic method can show clearly the improvement over traditional classification when the imbalance ratio of manufacturing data is in extreme level.

In our further research, we plan to use this method in misclassification cost adjustment with data in other fields that have extremely high level of imbalance ratio. Moreover, the multiclass target classification is also the challenging area that we would like to tackle with this method.

Acknowledgment

This research work has been supported by grants from the National Research Council of Thailand (NRCT). The first author has been support by the scholarship from Suranaree University of Technology. All three authors are researchers of the Data and Knowledge Engineering Research Unit that has been fully supported by research grant from Suranaree University of Technology.

References

[1] W.-L. Chao, J.-Z. Liu, and J.-J. Ding, "Facial age estimation based on label-sensitive learning and age-oriented regression," *Pattern Recognit.*, vol. 46, no. 3, pp. 628–641, 2013.
[2] M. Kubat, R. C. Holte, and S. Matwin, "Machine learning for the detection of oil spills in satellite radar images," *Mach. Learn.*, vol. 30, no. 2–3, pp. 195–215, 1998.
[3] W. Khreich, E. Granger, A. Miri, and R. Sabourin, "Adaptive ROC-based ensembles of HMMs applied to anomaly detection," *Pattern Recognit.*, vol. 45, no. 1, pp. 208–230, 2012.
[4] T. Fawcett and F. Provost, "Adaptive fraud detection," *Data Min. Knowl. Discov.*, vol. 1, no. 3, pp. 291–316, 1997.
[5] L. Pelayo and S. Dick, "Applying novel resampling strategies to software defect prediction," in *Fuzzy Information Processing Society, 2007. NAFIPS'07. Annual Meeting of the North American*, 2007, pp. 69–72.
[6] D. Zhang, M. M. Islam, and G. Lu, "A review on automatic image annotation techniques," *Pattern Recognit.*, vol. 45, no. 1, pp. 346–362, 2012.
[7] G. M. Weiss, "Mining with rarity: A unifying framework," *ACM Sigkdd Explor. Newsl.*, vol. 6, no. 1, pp. 7–19, 2004.
[8] G. E. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *ACM SIGKDD Explor. Newsl.*, vol. 6, no. 1, pp. 20–29, 2004.
[9] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowl. Data Eng.*, no. 9, pp. 1263–1284, 2008.
[10] I. Mani and I. Zhang, "kNN approach to unbalanced data distributions: A case study involving information extraction," in *Proc. Workshop on Learning from Imbalanced Datasets*, 2003, vol. 126.
[11] W. Liu and S. Chawla, "Class confidence weighted knn algorithms for imbalanced data sets," in *Proc. Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2011, pp. 345–356.
[12] F. Provost, "Machine learning from imbalanced data sets 101," in *Proc. the AAAI workshop on imbalanced data sets*, 2000, pp. 1–3.
[13] Y. Sun, M. S. Kamel, A. K. C. Wong, and Y. Wang, "Cost-sensitive boosting for classification of imbalanced data," *Pattern Recognit.*, vol. 40, no. 12, pp. 3358–3378, 2007.

[14] X.Y. Liu, J. Wu, and Z.H. Zhou, "Exploratory undersampling for class-imbalance learning," *IEEE Trans. Syst. Man, Cybern. Part B*, vol. 39, no. 2, pp. 539–550, 2009.
[15] R. Barandelaa, J. S. Sanchezb, and V. Garcia, "Strategies for learning in class imbalance problems," 2003.
[16] M. A. Tahir, J. Kittler, and F. Yan, "Inverse random under sampling for class imbalance problem and its application to multi-label classification," *Pattern Recognit.*, vol. 45, no. 10, pp. 3738–3750, 2012.
[17] N. V Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, 2002.
[18] S. Garcia and F. Herrera, "Evolutionary undersampling for classification with imbalanced datasets: Proposals and taxonomy," *Evol. Comput.*, vol. 17, no. 3, pp. 275–306, 2009.
[19] G. M. Weiss, K. McCarthy, and B. Zabar, "Cost-sensitive learning vs. sampling: Which is best for handling unbalanced classes with unequal error cost," *DMIN*, vol. 7, pp. 35–41, 2007.
[20] Z.-H. Zhou and X.-Y. Liu, "Training cost-sensitive neural networks with methods addressing the class imbalance problem," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 1, pp. 63–77, 2006.
[21] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, "A comparative study of data sampling and cost sensitive learning," in *Proc. IEEE International Conference on Data Mining Workshops, 2008*, 2008, pp. 46–52.
[22] P. Su, W. Mao, and D. Zeng, "An empirical study of cost-sensitive learning in cultural modeling," *Inf. Syst. E-bus. Manag.*, vol. 11, no. 3, pp. 437–455, 2013.
[23] S. Lomax and S. Vadera, "A survey of cost-sensitive decision tree induction algorithms," *ACM Comput. Surv.*, vol. 45, no. 2, p. 16, 2013.
[24] J. R. Quinlan, "Discovering rules by induction from large collections of examples," *Expert Syst. Micro Electron. Age*, 1979.
[25] J. R. Quinlan, "Learning efficient classification procedures and their application to chess end games," *Machine Learning*, Elsevier, vol. I, pp. 463–482, 1983.
[26] L. A. Breslow and D. W. Aha, "Simplifying decision trees: A survey," *Knowl. Eng. Rev.*, vol. 12, no. 1, pp. 1–40, 1997.
[27] J. R. Quinlan, *C4. 5: Programs for Machine Learning*, Elsevier, 2014.
[28] S. Bertolini, A. Maoli, G. Rauch, and M. Giacomini, "Entropy-driven decision tree building for decision support in gastroenterology," *Stud Heal. Technol Inf.*, vol. 186, pp. 93–97, 2013.
[29] T. Wendler and S. Gröttrup, *Data mining with SPSS Modeler: Theory, Exercises and Solutions*, Springer, 2016.
[30] A. Estabrooks, T. Jo, and N. Japkowicz, "A multiple resampling method for learning from imbalanced data sets," *Comput. Intell.*, vol. 20, no. 1, pp. 18–36, 2004.
[31] W. Lu, Z. Li, and J. Chu, "Adaptive ensemble undersampling-boost: A novel learning framework for imbalanced data," *J. Syst. Softw.*, vol. 132, pp. 272–282, 2017.
[32] W. C. Lin, C. F. Tsai, Y.-H. Hu, and J.-S. Jhang, "Clustering-based undersampling in class-imbalanced data," *Inf. Sci. (Ny).*, vol. 409, pp. 17–26, 2017.
[33] L. Peng, H. Zhang, B. Yang, and Y. Chen, "A new approach for imbalanced data classification based on data gravitation," *Inf. Sci. (Ny).*, vol. 288, pp. 347–373, 2014.
[34] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Inf. Process. Manag.*, vol. 45, no. 4, pp. 427–437, 2009.
[35] J. M. Lobo, A. Jiménez-Valverde, and R. Real, "AUC: A misleading measure of the performance of predictive distribution models," *Glob. Ecol. Biogeogr.*, vol. 17, no. 2, pp. 145–151, 2008.
[36] L. Sun, J. Wang, and J. Wei, "AVC: Selecting discriminative features on basis of AUC by maximizing variable complementarity," *BMC Bioinformatics*, vol. 18, no. 3, p. 50, 2017.

Anusara Hirunyawanakul is a Ph.D. student, School of Computer Engineering, Suranaree University of Technology (SUT), Thailand. She received her B.E. and M.E. in computer engineering from Suranaree University of Technology, Thailand, in 2006 and 2014. Her research of interest includes Data Mining Applications, Machine Learning, and Artificial Intelligence in Manufacturing.

Nittaya Kerdprasop is an associate professor and the head of Data Engineering Research Unit, School of Computer Engineering, Suranaree University of Technology (SUT), Thailand. She received her B.S. in radiation techniques from Mahidol University, Thailand, in 1985, M.S. in computer science from the Prince of Songkla University, Thailand, in 1991 and Ph.D. in computer science from Nova Southeastern University, U.S.A., in 1999. Her research of interest includes Data Mining, Artificial Intelligence, Logic and Constraint Programming.

Kittisak Kerdprasop is an associate professor at the School of Computer Engineering, Chair of the School, and the head of Knowledge Engineering Research Unit, SUT. He received his bachelor degree in Mathematics from Srinakarinwirot University, Thailand, in 1986, MS in computer science from the Prince of Songkla University, Thailand, in 1991 and Ph.D. in computer science from Nova Southeastern University, U.S.A., in 1999. His current research includes Machine Learning and Artificial Intelligence.

Misclassification Cost Tuning for Root Cause Analysis on Hard Disk Drive Manufacturing

Anusara Hirunyawanakula[1], Nittaya Kerdprasop[2] and Kittisak Kerdprasop[3]

[1,2,3] *School of Computer Engineering, Suranaree University of Technology, Thailand*
[2] *Data Engineering Research Unit, Suranaree University of Technology.*
[3] *Knowledge Engineering Research Unit, Suranaree University of Technology*
Anusara.hi@gmail.com

Abstract— Currently, one of the most popular application area of Machine Learning and Data Mining for inducing new knowledge for problem solving is the manufacturing field. Data in production line of manufacturing process are very big and complicated. Imbalance data can be generally found in many study cases like a failure root-cause analysis due to the fact that failure data are very small compared to the majority data that pass the quality test. Discovering useful knowledge from imbalance data is the challenging problem because the model is basically built for the best overall accuracy. But the overall accuracy is not so important when we focus on how much percentage that we can predict correctly the positive class, which is the minority data containing failure products. The simple model may predict the target class biasedly to majority class because it is the most straightforward tactic to achieve high predictive accuracy, while true positive rate of recognizing the failure case may be zero. The technique that data mining experts favor to use for tackling this problem is adjusting misclassification cost to make the model shows the higher true positive rate. However, this technique has problem guessing reasonable value of misclassification cost at first trial. If the first value is far from the optimal one, the experts need to do a lot of trials to adjust the value of misclassification cost for the good true positive rate result. This research focuses on how to improve efficiency on misclassification cost adjustment process by proposing the novel heuristic method to guide the optimal value for the first trial of setting misclassification cost. The proposed method has been tested on five study cases from real data of the hard-disk drive production manufacturing. We found that the new method can help the model achieve the best true positive rate in a minimal number of trials.

Index Terms— *Misclassification Cost; Imbalance Data; Intelligent Manufacturing; Root Cause Analysis; Decision Tree Learning.*

I. Introduction

Since the industrial revolution, efficiency in manufacturing has been constantly improve. It started at the dawn of mechanization which used the water and stream power and transformed into mass production with electricity-powered operations. The subsequent generation is powered by digitalization and power of electronics. Digital data in production line were used for feedback control in real time. Current is generation of industrial 4.0 in which data in production line are used for advanced analytics [1] – [7]. The machine learning and data mining roll into this area and were used to find solution in many problems. The key goal of using data analytics is to improve productivity by reducing costs without compromising quality. We can group key jobs in need for analytics into 4 categories: "reduce test time and calibration time", "warranty cost reduction" by improving quality test performance, "perform predictive maintenance", and "yield improvement". Yield improvement is a very popular category that we often see data analytics applied on. The analytical tasks include benchmark analysis across lines and plants, improving first-pass yield, and pinpoint the root cause of failure.

On failure root cause analysis, the classification technique is one of the most often used. Generally, measurement tools of classification are accuracy, precision, and true positive rate. Each measurement tool is suitable for particularly data. Such as true positive rate is suitable for dataset or problem that we need to focus on the interesting class (positive class). In manufacturing data analysis, the proportion of "passer" (the unit that can pass through the process as finish good unit) and "failure" (the unit that cannot pass the process) is very imbalance. Normally, we always see that passer ratio is much higher. This makes sense because one of product cost is came from scrap cost. If we can reduce the failure, that means we can reduce the product cost as well. For data analysis experts, inducing valuable knowledge from such high imbalance data is challenging. When they are facing imbalance data, they often use technique called misclassification cost adjustment. This technique is based on assumption that the positive class should be more important than negative class then we weighted positive data with more value than those in negative class. However, the difficulty of this process is finding the optimal value of misclassification cost. Optimal cost should be the value

that can cause the highest true positive rate. Traditionally, the optimal misclassification cost results from many times of trial that the engineers (or data analysis expert) try to input value and run the model to see the result. This process takes very long times because we need to do the process repeatedly with different number of misclassification cost until we satisfy with the result. In this paper, we focus on the relation between imbalance ratio and suitability of misclassification cost at first trial. And then, we introduce the proposed heuristic method with the simple equation. We demonstrate efficiency through the test with 5 study cases that we obtain from real manufacturing production line. The result is satisfied, and the proposed heuristic can be applied in real practice.

II. Background Theory

A. Decision tree

Decision tree is probably the most widely used classifier [8]. There are 3 key items on the decision tree: root node, leaf nodes, and branches. The root node is the starting node of the tree, and both root and leaf nodes contain questions or criteria to be answered. Branch is a connection line between nodes, showing the flow from question to answer. The tree building starts with all training data in the first node. The first split is made from using a predictor variable for segmenting the data from one to many child nodes. The terminal node is the node that we cannot further split. And the predictions are made from the terminal nodes. Using a decision tree for prediction, the split decisions are followed until a terminal node is reached. Decision trees is claimed to be one of the most popular tools for presenting a decision-making process [9] because tree is easy for understanding. The most well-known algorithm for decision tree learning is ID3, which was developed by Quinlan [10] – [12]. ID3 takes a table of examples as input, where each example consists of a collection of attributes, together with a class. And then, induces a decision tree, where each node is a test on an attribute, each branch is the resulting value of that test. The last step at the end of leaf nodes indicates the class to which the example, when following that path, belongs. With the continuous improvement many algorithms are developed. C4.5 and C5.0 [13], OC1 [14], and CART [15] are developed to focus on how to make a decision tree maximizes its accuracy.

Overall, the strong points of decision trees are robust to outlier, easy to understand, able to use with many types of data, fast in prediction and no need for assumption on variable distribution [16].

B. Classification over imbalance data sets

In classification jobs, imbalance data sets refer to those that the samples size from the majority class is heavily higher than minority class [17]. Many of existing learning algorithms show frequent incorrect classifications of the samples from the minority class [18]. This is the motivation for the research on the classification problem over imbalance data sets. There are three categories of classification approaches on imbalance data sets: data processing level methods, classifier level methods, and cost-sensitive methods [19]. The data processing is actually naively process level. This method adjusts the ratio of two classes to form balanced data sets by re-sampling strategies. The applied re-sampling technique can be over-sampling [20], under-sampling, or in combination of both [21] – [22]. For the classifier level methods, many classifiers have been proposed based on intelligent algorithms. However, the main drawback of those methods is that the accuracy of the classifiers is easily affected by the performance of the intelligent algorithms.

The cost-sensitivity methods can be divided into the following three groups depending on what cost is considered: the cost from non-uniform misclassification, the cost of tests, and a hybrid cost from a combination of different types of costs. Among cost-sensitive classifier methods [23] – [24], the cost-sensitive decision tree method is one of the most popular methods. Many cost-sensitive methods have been further developed based on cost-sensitive decision trees.

C. Root cause analysis performance evaluation

Confusion Matrix is the well-known table for presenting the result of classification. The main idea is to compare between “Predicted Labels” from model and “Actual Labels” which are known. Confusion matrix in figure 1 is example case of classifying data into 2 groups “TRUE” and “FALSE”. So, the results from this confusion matrix are composed of 4 values:

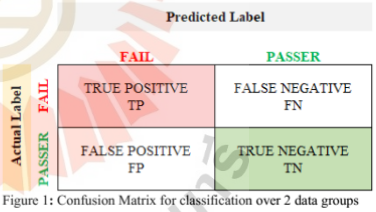


Figure 1: Confusion Matrix for classification over 2 data groups

True Positive (TP): "Actual Labels" is FAIL and "Predicted Labels" is FAIL as well. This case the model predicts correctly.

True Negative (TN): "Actual Labels" is PASSER and "Predicted Labels" is PASSER as well. This case the model predicts correctly.

False Positive (FP): "Actual Labels" is PASSER but "Predicted Labels" is FAIL. This case the model predicts incorrectly.

False Negative (FN): "Actual Labels" is FAIL but "Predicted Labels" is PASSER. This case the model predicts incorrectly.

The accuracy rate is a measurement to assess the overall effectiveness of the model. Calculation for the accuracy is shown in (1).

$$Accuracy = \frac{(TP+TN)}{Total\ data} \quad (1)$$

Even though the accuracy is able to represent the overall performance, this measurement is not suitable when data are imbalance. In case we are interested on positive class that this is the minority class with imbalance scale, the appropriate matrix is True Positive Rate (also known as Recall or Sensitivity) [25]. This measurement focuses on "Actual Labels = FAIL". It presents the ratio of true positive and total of actual positive data, as shown in (2).

$$True\ Positive\ Rate = \frac{TP}{(TP+FN)} \quad (2)$$

III. Materials and Methods

This research uses realistic dataset from production line of hard disk drive manufacturing in which we are provided 5 study cases from the company. The 5 datasets are of various sizes. The number of rows are 1430 – 154,594 rows and number of columns are 10 – 20 columns. We use the data with high range of rows because we would like to prove the novel heuristic method that it is still effective in all ranges of dataset. All these datasets share the same characteristic of imbalancing. The imbalance ratio is 41.1 of the minimum one and 1,245.7 of the maximum one. We hold the 70% of the original data for building the model and use the remaining 30% for testing the effectiveness of model.

The novel heuristic method that we introduce in this research is extracted from the experience of researchers while had been working in the manufacturing companies. The equation for the novel method is inspired from the relation between misclassification cost, imbalance ratio, and e "Euler's number" (~2.71828…). And the proposed equation in (3).

$$CS = \sqrt{\frac{IR^2}{e}} \quad (3)$$

Where CS = Cost sensitive or misclassification Cost for the first trial,
IR = Imbalance Ratio, and
e = Euler's number (constant number ~2.718…).

And the IR Imbalance Ratio is defined by (4).

$$IR = \frac{\text{Number of majority class}}{\text{Number of minority class}} \quad (4)$$

IV. Research Framework and Workflow

The framework of this research (Figure 2) is based on the main objective that we would like to help data analysis experts in the process of misclassification cost adjustment. So, we design the research following this assumption and the later evaluation steps are the test whether the novel heuristic method can achieve the best true positive rate faster than traditional manual method or not.

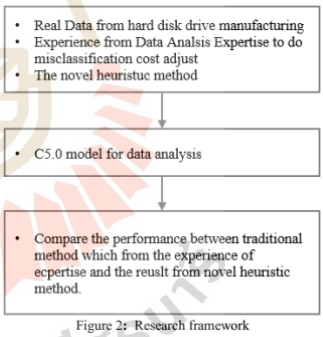


Figure 2: Research framework

The research workflow of this research is shown in Figure 3. We start with data from real production data in manufacturing and input them into C5.0 model. The result from this step is going to be the base case for performance comparison. The same datasets are then adjusted the misclassification cost with 2 methods: traditional one which is normally done by experts, and the novel heuristic method that we initiate and present in this paper. After we have got the results from both methods, we then do the performance comparison in terms of true positive rate in each trial.

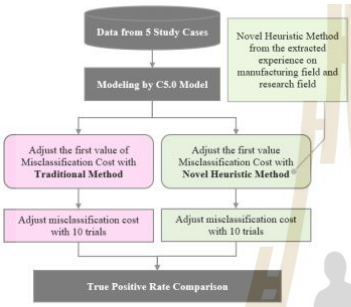


Figure 3: Research workflow

V. RESULTS

The result from the proposed heuristic method has been compared against the traditional method used by engineers and data experts. Table 1 shows true positive rate from novel method and traditional method in first trial up to the tenth trial. In study case 1, 3, 4 and 5 the propose method can achieve the best true positive rate at the first trial while traditional method can achieve the best true as well but in higher number of trials for study case 1, 3 and 4. But in study case 5 with traditional method, it cannot reach the best true positive rate. The study case 2 is quite different from other study cases; the novel method cannot reach the best true positive rate at first, but it can achieve the best rate in the 8th trial.

Table 1
True Positive Rate comparison between Novel Method and Traditional Method for 5 study case

	Study Case#01		Study Case#02		Study Case#03		Study Case#04		Study Case#05	
	Novel Method	Traditional Method	Novel Method	Traditional Method	Novel Method	Traditional Method	Novel Method	Traditional Method	Novel Method	Traditional Method
1st trial	98.4%	80.0%	88.0%	0.0%	90.0%	0.0%	100.0%	100.0%	92.0%	0.0%
2nd trial	98.4%	80.0%	88.0%	0.0%	90.0%	0.0%	100.0%	100.0%	92.0%	80.0%
3rd trial	98.4%	80.0%	88.0%	88.0%	90.0%	90.0%	100.0%	100.0%	92.0%	80.0%
4th trial	98.4%	85.0%	88.0%	88.0%	90.0%	90.0%	100.0%	100.0%	92.0%	80.0%
5th trial	98.4%	85.0%	88.0%	88.0%	90.0%	90.0%	100.0%	100.0%	92.0%	83.0%
6th trial	98.4%	85.0%	88.0%	88.0%	90.0%	90.0%	100.0%	100.0%	92.0%	88.0%
7th trial	98.4%	98.4%	94.0%	94.0%	90.0%	90.0%	100.0%	100.0%	92.0%	88.0%
8th trial	98.4%	98.4%	100.0%	100.0%	90.0%	90.0%	100.0%	100.0%	92.0%	90.0%
9th trial	98.4%	98.4%	100.0%	100.0%	90.0%	90.0%	100.0%	100.0%	92.0%	90.0%
10th trial	98.4%	98.4%	100.0%	100.0%	90.0%	90.0%	100.0%	100.0%	92.0%	90.0%

The comparisons of our proposed heuristic method and the traditional method in each of the five study cases are also graphically shown in Figure 4 to 8.

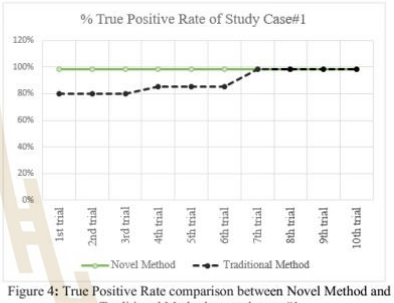


Figure 4: True Positive Rate comparison between Novel Method and Traditional Method on study case#1

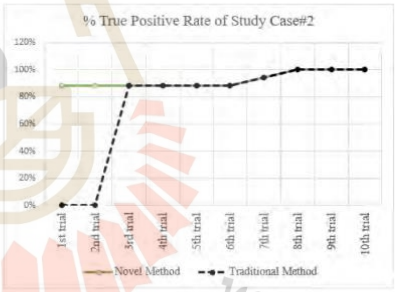


Figure 5: True Positive Rate comparison between Novel Method and Traditional Method on study case#2

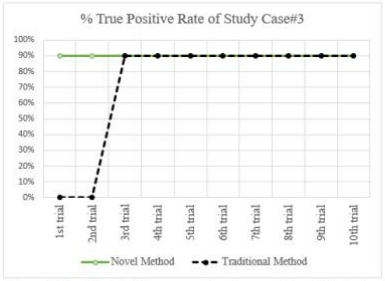


Figure 6: True Positive Rate comparison between Novel Method and Traditional Method on study case#3

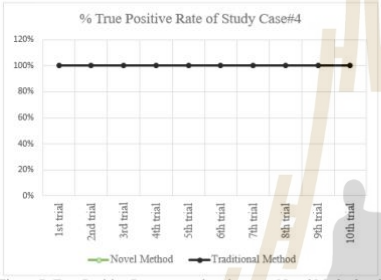


Figure 7: True Positive Rate comparison between Novel Method and Traditional Method on study case#4

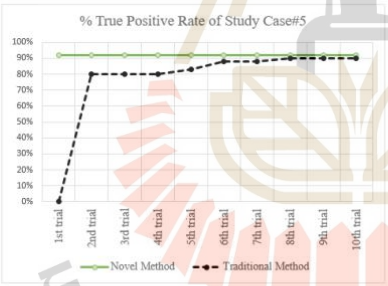


Figure 8: True Positive Rate comparison between Novel Method and Traditional Method on study case#5

VI. Conclusion

Based on data from 5 study cases which we are provided from real production line of the manufacturing process, we can see that the proposed novel heuristic method can achieve the best true positive rate within minimal number of trials when compared to traditional method used by most industrial and data engineers. This means that we can save time on the misclassification cost adjustment step. So, we can conclude that the proposed method can help the data experts in the real practice and the result is satisfying.

Recommendation

The 5 study cases from manufacturing of hard disk drive can experimentally prove that this novel method can help data analysis. But in manufacturing with various production process, there are many interesting problem areas that are potential for our heuristic approach adoption. Our expectation is to study more cases and apply the datamining knowledge to solve the problem in other fields. Another thing that we would like to do in further research is to validate our novel method to other algorithms such as CHAID, CART and QUEST.

Acknowledgment

This research work has been supported by grants from the National Research Council of Thailand (NRCT). Data Engineering and Knowledge Engineering Research Units are fully supported by Suranaree University of Technology.

References

[1] H. Wang, C. Liu, S. Wang, and Y. Li, "Application to car quality evaluation using decision tree technology with imbalance correction coefficient," in *Proceedings of 20th International Conference on Industrial Engineering and Engineering Management, 2013*, pp. 571–581.

[2] C. Seiffert, T. M. Khoshgoftaar, and J. Van Hulse, " Improving software-quality predictions with data sampling and boosting," *IEEE Trans. Syst. Man, Cybern. A Syst. Humans*, vol. 39, no. 6, pp. 1283–1294, 2009.

[3] S. Wang and X. Yao, "Using class imbalance learning for software defect prediction," *IEEE Trans. Reliab.*, vol. 62, no. 2, pp. 434–443, 2013.

[4] M. El-Banna, " A novel approach for classifying imbalance welding data: Mahalanobis genetic algorithm (MGA)," *Int. J. Adv. Manuf. Technol.*, vol. 77, no. 1–4, pp. 407–425, 2015.

[5] V. A. Skormin, V. I. Gorodetski, and L. J. Popyack, " Data mining technology for failure prognostic of avionics," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 38, no. 2, pp. 388–403, 2002.

[6] A. K. Choudhary, J. A. Harding, and M. K. Tiwari, "Data mining in manufacturing: a review based on the kind of knowledge," *J. Intell. Manuf.*, vol. 20, no. 5, p. 501, 2009.

[7] L. Monostori, A. Márkus, H. Van Brussel, and E. Westkämpfer, " Machine learning approaches to manufacturing," *CIRP Ann. Technol.*, vol. 45, no. 2, 1996.

[8] P. Su, W. Mao, and D. Zeng, " An empirical study of cost-sensitive learning in cultural modeling," *Inf. Syst. E-bus. Manag.*, vol. 11, no. 3, pp. 437–455, 2013.

[9] S. Lomax and S. Vadera, " A survey of cost-sensitive decision tree induction algorithms," *ACM Comput. Surv.*, vol. 45, no. 2, p. 16, 2013.

[10] J. R. Quinlan, "Discovering rules by induction from large collections of examples," *Expert Syst. micro Electron. age*, 1979.

[11] J. R. Quinlan, *Learning efficient classification procedures and their application to chess end games.*, in Machine Learning, Volume I, Elsevier, 1983, pp. 463–482.

[12] J. R. Quinlan, "Simplifying decision trees," *Int. J. Man. Mach. Stud.*, vol. 27, no. 3, pp. 221–234, 1987.

[13] J. R. Quinlan, *C4. 5: Programs for machine learning*. Elsevier, 2014.

[14] S. K. Murthy, S. Kasif, and S. Salzberg, " A system for induction of oblique decision trees," *J. Artif. Intell. Res.*, vol. 2, pp. 1–32, 1994.

[15] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and regression trees*. CRC press, 1984.

[16] T. Wendler and S. Gröttrup, *Data mining with SPSS modeler: Theory, exercises and solutions*. Springer, 2016.

[17] F. Li, X. Zhang, X. Zhang, C. Du, Y. Xu, and Y.-C. Tian, " Cost-sensitive and hybrid-attribute measure multi-decision tree over imbalanced data sets," *Inf. Sci. (Ny).*, vol. 422, pp. 242–256, 2018.

[18] S. Alshomrani, A. Bawakid, S.-O. Shim, A. Fernández, and F. Herrera, " A proposal for evolutionary fuzzy systems using feature weighting: dealing with overlapping in imbalanced datasets," *Knowledge-Based Syst.*, vol. 73, pp. 1–17, 2015.

[19] M. Antonelli, P. Ducange, and F. Marcelloni, " An experimental study on evolutionary fuzzy classifiers designed for managing imbalanced datasets," *Neurocomputing*, vol. 146, pp. 125–136, 2014.

[20] I. Nekooeimehr and S. K. Lai-Yuen, " Adaptive semi-unsupervised weighted oversampling (A-SUWO) for imbalanced datasets," *Expert Syst. Appl.*, vol. 46, pp. 405–416, 2016.

[21] S. Cateni, V. Colla, and M. Vannucci, " A method for resampling imbalanced datasets in binary classification tasks for real-world problems," *Neurocomputing*, vol. 135, pp. 32–41, 2014.

[22] B. W. Yap, K. A. Rani, H. A. A. Rahman, S. Fong, Z. Khairudin, and N. N. Abdullah, " An application of oversampling, undersampling, bagging and boosting in handling imbalanced datasets," in *Proceedings of the first international conference on advanced data and information engineering (DaEng-2013)*, 2014, pp. 13–22.

[23] A. C. Bahnsen, D. Aouada, and B. Ottersten, "Example-dependent cost-sensitive decision trees," *Expert Syst. Appl.*, vol. 42, no. 19, pp. 6609–6619, 2015.

[24] S. Bernard, C. Chatelain, S. Adam, and R. Sabourin, " The multiclass roc front method for cost-sensitive classification," *Pattern Recognit.*, vol. 52, pp. 46–60, 2016.

[25] M. Buckland and F. Gey, " The relationship between recall and precision," *J. Am. Soc. Inf. Sci.*, vol. 45, no. 1, p. 12, 1994.

International Journal of Machine Learning and Computing, Vol. 10, No. 2, February 2020

Efficient Machine Learning Methods for Hard Disk Drive Yield Prediction Improvement

Anusara Hirunyawanakul, Nittaya Kerdprasop, and Kittisak Kerdprasop

***Abstract*—Deployment of machine learning techniques are prevailing in world-wide problem solving. Hard disk drive manufacturing is another prominent field seeking for the application of these knowledge intensive techniques. The manufacturing tasks that urgently require support from machine learning are in the portions of failure analysis and yield improvement. We focus our research on the yield improvement sector. Manufacturing yield prediction opens big opportunity for machine learning application because yield is a very important metric in many parts of manufacturing process. But, there rarely is research work about yield prediction in hard disk drive manufacturing found until today. So, we introduce yield prediction improvement by statistical analysis and machine learning methods including the multiple linear regression (MLR), artificial neural networks (ANN), classification and regression tree (CART). Moreover, we introduce technique to group quantity of data for yield prediction by considering consistency number, instead of grouping by calendar period as used in traditional method. The result of our technique shows the better performance. Means absolute error (MAE) of our proposal is 0.010 with a tide error rate produced by MLR and CART algorithms. The best performance from traditional calendar-based grouping is ANN algorithm with the error metric 0.017 MAE.**

***Index Terms*—Yield prediction, hard disk drive (HDD), multiple linear regression (MLR), artificial neural network (ANN), classification and regression tree (CART).**

I. Introduction

Current Hard Disk Drive (HDD) manufacturing process is characterized by capital intensity, customer reliability, and technology migration. As a result, the HDD manufacturers really need "Customer shipment planning", "Material usage planning", "Production planning", "Test capacity planning", and "Product Pricing" accurately [1]-[6]. All of these activities have "Yield Prediction" as the main concern. For example, the precise yield prediction leads to the optimal stocking of material supply quantity.

Suppose yield projection is too high from the optimistic point of view regarding the quality of the production process. Based on this yield projection scenario, we will order material supply lower than the real needs. It turns out that at the end of the production line, the number of qualified products is much less than the amount expected. The serious impact is that we will not able to supply product shipment to our customers as we had committed. On the contrary, if yield projection is too low, that mean we are pessimistic against the production quality. We thus end up over-ordering material supply. The even more serious impact is that it will reduce the cash flow of the company from inventory sink cost [7]-[9].

The method in current practice for yield prediction in HDD manufacturing is based solely on expertise of the process engineers. In recently years, the maturity of machine learning and the introduction of new techniques such as deep learning have captured the interest of engineers worldwide including those in electronics and computer parts industries like HDD manufacturing. From the literature review, we found that those state-of-art techniques are used mostly in the tasks such as failure root-cause analysis and yield improvement, while yield prediction is quite rare.

Yield prediction task in the HDD manufacturing is still using traditional method. As far as we know, there is no research work on yield prediction with machine learning technique. From the literature review, there exist many research papers making yield prediction by means of machine learning or data mining techniques in several fields such as semiconductors manufacturing, PCB (Printed Circuit Board) manufacturing, crop yield prediction, and other agricultural product yield prediction [10]-[15]. We thus propose the initiation of applying machine learning to HDD yield prediction.

In this paper, we focus on generating the HDD yield prediction models that are based on the three prominent algorithms: multiple linear regression (MLR), artificial neural network (ANN), and classification and regression tree (CART). The difficult part on modeling the HDD manufacturing yield prediction is the numerous parameters (or attributes) obtained from many processes along the production line of HDD. It is almost impossible to consider all attributes (about 400 attributes) for yield prediction task. So, we seek for collaboration from the engineering expert in real HDD industry to select from 400 attribute to be only 5 key attributes that relate to yield and failure rate concern.

The traditional yield calculation or forecasting is always based on grouping by calendar periodic such as "by day", "by week", or "by month". This method results in poor accuracy from inconsistency of quantity in each group. The products from some weeks may be of high quantity, whereas those from other weeks may be of low quantity. This inconsistency issue may influence the yield-by-week scale that we use as reference to compare with yield prediction of traditional method.

Therefore, we introduce the new quantity grouping method to improve yield prediction by grouping quantity of data into consistency numbers i.e. group of 1,000 rows, group of 5,000 rows and group of 10,000 rows (from total data containing 4,192,000 rows). According to this proposed method, we expect to mitigate the problem of quantity fluctuation.

The rest of this paper is organized as follows. In Section II, we describe the HDD background and details of yield

Manuscript received August 5, 2019; revised January 12, 2020.

The authors are with the School of Computer Engineering, Suranaree University of Technology, Nakhon Ratchasima 30000, Thailand (e-mail: anusara.hi@gmail.com, nittaya@sut.ac.th, kerdpras@sut.ac.th).

doi: 10.18178/ijmlc.2020.10.2.926

calculation in HDD production process, as well as brief introduction of the three machine learning algorithms (MLR, ANN, and CART). In Section III, we explain the material and method that we use to develop the yield prediction models. The third section also contains details of dataset, variable selection method, research framework, and research workflow. Section IV is experimentation and results. The conclusion is presented in Section V. In Section VI, we provide discussion and suggestion for future research.

II. Background Theory and Literature Review

A. Hard Disk Drive

Hard Disk Drive (HDD) is a kind of digital data recording devices. This device stores data on the durable material hard disk (or platter) by magnetic storage technology. HDD is a non-volatile storage in that stored data are still retained even when power is off [16]-[18]. Disks are paired with the heads which are used for reading and writing data on the disks. HDD is always claimed that it is the most reliable storage among various existing technologies in the data storage industry. Conventional HDD hardware consists of many components (as schematically shown in Fig. 1) and the key components are as follows.

- HSA (Head Stack Assembly) [19] is the base plate of reader/writer heads. HSA moves synchronously with the rotation speed of disk to bring the head to the location that we need to read or write the data on disk. This movement really needs the accurate calculation of movement due to the high rotation speed of disk at 5000 RPM (Round per Minute) up to 10000 RPM.
- Media, Platter or Disk [20] is the main part for storing data on the magnetic layer. The substrate layer of disk is made from materials like aluminum or glass that are high tolerant from deformation. The surface of disk is needed to be smoothening as much as possible for data recording at a very small scale.
- VCM (Voice Coil Motor) [21] is a permanent magnet component working in accordance with HSA. It is the key part that moves HSA to the desired location based on the working principle of the magnetic field.
- MBA (Motor Base Assembly) [22] is the component composing of motor and motor base plate hub. The motor is used for rotating disks with the high speed. The motor base plate hub is the strongest part of HDD used for protecting other components in the HDD from external impact force.
- PCBA (Printed Circuit Board Assemble) [23] is the key controller of HDD composed from several wired copper, controllers, and ports to connect with computer or HDD tester.

After process of assembling all components with the five steps as depicted in Fig. 1, the HDD product unit is complete. The next step is to test the mechanical and functional operation performance of HDD. In some product series that require high capacity unit, flow of test time can be as long as 1 month with more than 10 operation steps of performance test. This long testing time is because there are numerous of components in an unit of HDD and the factory must ensure that every component properly functions and operates synchronously with other components [24]-[27].

Reliability is the most important quality criteria in data storage manufacturing. Therefore, HDD makers must test every single block (the smallest unit of data storage) to ensure that it is in good condition until the end of product lifetime. The HDDs that pass the test process are called “pass units” and those that cannot pass the test process are called “fail units”.

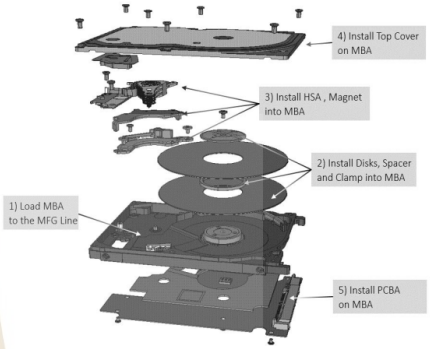


Fig. 1. Key components of hard disk drive.

B. Yield in Manufacturing

Yield in HDD manufacturing is the ration between “output units” and “input units” in the particular process. The term “units” refer to the amount of HDDs. The calculation of yield can be describe by equation 1.

$$\text{Yield} = \frac{\text{Output uantity}}{\text{Input Quantity}} \quad (1)$$

Yield in equation 1 is the calculation method of only one test operation. In real life HDD production manufacturing, there are many test operations. Diagram in Fig. 2 shows the calculation method for 3 test operations.

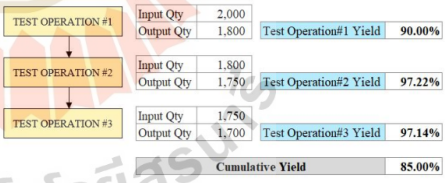


Fig. 2. The example of multi-operations yield calculation.

Fig. 2 demonstrates the example of yield computation in the multi-operation process. Input of test operation#1 (initial operation) is 2,000. Yield of test operation#1 is 90% (that is, 1,800 divided by 2,000). The input of operation#2 is exactly the number of output quantity from operation#1, which is 1,800, and the output of operation#2 is 1,750. Therefore, yield of operation#2 is 1,750/1,800 = 97.22%. In the same manner of computation, yield of operation#3 is 97.14%

(computed from 1,700 divided by 1,750). Cumulative yield of the whole test process is computed from the output quantity that pass the last test operation (that is, 1,700) divided by the input quantity of the first operation (that is, 2,000).

With this diagram, we can see that the operation test yield in one step has strong influence to operation yield in the next step. Typically, the early test operations (or first operation) almost always relate to key functional performance like mechanical test, electrical test, and basic test of read-write performance. So, the first operation always shows the lower yield than later operations. Accurately yield prediction in earlier operations certainly impacts efficiency and benefit in the planning activities of manufacturing [7]-[15].

Yield prediction is one of high concerns in HDD production and test capacity planning [28]. Precise yield prediction positively affect many activities including financial budgeting for material, production and tester capacity planning, machinery downtime schedule planning, shipment and customer delivery planning, and pricing plan in each lot of HDD products.

C. Algorithms for Yield Prediction

In the literature, linear regression is a statistical-based method popularly applied for yield prediction because of its simplicity and acceptable efficiency. However, with the advancement of machine learning technology, we consider this new technology as an interesting alternative for yield prediction in HDD industry. The two popular machine learning techniques used in our yield prediction are artificial neural network (ANN) and classification and regression tree (CART).

Linear regression analysis [29-31] is the statistical method for studying quantitatively correlation among two or more variables. One variable is defined as a target of analysis; this variable is called dependent variable, Y. Other variables are used for predicting the value of a target variable; these variables are called independent variables, X. The regression modeling is bases on mathematical calculation as shown in equation 2. The computation process is to find the best coefficient of variable X and some constant value that altogether can predict the Y value with minimal error. The equation can be plotted with linear graph. So, we called this algorithm simple linear regression analysis.

$$\hat{Y} = a + bX \quad (2)$$

when

$\hat{Y}$ is dependent variable (or target variable for prediction)

X is independent variable

a is constant of regression (or cutting point on Y axis)

b is slope of line (or regression coefficient of X)

In case of multiple input values (or multiple independent variables), the modeling will be called multiple linear regression (MLR) analysis. The computation of MLR can be described with equation 3.

$$\hat{Y} = b_0 + b_1X_1 + b_2X_2 + ... + b_kX_k \quad (3)$$

when

$\hat{Y}$ is dependent variable

$X_1, X_2, X_3, \ldots, X_k$ is a set of k independent variables

b_0 is a constant of regression (or cutting point on Y axis)

$b_1, b_2, b_3, \ldots, b_k$ is a set of line's slopes (or regression coefficients of the k independent variables)

ANN or Artificial Neural Network is the machine learning algorithm that is inspired by the biological neural networks that constitute human brains [32]-[36]. There are numerous small size neural nodes in human brain connecting together to construct the big networks with complex relation and very detailing. ANN resembles this scheme; it consists of many nodes connected with lines to compute, learn, and perform tasks. The learning is done through considering examples then adjusting weight in each connecting line to best fit the examples. The learning process can be done without programming task-specific rules. The general architecture of ANN is shown in Fig. 3. There are 3 main levels of ANN, that are, input, hidden, and output layers.

Input layer consists of input nodes and connected lines to hidden layer. The number of input nodes is equal to number of features or attributes of dataset.

Hidden layer consists of hidden nodes and connected lines to the next level. There can be more than one level in this hidden layer. Hidden layer is provided information from previous hidden layer or input layer.

Output layer consists of output nodes. The number of nodes is equal to number of values of target variable. The output nodes are always provided the information by last hidden layer.

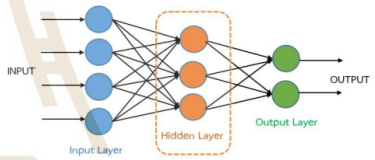


Fig. 3. Example of simple artificial neural network.

Classification and regression tree (CART) is one of the tree-based algorithms that can perform classification and prediction tasks. This algorithm is introduced by Breiman in 1984 [37]-[39]. CART is a binary decision tree in that the node can split only 2 branches. The tree is consisted of a root node (the node at level 0 in Fig. 4) and two groups of binary subtree called left subtree and right subtree. The nodes are features of data. CART uses the Gini index for feature selection in the classification task and uses sum of squared error for predicting values. The classified target or predicted value is in the leaf node.

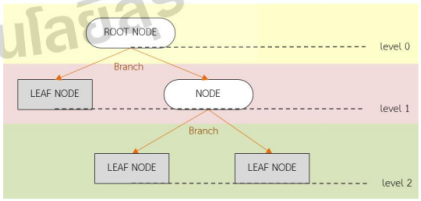


Fig. 4. Structure of CART.

D. Performance Measurement

For HDD yield prediction, we adopt mean absolute error (MAE) as the prediction performance measurement. The adopting of MAE is to comply with other yield prediction works appeared in the literature [7]-[15]. The calculation [40] of MAE can be done by averaging gap between real values of target variable and the predicted values as demonstrated in equation 4.

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i| \quad (4)$$

when

MAE is Mean Absolute Error
n is numbers of data
y_i is real value of target variable
$\hat{y}_i$ is predicted value from model

III. Material and Method

A. Dataset and Variable Selection

Our dataset is collected from the real hard disk drive production manufacturing in Thailand. The collected dataset is very big because it covers 12 months of production timeframe. Number of rows is 4,192,000 rows and number of features is more than 100.

From the assistance of engineering expert, number of features are reduced to five as they are expected to be key attributes (or features). These attributes are from type (Prime or Recycle) of key components. (PCBA Type, HSA Type, Media Type, MBA and VCM Type). Type of these key components is the main factor contributing to yield. Type "prime material" always provides the better or higher yield than "recycle material". Definition of "prime material" is the new and fresh material from suppliers and it has never been used to assemble in any HDD. Definition of "recycle material" is part of components that had been installed in some HDD. When that HDD failed in the previous test process, the reusable components will be torn down and input to rework in the recycle process.

In traditional method all rows of data are summarized in group of "date" and "week" for predicting yield. In this research, we try to mitigate the problem of quantity unit inconsistency by grouping rows of data into constant number i.e. "grouping by 1K rows", "grouping by 5K rows" and "grouping by 10K rows". Therefore, in our experiment, there are totally 5 new datasets.

B. Research Framework

Our research steps are explained in Fig. 5. Based on the objective of yield prediction performance comparison between 3 algorithms and 5 types of data grouping method, we design the research experiment according to assumption that the consistency quantity of rows (or data) in each group affects yield prediction performance.

C. Research Workflow

The flow chart in Fig. 6 depicts our research workflow. It starts by aggregating the 4.192 million rows of HDD manufacturing dataset to form the new 5 datasets: "Group of date", "Group of week", "Group of size 1K", "Group of size 5K", and "Group of size 10K". After that, we separate each 5 new datasets into 80% and 20%. The first 80% is for training the model, and the rest 20% is for testing model performance. The training datasets are input into 3 algorithms. In the performance comparison stage, we compare along the two main aspects: comparison between 5 grouping types and comparison between 3 algorithms.

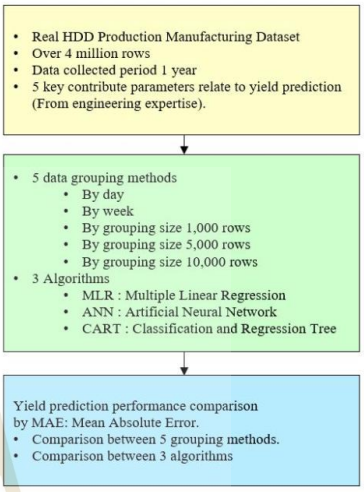


Fig. 5. Research framework.

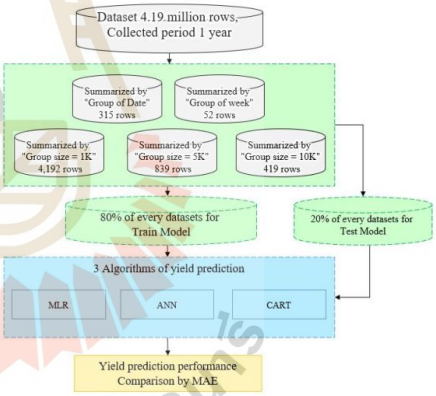


Fig. 6. Research workflow diagram

IV. Experimentation and Results

The original HDD manufacturing dataset contains 4.192 million rows. We perform five different granularity levels of data grouping: group by date, group by week, group by size of 1K, group by size of 5K, and group by size of 10K. All groups have the same set of features and number of selected features is 5. After data preparation, we modeling each dataset using 3 algorithms: MLR, ANN and CART. The

model evaluation results in terms of MAE are shown in Table 1.

On comparing model performance based on data grouping by calendar period like "By date" and "By week", the best performance is ANN with MAE on test data at 0.039 and 0.017, respectively. When considering model performance based on our proposed method for grouping data by quantity consistency with the group size of 1K, ANN shows the best performance with MAE at 0.041. We can notice that when the group size is larger (5K and 10K), the MAE drops significantly (0.019 in 5K group and 0.01 in 10K group). The best models are those from the MLR and CART algorithms, which perform slightly better than ANN.

TABLE I: YIELD PREDICTION PERFORMANCE COMPARISON BASED ON MEAN ABSOLUTE ERROR (MAE)

Grouping Method	No.	Dataset	#Rows	Learning Algorithm	Mean Absolute Error	
					Training	Testing
Traditional Method:Grouping by Calendar Periodic	1	Group by date	315	MLR	0.034	0.041
				ANN	0.032	0.039
				CART	0.033	0.043
	2	Group by week	52	MLR	0.004	0.067
				ANN	0.005	0.017
				CART	0.009	0.027
Proposed Method: Grouping by Quantity Consistency	3	Group size = 1K	4,192	MLR	0.042	0.043
				ANN	0.040	0.041
				CART	0.049	0.049
	4	Group size = 5K	836	MLR	0.017	0.019
				ANN	0.018	0.019
				CART	0.018	0.019
	5	Group size = 10K	416	MLR	0.010	0.010
				ANN	0.010	0.011
				CART	0.011	0.010

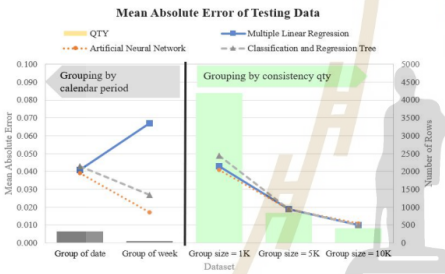


Fig. 7. Comparison of MAE (left) and number of rows (right).

For ease of interpretation, we also show comparison in graphical form in Fig. 7. A graph on the left hand side compares MAE of each learning algorithm trained with HDD production data grouped by calendar period as daily and weekly. A larger production timeframe unit from daily to weekly shows accuracy improvement on yield prediction of ANN and CART algorithms, whereas larger timeframe results in more error prediction in MLR algorithm.

A graph on the right hand side of Fig. 7 illustrates error trends of yield prediction models trained with data grouped by quantity consistency method. It can be noticed from the trend line that grouping by 10K rows provides the lowest error rate with almost equal predictive performance among the three learning algorithms. This graph also depicts the observation that with larger scale of grouping by consistency number (from 1K to 5K and then 10K), the trend of MAE is getting lower in accordance with the number of rows in the group. This is in contrast to the grouping by calendar that the prediction results of the models are quite stray and fluctuate.

V. Conclusion and Discussion

This paper introduces the case study of applying machine learning and statistical analysis techniques to predict yield in the hard disk drive (HDD) industry. We also presents the idea for yield prediction performance improvement by grouping data with consistency of quantity. Our assumption is that the fluctuation of quantity between groups has more or less influence to the low performance on yield prediction. Therefore, we design the empirical research framework by grouping data in 3 granular scales, that is, grouping of 1K rows, grouping of 5K rows, and grouping of 10K rows. The proposed method for data grouping has been experimentally compared against the traditional method that groups data either by date or by week.

The experimentation has been done with data that were collected from real life HDD manufacturing. The dataset contains over 4 million rows covering 1 year of production records. From the experimental results, we can conclude that our proposed method of grouping by quantity consistency of rows shows the better performance of yield prediction when compared against the traditional method that groups data by calendar period. The performance comparison is based on the mean absoluter error measured from yield prediction. We also notice the better trend when number of rows is getting higher. The three learning algorithms depict the same trend of this significant observation. So, these results allow us to conclude that different schemes on data grouping can result in diverse performance of HDD yield prediction.

VI. Recommendation

This research uses five key attributes as independent variables to train the learning models. The feature selection depends solely on experience of the expert engineer. In our future research work, we plan to make this step more systematic by applying the available feature selection techniques to evaluate each feature and select the most promising ones. Moreover, regarding our proposed scheme of data grouping, we plan to investigate several sizes of data

group. However, that means we are challenging by the very big data size that may contain over 10 million rows of data.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

The first author is responsible for designing the research framework, organizing the experimentation steps and preparing the manuscript. The second author helps editing the manuscript and validating the research steps. The last author helps confirming the experimental results and discussing the future research trends.

Acknowledgment

This research work has been supported by grants from the National Research Council of Thailand (NRCT). Data and Knowledge Engineering Research Unit has been fully supported by a research grant from Suranaree University of Technology.

References

[1] K. N. Malitski, *Method for Shipment Planning/Scheduling*, U.S. Patent 8,244,645, 2012.
[2] A. K. R. Katta and R. Allgor, "Heuristic methods for customer order fulfillment planning," U.S. Patent 8,352,382, Amazon Technologies Inc, 2013.
[3] M. Braglia, D. Castellano, M. Frosolini, and M.Gallo, "Overall material usage effectiveness (OME): A structured indicator to measure the effective material usage within manufacturing processes," *Production Planning & Control*, vol. 29, no. 2, pp. 143-157, 2018.
[4] J. Shi, G. Zhang, and J.Sha, 2011, "Optimal production planning for a multi-product closed loop system with uncertain demand and return," *Computers & Operations Research*, vol. 38, no. 3, pp. 641-650.
[5] A. P. Rastogi, J. W. Fowler, W. M. Carlyle, O. M. Araz, A. Maltz, and B. Büke, "Supply network capacity planning for semiconductor manufacturing with uncertain demand and correlation in demand considerations," *International Journal of Production Economics*, vol. 134, no. 2, pp. 322-332, 2011.
[6] S. M. Liozu and A. Hinterhuber, "Industrial product pricing: A value-based approach," *Journal of Business Strategy*, vol. 33, no. 4, pp. 28-39, 2012.
[7] H. Lee, C. O.Kim, H. H. Ko, and M. K. Kim, "Yield prediction through the event sequence analysis of the die attach process," *IEEE Transactions on Semiconductor Manufacturing*, vol. 28, no. 4, pp. 563-570, 2015.
[8] J. Li, X. Ji, Y. Jia *et al.*, "Hard drive failure prediction using classification and regression trees," in *Proc. 2014 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, 2014, pp. 383-394.
[9] T. Yuan, S. Z. Ramadan, and S. J. Bae, "Yield prediction for integrated circuits manufacturing through hierarchical Bayesian modeling of spatial defects," *IEEE Transactions on Reliability*, vol. 60, no. 4, pp. 729-741, 2011.
[10] T.Chen, "A heterogeneous fuzzy collaborative intelligence approach for forecasting the product yield," *Applied Soft Computing*, vol. 57, pp. 210-224, 2017.
[11] D. An, H. H. Ko, T.Gulambar, J. Kim, J. G. Baek, and S. S. Kim, 2009, "A semiconductor yields prediction using stepwise support vector machine," in *Proc. 2009 IEEE International Symposium on Assembly and Manufacturing*, pp. 130-136.
[12] K. Roy, A. Mukherjee, and D. K. Jana, "Prediction of maximum oil-yield from almond seed in a chemical industry: A novel type-2 fuzzy logic approach," *South African Journal of Chemical Engineering*, vol. 29, pp. 1-9, 2019.
[13] L. Kouadio, R. C. Deo, V. Byrareddy, *et al.*, "Artificial intelligence approach for the prediction of Robusta coffee yield using soil fertility properties," *Computers and Electronics in Agriculture*, vol. 155, pp. 324-338, 2018.
[14] T. Chen, "An ANN approach for modeling the multisource yield learning process with semiconductor manufacturing as an example," *Computers & Industrial Engineering*, vol. 103, pp. 98-104, 2017.
[15] C. Y. Lee and T. L. Tsai, "Data science framework for variable selection, metrology prediction, and process control in TFT-LCD manufacturing," *Robotics and Computer-Integrated Manufacturing*, vol. 55, pp.76-87, 2019.
[16] D. A. Patterson and J. L. Hennessy, "Computer organization and design MIPS edition: The hardware/software interface," *Newnes*, 2013.
[17] J. S. Domingo, "SSD vs HDD: What's the difference?," *PC Magazine*, 2015.
[18] N. U. Mustafa, A. Armejach, O. Ozturk, A. Cristal, and O. S. Unsal, "Implications of non-volatile memory as primary storage for database management systems," in *Proc. 2016 International Conference on Embedded Computer Systems: Architectures, Modeling and Simulation*, 2016, pp. 164-171.
[19] G. G. Foisy, N. E. Larson, S. Narayan, A. F. Sancheti, and J. Edwards, "Western Digital Corp,"*Head Stack Assembly for a Disk Drive Having a Unitary Molded Plastic e-Block*, 2000. .
[20] K. Takaishi, Y. Uematsu, T. Yamada, M. Kamimura, M. Fukushi, and Y. Kuroba, "Hard disk drive servo technology for media-level servo track writing," *IEEE Transactions on Magnetics*, vol. 39, no. 2, pp. 851-856, 2003.
[21] T. R. Simon, L. Cong, Y. Zhai, Y. Zhu, and F. Zhao, "A semi-automatic system for efficient recovery of rare earth permanent magnets from hard disk drives," *Procedia CIRP*, vol. 69, pp. 916-920, 2018.
[22] A. Al Mamun, G. Guo, and C. Bi, *Hard Disk Drive: Mechatronics and Control*, CRC press, 2017.
[23] V. W. Santini and A. D. Little, Western Digital Technologies Inc, "Disk drive including a printed circuit board assembly and a PCBA shield with tabs engaged in slots of a disk drive base," U.S. Patent 7,271,978, 2007.
[24] N. Samattapapong and N. Afzulpurkar, "A production throughput forecasting system in an automated hard disk drive test operation using GRNN," *Journal of Industrial Engineering and Management*, vol. 9, no. 2, pp. 330-358, 2016.
[25] S. Sankar, M. Shaw, K. Vaid, and S. Gurumurthi, "Datacenter scale evaluation of the impact of temperature on hard disk drive failures," *ACM Transactions on Storage (TOS)*, vol. 9, no. 2, p. 6, 2013.
[26] W. Song,A. Ovcharenko, B. Knigge, M. Yang, and F. E.Talke, "Effect of contact conditions during thermo-mechanical contact between a thermal flying height control slider and a disk asperity," *Tribology International*, vol. 55, pp. 100-107, 2012.
[27] Z. S. Ye, M. Xie, and L. C. Tang, "Reliability evaluation of hard disk drive failures based on counting processes," *Reliability Engineering & System Safety*, vol. 109, pp. 110-118, 2013.
[28] H. U. Nwosu, C. C. Obieke, and A. J. Ameh, "Failure analysis and shock protection of external hard disk drive," *Nigerian Journal of Technology*, vol. 35, no. 4, pp. 855-865, 2016.
[29] G. A. Seber and A. J. Lee, *Linear Regression Analysis*, vol. 329, John Wiley & Sons, 2012.
[30] D. C. Montgomery, E. A. Peck, and G. G. Vining, 2012, *Introduction to Linear Regression Analysis*, vol. 821, John Wiley & Sons.
[31] J.Neter, M. H. Kutner, C. J. Nachtsheim, and W.Wasserman, *Applied Linear Statistical Models*, vol. 4, p. 318, 1996, Chicago: Irwin.
[32] X.Yao, "Evolving artificial neural networks," in *Proc. the IEEE*, vol. 87, no. 9, pp. 1423-1447, 1999.
[33] A. K.Jain, J. Mao, and K. M. Mohiuddin, "Artificial neural networks: A tutorial," *Computer*, vol. 3, pp. 31-44, 1996.
[34] G. Zhang, B. E. Patuwo, and M. Y. Hu, "Forecasting with artificial neural networks: The state of the art," *International Journal of Forecasting*, vol. 14, no. 1, pp. 35-62, 1998.
[35] Y. T. Chae, R.Horesh, Y. Hwang, and Y. M. Lee, "Artificial neural network model for forecasting sub-hourly electricity usage in commercial buildings," *Energy and Buildings*, vol. 111, pp. 184-194, 2016.
[36] L. Bertinetto, J. F. Henriques, J. Valmadre, P. Torr, and A. Vedaldi, "Learning feed-forward one-shot learners," *Advances in Neural Information Processing Systems*, pp. 523-531, 2016.
[37] L. Breiman, *Classification and Regression Trees*, Routledge, 2017.
[38] J. Lepping, *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2018.
[39] Y. Y. Song and L. U. Ying, "Decision tree methods: applications for classification and prediction," *Shanghai Archives of Psychiatry*, vol. 27, no. 2, p. 130, 2015.
[40] C. J. Willmott and K. Matsuura, "Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance," *Climate Research*, vol. 30, no. 1, pp. 79-82, 2005.

A. Hirunyawanakul is a Ph.D. student in the School of Computer Engineering, Suranaree University of Technology (SUT), Thailand. She received her B.E. and M.E. in computer engineering from Suranaree University of Technology, Thailand, in 2006 and 2014, respectively. Her research of interest includes data mining, machine learning, and artificial intelligence.

N. Kerdprasop is an associate professor and the head of Data Engineering Research Unit, School of Computer Engineering, Suranaree University of Technology (SUT), Thailand. She received her B.S. in radiation techniques from Mahidol University, Thailand, in 1985, M.S. in computer science from the Prince of Songkla University, Thailand, in 1991 and Ph.D. in computer science from Nova Southeastern University, U.S.A., in 1999. Her research of interest includes data mining, artificial intelligence, logic and constraint programming.

K. Kerdprasop is an associate professor at the School of Computer Engineering, chair of the School, and the head of Knowledge Engineering Research Unit, SUT. He received his bachelor degree in mathematics from Srinakarinwirot University, Thailand, in 1986, MS in computer science from the Prince of Songkla University, Thailand, in 1991 and Ph.D. in computer science from Nova Southeastern University, U.S.A., in 1999. His current research includes machine learning and artificial intelligences.

International Journal of Electrical and Electronic Engineering & Telecommunications Vol. 9, No. 6, November 2020

Feature Selection to Improve Performance of Yield Prediction in Hard Disk Drive Manufacturing

Anusara Hirunyawanakul[1], Nuntawut Kaoungku[2], Nittaya Kerdprasop[2], and Kittisak Kerdprasop[2]
[1] School of Computer Engineering, Suranaree University of Technology, Thailand
[2] Data and Knowledge Engineering Research Unit, Suranaree University of Technology, Thailand
Email: anusara.hi@gmail.com; {nuntawut; nittaya; kerdpras}@sut.ac.th

***Abstract*—Hard Disk Drive (HDD) manufacturing is one real-world application area that machine learning has been extensively adopted for problem solving. However, most problem solving activities in HDD industry tackle on failure root-cause analysis task. Machine learning is rarely applied in a task of yield prediction. This research presents the application of machine learning and statistical techniques to select appropriate features to be used in yield prediction for the HDD manufacturing process. The seven well-known algorithms are used in the feature selection step. These algorithms are decision tree (C5 and CART), Support Vector Machine (SVM), stepwise regression, Genetic Algorithm (GA), chi-square and information gain. The two prominent learning algorithms, Multiple Linear Regression (MLR) and Artificial Neural Networks (ANN), are used in the yield prediction modeling step. Yield prediction performance has been assessed based on the two evaluation metrics: Root Mean Square Error (RMSE) and Mean Absolute Error (MAE). Yield prediction with MLR shows higher accuracy than yield estimation traditionally performed by human engineers. Resulting to conclusion that the proposed novel learning steps can help HDD process engineers to predict yield with the better performance, especially on applying GA as feature selection tool, the MAE is reduced from 0.014 (yield estimated by human engineer) to 0.0059 (yield predicted by MLR). That means error reduction is about 60%.**

***Index Terms*—Artificial neural network, feature selection, genetic algorithm, hard disk drive, multiple linear regression, yield prediction**

I. Introduction

Hard Disk Drive (HDD) is still be the most important data storage device and more preferred in the current era of big data than Solid State Drives (SSD). This is due to the two key factors, reliability of data retention and cost per terabyte that makes HDD clearly better than SSD. Even though SSD and flash memory are commonly used in personal digital devices, the enormous amount of data is stored in the server farms driven by ensemble of a lot of HDDs [1]-[2].

The production process of HDDs consists of many steps. Every single unit of a complete HDD is assembled from several components and required significant periods of time to check its quality. In some product series of HDDs with high capacity (such as 14TB or 12TB), the quality control may consume time of test process for over 3 months. The HDDs that pass the test process are called the "passed units," whereas the rejected HDDs are called the "failed units." It is certain that manufacturing factory prefers to produce "passed units" as much as possible. The metric to measure efficiency of production is "passed units" per "input units." This metric is commonly referred to as "yield" [3]-[5].

HDD manufacturers have to make strategic planning such as production manufacturing line planning, material usage planning, tester and machinery capacity planning, and shipment planning accurately. All of these planning activities involve the action of "yield prediction". For example, the precise yield prediction leads to suitable stocking material and optimal workforce and tester capacity plan. In current HDD manufacturing, typical method for yield prediction practice is based on personal experiences of process engineers. Even though many machine learning techniques are applied to assist the HDD manufacturing, they are focused on only failure analysis task. As far as we know, there are no application to the yield prediction task.

The most difficult portion of yield prediction task is the excessive amount of attributes obtained from several steps and components along the assembly and test process of HDD production manufacturing line. At least over 100 attributes are generated for a unit of HDD. It is likely impossible to compute and consider all attributes in the modeling step of yield prediction task [6]-[11].

In this paper, we thus propose data preparation and feature selection methods with the main focus to improve modeling performance on predicting yield. Many machine learning algorithms and techniques are used in this paper including support vectors machine (SVM), classification and regression tree (CART), C5, feature selection by considering chi-square and information gain.

Manuscript received December 30, 2019; revised January 15, 2020; accepted March 16, 2020.
Corresponding author: Anusara Hirunyawanakul (email: anusara.hi@gmail.com).
This research work has been supported by grants from the National Research Council of Thailand (NRCT). Data and Knowledge Engineering Research Unit has been fully supported by a research grant from Suranaree University of Technology.

doi: 10.18178/ijeetc.9.6.420-428

Our intuitive idea is to adopt these algorithms and techniques to select only important attributes to use in yield prediction step and expect to see the better performance of yield prediction.

In the part of yield prediction, we use two algorithms: Multiple Linear Regression (MLR) and Artificial and Neural Networks (ANN). Dataset used in our experiments contains around 1,000,000 records of HDD units that had been tested within one year. We group these million records into 1,000 rows of 10,000 records per group and use this new dataset for yield prediction step.

The next section of this paper presents briefly the background information regarding HDD components, the HDD production steps, and yield calculation in HDD manufacturing. Feature selection algorithms, yield prediction algorithm and evaluation method are also described in this section as well. In Section III, we explain material and methods that we use to develop the feature selection models. Section IV explains our experimental setting and results. The conclusion of this paper is in section V. Finally, in section VI, we provide the suggestion and recommendation on applying our idea.

II. Background and Theory

A. Hard Disk Drive (HDD)

HDD is a digital data storage device which records data on the durable platter (or hard disk) by magnetic recording technology. HDD is non-volatile storage device, which means HDD is able to store data even if power is off [12]-[14]. HDD consists of numerous important hardware components working together in a synchronized manner. Synchronization speed can affect the read-write performance. The fundamental components of HDD are shown in Fig. 1 and can be described as follows:

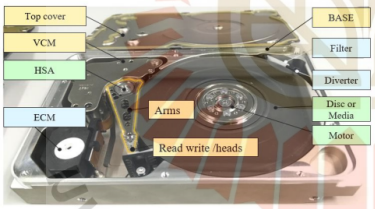


Fig. 1. Key components of a hard disk drive.

1) HSA (head stack assembly) [15] is the assembled part of reader/writer heads and base plate of the head components. HSA moves synchronously with rotation motor speed which drives the disk.

2) Media, disk or platter [16] is the key component for storing data. The digital data are written down from magnetic head to magnetic layer of disk. The substrate of disk component must be casted from durable and very smooth material such as aluminum or glass.

3) Motor base assembled (MBA) [17] is a component that consists of motor and motor hub plate. The key function of motor is to rotate the disk with consistency speed according to the read-write speed of each product revision. Motor hub plate is the strongest component of HDD because its function is to protect other components from external force.

4) Voice coil motor (VCM) [18] is consisted of two pieces of permanent magnet. This component works together with HSA to move HSA to the desired area for reading/writing data. This component function is working based on principle of magnetic field.

5) Printed circuit board assembled (PCBA) [19] is the controller of HDD. It is composed of several circuit wires, capacitance and microcontroller chip. Key function of this component is to communicate to computer or tester slot.

There exist other components that are also important such as environment control module (ECM) for controlling environment in the internal closed humidity and air flow HDD, Recirculate Filter for filtering out small particles from contaminating the HDD, Top-cover for sealing and enclosing the HDD. There are many other components with the main function to deliver the most reliable data storage as much as possible.

All of these components are assembled in the clean room called class-100, which is control count of particle size 500 nm to be lower than 100 counts. After completing the assembled process, the HDD is input into the "test process" to validate that the completed HDD is working properly in terms of mechanical and electrical properties, data storage, read-write performance, degradation and cosmetic outlook. In some product series that have high capacity in a unit, a complete flow of test process can be longer than 40 days with over 10 operation steps of test. The very long testing time is because the HDD maker must ensure that each of the many components work properly not only on a function of itself, but also operates synchronously with other components. The HDD that passes the test process is called the "pass unit," while the one rejected by test process is called the "fail unit."

B. Yield Definition and Its Calculation

Definition of "yield" in HDD manufacturing is simple and straightforward. It is a ratio between "pass unit" and "input unit" in the particular process or step. The calculation of yield is described as

$$\text{Yield} = \frac{\text{Quantity of pass units}}{\text{Quantity of input units}} \quad (1)$$

Every single HDD is composed of many components that are assembled through many steps. Therefore, there are many attributes generated from each steps of the assembly process. These attributes are important factors in yield calculation. Thus, prior to the modeling process for yield prediction, one necessary task is to find and select only the important attributes for yield prediction.

C. Feature Selection

Feature selection is an important data preparation step before employing any machine learning algorithms or

statistical analysis methods [20]-[22]. The objective of feature selection is to reduce the dimension of data to a manageable and computational size. Fundamental idea of feature selection is to find only the most powerful and discriminative attributes from many existing attributes or features. Feature selection techniques can be categorized into two major types: filter and wrapper.

Filter method performs attribute selection as a preprocessing step independent from a modeling step. Attributes are evaluated to select only the ones expected to have the most impact or importance on predicting the target attribute. Then those attributes will be prioritized in descending order. The threshold will be determined. If any of the important features do not reach the specified threshold, they will be discarded because of the assumption that they are not important enough. This method can be done in both univariate filter method and multivariate filter method. The most popular criteria for feature selection are chi-square and information gain. The advantages of filter method are simple calculation steps, fast computation, and the avoidance of overfitting problem.

Wrapper method, on the contrary, is tightly couple to the modeling step. The principle of this method is to take multiple features into consideration in the format of a set of features. Then, the modeling algorithm tries to find the feature set showing the most important association to the output feature. After the best feature set has been found, the learning algorithm will use this set in the subsequent step. There are 2 subtypes of wrapper method:

1) Forward stepwise is to continuously add more features one by one until the modeling algorithm can get the best set of attributes.

2) Backward stepwise is to put all the features in the set and then continuously take feature out of the set one by one until the best set is achieved.

The most popular methods for of wrapper feature selection method are stepwise regression and genetic algorithm.

The advantages of the wrapper method are simplicity and high efficiency. The disadvantage is that this method takes more time than the filter method and may cause overfitting problem.

D. Algorithm for Feature Selection: GA

Genetic Algorithm (GA) [23]-[25] is a technique for finding an optimal solution or approximate answers to a problem based on the theory of evolution from biology and natural selection. That is, the most suitable organisms can survive. GA consists of 5 main functions:

1) Chromosome encoding is taking the features of possible answers into a form of chromosome.

2) Initial population is to define the number of populations that we would like to create, usually done by randomized. Then, chromosomes are randomly generated by that amount.

3) Fitness function is to identify the function to be used for determining which chromosomes should go to next round. The criteria for justification will be different for each problem.

4) Genetic operator (selection, crossover and mutation) is a method of adjusting the structure of the chromosomes for the next model.

5) Termination is the function to define the point that we are satisfied, such as the best fitness score is achieved or the score is steady for many generations consecutively.

E. Algorithm for Feature Selection: Decision Tree

Decision Tree is a widely known algorithm for data classification algorithm. The concept of decision tree is finding the pattern of the attribute that needs to be classified. The structure of the data classification model is in the hierarchy [26], [27] and represented as a tree. The tree is consisted of nodes and branches. Nodes can be divided into 3 types as root node, internal node and leaf node as shown in Fig. 2.

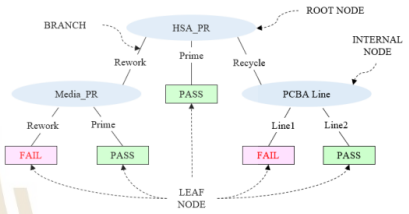


Fig. 2. Example structure of decision tree in HDD manufacturing.

Root nodes and internal nodes are features that the learning algorithm selects for being decision criteria for splitting data of mixed classes to be data subsets with more purity of class mixture. The root node is the initial feature chosen by the algorithm to create a tree. The next step is creating the branches of the root node. The number of branches will equal to all possible values of the features selected as the root node. If any child node contains data having the same class, then that node becomes a leaf node. Conversely, if data in the child node are of mixed classes, the tree is growing by repeating the splitting process until all leaf nodes are of homogeneous class or until some stopping criterion has been met. In this work, we apply the decision tree as one of our feature selection methods. There are many criteria for splitting nodes in a decision tree as shown in Table I.

TABLE I: EXAMPLE OF ALGORITHMS AND CRITERIA FOR DECISION TREE MODELING

Criteria for construct decision tree	Algorithm name
Information Gain	ID3, C4.5, C5.0
Gini Index	CART
Chi-Square	CHAID
Variance Reduction	CART

F. Algorithm for Yield Prediction: MLR

Regarding to literature review on yield prediction, the one prominent algorithm is linear regression because of its simplicity and predictive performance. Linear regression [28]-[30] is the statistical method seeking for quantitative correlation among two or more variables. One variable has to be defined as a target of analysis; this variable is called dependent variable, denoted with a common symbol Y. The other variables are used for

predicting the value of a target variable; these variables are called independent variables, denoted as X_i, when i is 1 to k for the case that there exist k independent variables. The modeling of linear regression is based on the calculation defined as

$$\hat{Y} = a + bX \tag{2}$$

where $\hat{Y}$ is the dependent variable (or target variable for prediction), X is the independent variable, a is the constant of regression (or cutting point on Y axis), and b is the slope of a line (or regression coefficient of X).

In (2) we assume there is only one independent variable. The computation processing of this linear regression is to find the best coefficient of variable X and some constant value to predict the value of variable Y with least error. The relation of variable X and Y can be plotted with linear graph. This computation is also called a simple linear regression analysis.

In case of multiple independent variables, the modeling will be called multiple linear regression (MLR) analysis. The computation of MLR can be done by

$$\hat{Y} = b_0 + b_1X_1 + b_2X_2 + \cdots + b_kX_k \tag{3}$$

where $\hat{Y}$ is the dependent variable, (X_1, X_2, ···, X_k) is a set of k independent variables, b_0 is a constant of regression (or cutting point on Y axis), and (b_1, b_2, ···, b_k) is a set of line's slopes (or regression coefficients of the k independent variables).

G. Algorithm for Yield Prediction: ANN

Besides linear regression analysis, machine learning is another new technology being adopted as an interesting alternative method for yield forecasting. The most popular machine learning technique used in yield prediction is artificial neural network (ANN). Popularity is due to its outstanding performance. ANN is machine learning algorithm that is inspired by the biological neural networks of brains [31]-[33]. There are plenty of small size neural nodes in human brain that are connected together to construct the considerable networks with complexity relationship. ANN consists of many nodes connecting with lines to compute, learn, and operate specific task. The learning and computation will be done by considering training examples then adjusting weight in each connecting line for the optimum result of predicting value of a target variable. This learning process is self-learning model; that means result can be provided without programming specific rules. The diagram in Fig. 3 shows general architecture of ANN. There are 3 majority layers: input, hidden, and output layers.

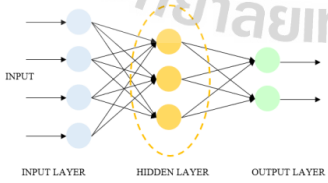


Fig. 3. General structure of simple Artificial Neural Network

1) Input layer consists of input nodes with the number of nodes equals to number of features of a dataset. All of input nodes are connected to hidden layer.

2) Hidden layer consists of hidden nodes with lines connecting to the next level. There can be one or more levels in this hidden layer. Hidden layer is provided information from the nodes in previous hidden layer or input layer.

3) Output layer consists of output nodes. The number of nodes equals to number of values of target variable. The output nodes are always provided the information by the last hidden layer.

H. Performance Evaluation: MAE and RMSE

To evaluate yield prediction performance, we use mean absolute error (MAE) and root mean square error (RMSE) as the measurement tools. MAE and RMSE are typical measurement metrics in yield prediction and many other fields [3]-[5]. The calculation of MAE can be done by averaging differences between actual values of target variable and the predicted values, defined as

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{4}$$

where MAE stands for the mean absolute error, n is the numbers of data, y_i is the real value of target variable, and $\hat{y}_i$ is the predicted value made by the model.

RMSE uses the same concept as MAE but the computation is slightly different in that RMSE is to find square root of average of differences between real value of target variable and predicted value power by 2. The formula is provided as

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{5}$$

where RMSE stands for the root mean square error.

III. Material and Method

A. Dataset

The dataset used in this research has been collected from the real manufacturing of hard disk drive. The time frame of data collection is 3 months of HDD production. The number of record (or rows) is 1,000,000 rows and number of features (or attributes) is more than 100.

Attributes are the information recorded in the production and test process for every individual HDD unit, as shown in Fig. 4. The 12 attributes given in Fig. 4 are described as follows:

1) Drive serial number (drive SN): This attribute is identification number of each HDD lot. This number is unique for each HDD lot.

2) WEEK: This is the fiscal week that the particular HDD had been assembled.

3) STATUS: This attribute records the status of test process. There are only 2 possible values: Pass and fail. The "pass" status indicates that this HDD passed the test process and be able to be input of the next operation step or ready to ship to customer. The "fail" status means this

HDD is rejected from the test process and must go to either "rework", "retest", "recycle" or "scrap" process according to the debug diagnostic failure symptom.

4) HSA prime-rework status (HSA_PR): This attribute reveals the condition of HSA component. The two possible values of this component are prime and rework. "Prime" means this HSA is the fresh new built component and never been installed in any other HDD before. "Rework" means this HSA is a component that had been installed in another HDD, but that HDD had been rejected in the test process with the HSA labeled as rework. Thus, this HSA is recycled by being rebuilt again in this HDD. (Note that definitions of Prime and Rework are also used in the attributes 5 through 9.)

5) Media prime-rework status (media_PR): This attribute is either the prime or rework condition of media.

6) MBA prime-rework status (MBA_PR): This attribute refers to the prime or rework condition of motor base assembled.

7) VCM prime-rework status (VCM_PR): This attribute describes the prime or rework condition of VCM.

8) TC prime-rework status (TC_PR): This attribute is the prime or rework condition of Top cover.

9) PCBA prime-rework status (PCBA_PR): This attribute is the prime or rework condition of PCBA.

10) DB_Line: This attribute is the identification number of the HDD assembly line.

11) HSA_Line: This attribute is the identification number of the HSA assembly line.

12) PCBA_Line: This attribute reveals the production line for installing PCBA into the HDD.

Drive SN	WEEK	STATUS	HSA_PR	MEDIA_PR	MBA_PR	VCM_PR	TC_PR	PCBA_PR	DB_LINE	PCBA_LINE	HSA_LINE
SN0000001	W01	Pass	Prime	Prime	Prime	Prime	Prime	Prime	DB_1	PCBA_2	HSA_3
SN0000002	W01	Pass	Prime	Prime	Prime	Prime	Prime	Prime	DB_1	PCBA_2	HSA_3
SN0000003	W01	Fail	Prime	Rework	Prime	Prime	Prime	Prime	DB_1	PCBA_2	HSA_3
SN0000004	W01	Fail	Rework	Rework	Prime	Prime	Prime	Prime	DB_1	PCBA_2	HSA_3
SN0000005	W01	Pass	Prime	Prime	Prime	Prime	Prime	Prime	DB_1	PCBA_2	HSA_1
SN0000006	W01	Pass	Prime	Prime	Prime	Prime	Prime	Prime	DB_1	PCBA_2	HSA_1
SN0000007	W01	Pass	Prime	Prime	Prime	Prime	Prime	Prime	DB_1	PCBA_2	HSA_1
SN0000008	W01	Pass	Prime	Prime	Prime	Prime	Prime	Prime	DB_1	PCBA_1	HSA_1
SN0000009	W01	Pass	Prime	Prime	Prime	Rework	Prime	Prime	DB_1	PCBA_2	HSA_1
SN0000010	W01	Pass	Prime	Prime	Prime	Prime	Rework	Rework	DB_2	PCBA_3	HSA_1
SN0000011	W02	Fail	Prime	Rework	Rework	Prime	Rework	Rework	DB_2	PCBA_4	HSA_2
SN0000012	W02	Pass	Prime	Prime	Prime	Prime	Prime	Prime	DB_2	PCBA_5	HSA_2
SN0000013	W02	Pass	Prime	Prime	Prime	Prime	Prime	Prime	DB_2	PCBA_6	HSA_2
SN0000014	W02	Pass	Prime	Prime	Prime	Prime	Prime	Prime	DB_2	PCBA_7	HSA_2
SN0000015	W02	Pass	Prime	Prime	Prime	Prime	Prime	Prime	DB_2	PCBA_8	HSA_2
SN0000016	W03	Pass	Prime	Prime	Prime	Prime	Prime	Prime	DB_2	PCBA_9	HSA_2
SN0000017	W03	Pass	Prime	Prime	Prime	Prime	Prime	Rework	DB_2	PCBA_10	HSA_2
SN0000018	W03	Pass	Prime	Prime	Prime	Prime	Prime	Prime	DB_2	PCBA_11	HSA_2
SN0000019	W03	Pass	Prime	Prime	Rework	Rework	Rework	Prime	DB_2	PCBA_12	HSA_2
SN0000020	W03	Fail	Rework	Prime	Prime	Rework	Prime	Prime	DB_1	PCBA_13	HSA_3

Fig. 4. Example table to show attributes in a hard disk drive manufacturing being grouped by unit.

Beside these main attributes, there are also many other attributes with the diverse meanings and important in a unit of HDD. Total number of attributes used in our experiment is 125.

B. Feature Selection Step

Objective of this research is the improvement of yield prediction accuracy by focusing on feature selection part. That means we expect the better performance of yield prediction model built from applying various feature selection techniques.

The feature selection techniques based on machine learning algorithms are C5, CART, SVM, Stepwise Regression, and GA. Moreover, we also include the two techniques based on statistics like correlation filter and chi-square filter. Feature selection by human expert is also used as a baseline for performance comparison. The process engineers select only 5 attributes. These key attributes are considered important based on long-term experience of the engineers.

C. Research Framework

Experimentation steps from data collection, feature selection, data aggregation until yield prediction modeling are schematically displayed in Fig. 5. The 7 feature selection methods are experimented and compared against the key attribute selection method used by the engineers.

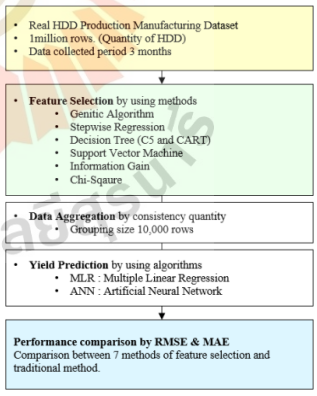


Fig. 5. Research framework

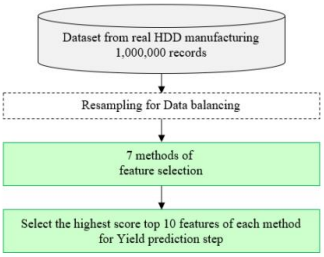


Fig. 6. Research workflow for the part of feature selection

D. Research Workflow

Flow chart diagram of Fig. 6 depicts the research workflow in a specific part of feature selection. The experiment starts by resampling to make data balancing. Data balancing is necessary because from observing the original data containing totally 1,000,000 records, we found that the target class (status) of dataset is skewed with higher number of pass than fail. The imbalance ratio between majority (pass) and minority (fail) is about 28:1. After data balancing step, each of the 7 feature selection methods are applied to select features from 125 attributes. These methods return the result by ranking the important factors in descending order. Then, we select the top 10 attributes of each method to be used further in the next step of modeling to create a yield prediction model.

In yield prediction modeling, the process starts by aggregating the 1,000,000 data records to become 100 rows in which each row contains 10,000 records. This aggregation steps is for accuracy improvement as we obse5rved from our preliminary experiments. After data aggregation step, we obtain new 7 datasets, each of which is a dataset with 100 rows and 10 attributes that are selected from 7 methods of feature selection. That means these new datasets have 100 rows aggregated from the original 10,000 record with ten attributes that can be different from one dataset to the others because they are selected with different methods.

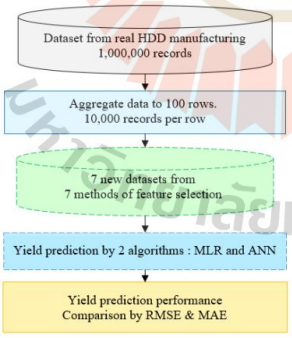


Fig. 7. Research workflow for the part of data aggregation and yield prediction modeling

After that, the learning algorithms MLR and ANN are applied to create yield prediction models. Finally, yield prediction performances are compared among 7 methods of feature selection and human selection method done by the process engineers. All of these steps are depicted as a workflow diagram and shown in Fig. 7.

IV. Experimental Results

The main focus of our experimentation is to study performances of 7 different methods for feature selection that should adopted for data preparation step prior to building the model to predict pass/fail of the assembled HDD units. The accuracy of pass/fail prediction is important for yield computation. The more accurate Pass/Fail prediction, the more precise yield estimation.

The 7 methods for feature selection used in this work are C5, CART, SVM, GA, stepwise regression, chi-square, and Information Gain. The first five algorithms are also learning algorithms, whereas the last two are only for feature ranking and selection. Thus, we firstly, apply the 7 algorithms for selecting the top-10 features anticipating to contribute the most toward yield prediction through the accurate forecasting of the HDD status as either pass or fail. The five algorithms that are both capable of feature selection and learning to build model are applied for both jobs. Their accuracies are computation time are reported and shown in Table II. The two algorithms (chi-square and information gain) that can only be used for feature selection are reported just for their computation time.

It can be observed from the results that the three models with highest pass/fail prediction accuracy are those built by C5, CART and SVM, respectively. In terms of computation time, chi-square and information gain show outstanding shorter time (1 second). GA takes the longest time of feature selection step at 4,620 seconds.

Table II: Accuracy and Computation Time of 7 Methods in Feature Selection Step

Feature Selection Methods	Accuracy (%)	Time (sec)
C5	72.38	646.8
CART	66.57	31.7
SVM	64.62	1,216.8
Stepwise Regression	64.76	56.5
Genetic Algorithm	61.94	4,620.2
Chi-Square	-	1.0
Information Gain	-	1.0

For the next part of our experimentation based on the dataset with selected features, yield has been computed as: (quantity of pass units) / (quantity of input units). Actual yield values and predicted yields made by the learning algorithms (MLR and ANN) are compared and the prediction errors are shown in Table III. At this step, yield prediction results based on the five key features selected by human experts are also shown in the first row of the table as a baseline for performance comparison.

TABLE III: ACCURACY AND COMPUTATION TIME OF 7 METHODS IN FEATURE SELECTION STEP

Feature selection method	Yield prediction algorithm	Test data	
		RMSE	MAE
Human Engineers	Traditional	0.01700	0.01400
C5	MLR	**0.00866**	**0.00605**
	ANN	0.01707	**0.01263**
CART	MLR	0.24105	0.05913
	ANN	**0.01630**	**0.01251**
SVM	MLR	0.02037	**0.01247**
	ANN	0.01864	**0.01384**
Stepwise Regression	MLR	0.10326	0.02842
	ANN	0.01851	**0.01306**
Genetic Algorithm	MLR	**0.00732**	**0.00559**
	ANN	0.01706	**0.01269**
Chi-Square	MLR	**0.00821**	**0.00690**
	ANN	0.01707	**0.01262**
Information Gain	MLR	**0.00821**	**0.00690**
	ANN	0.01707	**0.01262**

TABLE IV: ERROR REDUCTION FROM TRADITIONAL ENGINEERING METHOD

Feature selection and model building scheme	Error reduction	
	RMSE	MAE
C5 with MLR	-49%	-57%
C5 with ANN	0%	-10%
CART with MLR	1318%	322%
CART with ANN	-4%	-11%
SVM with MLR	20%	-11%
SVM with ANN	10%	-1%
Stepwise with MLR	507%	103%
Stepwise with ANN	9%	-7%
GA with MLR *	-57%	-60%
GA with ANN	0%	-9%
Chi-Square with MLR	-52%	-51%
Chi-Square with ANN	0%	-10%
Information Gain with MLR	-52%	-51%
Information Gain with ANN	0%	-10%

In terms of RMSE, statistical feature selection methods like chi-square and information gain when modeling with MLR perform better than feature selection made by human engineers. However, when building the model with ANN, both methods are as good as the human expert. C5 and GA are also comparable to traditional method when using ANN yield prediction model.

It can be seen from the results that feature selection with GA and then building the model with MLR yield the best result with least RMSE value at 0.00732. Comparing to traditional method and feature selection made by human expert, the combination of GA and MLR can significantly improve yield prediction performance with error reduction around 57% (as shown in Table IV).

When considering from the MAE metric with error reduction computed by using traditional method with human selected features as summarized in Table IV, it can be seen that almost all machine learning based and statistical based modeling methods with the base value of MAE = 0.014. This is except the two combinations, CART + MLR and stepwise regression + MLR that perform worse than human feature selection + traditional method. The best yield prediction scheme in terms of MAE metric is Genetic Algorithm with MLR prediction model.

V. CONCLUSION

This paper introduces the novel idea of applying machine learning and statistical analysis techniques in feature selection part to improve performance of yield prediction in the Hard Disk Drive (HDD) manufacturing. The assumption of this research is that the number of features from HDD manufacturing is typically numerous and thus prediction performance can be lessen by the shadow of too many features. We propose that proper feature selection technique can help improving yield prediction by selecting only key important features. Efficiency of this proposal has been confirm through experiments with the real-world data collected from HDD manufacturing containing 1 million records and 125 attributes. The experiments are done by applying 7 methods of feature selection and the yield prediction models are built from the two learning algorithms.

The experimental results demonstrate that in terms of RMSE metric, the 4 from 7 feature selection methods in combination with the MLR learning algorithm can help improving yield prediction performance. In terms of MAE metric, all 7 feature selection methods in combination with the ANN learning algorithm can improve yield prediction. The best combination is GA and MLR can improve performance when compared against traditional method that required human engineers to select key features the improvement is as high as 57%. However, the trade-off from using GA is the long computation time. These results lead to conclusion that the proposed novel idea of combining feature selection technique with powerful learning algorithm can help improving yield prediction performance in the real application of HDD manufacturing.

RECOMMENDATION

The dataset used in this research had been collected from 3 months of production timeframe in the steady and maturity performance phase. Yield computation of this dataset gives the results that are quite stable with low fluctuation. In the future, researchers and engineering experts in HDD manufacturing agree to make some challenging advancement by using dataset of "developing phase" instead of "maturity phase". This challenge can gain more benefit because of the successful result in data of developing phase can help manager to prepare good action in mass production of maturity phase.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

The first author is responsible for designing the research framework, organizing the experimentation steps

and preparing the manuscript. The second author advice and help in developing programs with R and Python languages. The third author helps editing the manuscript and validating the research steps. The last author helps confirming the experimental results and discussing the future research trends.

REFERENCES

[1] R. Wood, "Future hard disk drive systems," *Journal of Magnetism and Magnetic Materials*, vol. 321, no. 6, pp. 555-561, 2009.

[2] V. Kasavajhala "Solid state drive vs. hard disk drive price and performance study," *A Dell Technical White Paper*, Dell PowerVault Storage Systems, 2011, pp. 1-13.

[3] H. Lee, C. O. Kim, H. H. Ko, and M. Kim, "Yield prediction through the event sequence analysis of the die attach process," *IEEE Trans. on Semiconductor Manufacturing*, vol. 28, no. 4, pp. 563-570, 2015.

[4] J. Li, X. Ji, Y. Jia, *et al*., "Hard drive failure prediction using classification and regression trees," in *Proc. 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, 2014, pp. 383-394.

[5] T. Yuan, S. Z. Ramadan, and S. J. Bae, "Yield prediction for integrated circuits manufacturing through hierarchical Bayesian modeling of spatial defects," *IEEE Trans. on Reliability*, vol. 60, no. 4, pp. 729-741, 2011.

[6] K. N. Malitski, "Method for shipment planning/scheduling," Patent, US8244645, 2012.

[7] A. K. R. Katta and R. Allgor, "Heuristic methods for customer order fulfillment planning," Patent, US8352382, 2013.

[8] M. Braglia, D. Castellano, M. Frosolini, and M. Gallo, "Overall material usage effectiveness (OME): A structured indicator to measure the effective material usage within manufacturing processes," *Production Planning & Control*, vol. 29, no. 2, pp. 143-157, 2018.

[9] J. Shi, G. Zhang, and J. Sha, "Optimal production planning for a multi-product closed loop system with uncertain demand and return," *Computers & Operations Research*, vol. 38, no. 3, pp. 641-650, 2011.

[10] A. P. Rastogi, J. W. Fowler, W. M. Carlyle, O. M. Araz, A. Maltz, and B. Büke, "Supply network capacity planning for semiconductor manufacturing with uncertain demand and correlation in demand considerations," *International Journal of Production Economics*, vol. 134, no. 2, pp. 322-332, 2011.

[11] S. M. Liozu and A. Hinterhuber, "Industrial product pricing: a value-based approach," *Journal of Business Strategy*, vol. 33, no. 4, pp. 28-39, 2012.

[12] D. A. Patterson and J. L. Hennessy, *Computer Organization and Design ARM Edition: The Hardware Software Interface*, Morgan Kaufmann, 2016.

[13] J. S. Domingo. (January 25, 2019). SSD vs. HDD: What's the difference. *PC Magazine*. [Online]. Available: https://sea.pcmag.com/storage/1526/ssd-vs-hdd-whats-the-difference

[14] N. U. Mustafa, A. Armejach, O. Ozturk, A. Cristal, and O. S. Unsal, "Implications of non-volatile memory as primary storage for database management systems," in *Proc. Int. Conf. on Embedded Computer Systems: Architectures, Modeling and Simulation (SAMOS)*, 2016, pp. 164-171.

[15] G. G. Foisy, N. E. Larson, S. Narayan, A. F. Sancheti, and J. Edwards, "Head stack assembly for a disk drive having a unitary molded plastic E-block, " Patent, US6061206, 2000.

[16] K. Takaishi, Y. Uematsu, T. Yamada, M. Kamimura, M. Fukushi, and Y. Kuroba, "Hard disk drive servo technology for media-level servo track writing," *IEEE Trans. on Magnetics*, vol. 39, no. 2, pp. 851-856, 2003.

[17] A. A. Mamun, G. Guo, and C. Bi, *Hard Disk Drive: Mechatronics and Control*, CRC press, 2017.

[18] T. R. Simon, L. Cong, Y. Zhai, Y. Zhu, and F. Zhao, "A semi-automatic system for efficient recovery of rare Earth permanent magnets from hard disk drives," *Procedia CIRP*, vol. 69, pp. 916-920, 2018.

[19] V. W. Santini and A. D. Little, "Disk drive including a printed circuit board assembly and a PCBA shield with tabs engaged in slots of a disk drive base," Patent, US7271978, 2007.

[20] H. Turabieh, M. Mafarja, and X. Li, "Iterated feature selection algorithms with layered recurrent neural network for software fault prediction," *Expert Systems with Applications*, vol. 122, pp. 27-42, 2019.

[21] A. Suppers, A. J. V. Gool, and H. J. Wessels, "Integrated chemometrics and statistics to drive successful proteomics biomarker discovery," *Proteomes*, vol. 6, no. 2, pp. 20, 2018.

[22] H. Liu and M. Zhou, "Decision tree rule-based feature selection for large-scale imbalanced data," in *Proc. 26th Wireless and Optical Communication Conference*, 2017, pp. 1-6.

[23] H. Frohlich, O. Chapelle, and B. Scholkopf, "Feature selection for support vector machines by means of genetic algorithm," in *Proc. 15th IEEE International Conference on Tools with Artificial Intelligence*, 2003, pp. 142-148.

[24] J. Huang, Y. Cai, and X. Xu, "A hybrid genetic algorithm for feature selection wrapper based on mutual information," *Pattern Recognition Letters*, vol. 28, no.13, pp. 1825-1844, 2007.

[25] M. Anbarasi, E. Anupriya, and N. C. S. N. Iyengar, "Enhanced prediction of heart disease with feature subset selection using genetic algorithm," *International Journal of Engineering Science and Technology*, vol. 2, no. 10, pp. 5370-5376, 2010.

[26] G. Stein, B. Chen, A. S. Wu, and K. A. Hua, "Decision tree classifier for network intrusion detection with GA-based feature selection," in *Proc. 43rd Annual Southeast Regional Conference*, vol. 2, 2005, pp. 136-141.

[27] R. Pandya and J. Pandya, "C5.0 algorithm to improved decision tree with feature selection and reduced error pruning," *International Journal of Computer Applications*, vol. 117, no. 16, pp. 18-21, 2015.

[28] G. A. Seber and A. J. Lee, *Linear Regression Analysis*, John Wiley & Sons, 2012.

[29] D. C. Montgomery, E. A. Peck, and G. G. Vining, *Introduction to Linear Regression Analysis*, John Wiley & Sons, 2012.

[30] J. Neter, M. H. Kutner, C. J. Nachtsheim, and W. Wasserman, *Applied Linear Statistical Models*, Chicago, 1996.

[31] G. Zhang, B. E. Patuwo, and M. Y. Hu, "Forecasting with artificial neural networks: The state of the art," *International Journal of Forecasting*, vol. 14, no. 1, pp. 35-62, 1998.

[32] Y. T. Chae, R. Horesh, Y. Hwang, and Y. M. Lee, "Artificial neural network model for forecasting sub-hourly electricity usage in commercial buildings," *Energy and Buildings*, vol. 111, pp. 184-194, 2016.

[33] L. Bertinetto, J. F. Henriques, J. Valmadre, P. Torr, and A. Vedaldi, "Learning feed-forward one-shot learners," in *Proc. Advances in Neural Information Processing Systems*, 2016, pp. 523-531.

A. Hirunyawanakul is a Ph.D. student, School of Computer Engineering, Suranaree University of Technology (SUT), Thailand. She received her B.E. and M.E. in computer engineering from Suranaree University of Technology, Thailand, in 2006 and 2014. Her research of interest includes Data Mining, Machine Learning, and Artificial Intelligence.

N. Kaoungku is currently a lecturer at School of Computer Engineering, Suranaree University of Technology (SUT), Thailand. He received his bachelor, master, and doctoral degrees in Computer Engineering from SUT in 2012, 2013, and 2015, respectively. His current research work includes Data Mining, Knowledge Engineering, and Semantic Web.

N. Kerdprasop is an associate professor and the head of Data Engineering Research Unit, School of Computer Engineering, Suranaree University of Technology (SUT), Thailand. She received her B.S. in radiation techniques from Mahidol University, Thailand, in 1985, M.S. in computer science from the Prince of Songkla University, Thailand, in 1991 and Ph.D. in computer science from Nova Southeastern University, U.S.A., in 1999. Her research of interest includes Data Mining, Artificial Intelligence, Logic and Constraint Programming.

K. Kerdprasop is an associate professor at the School of Computer Engineering, Chair of the School, and the head of Knowledge Engineering Research Unit, SUT. He received his bachelor degree in Mathematics from Srinakarinwirot University, Thailand, in 1986, MS in computer science from the Prince of Songkla University, Thailand, in 1991 and Ph.D. in computer science from Nova Southeastern University, U.S.A., in 1999. His current research includes Machine Learning and Artificial Intelligences.

ประวัติผู้เขียน

นางสาวอนุสรา หิรัญวนากุล เกิดเมื่อวันที่ 17 กันยายน พ.ศ. 2528 ที่จังหวัดแพร่ สำเร็จการศึกษาระดับชั้นมัธยมศึกษาตอนต้นปีที่ 3 ที่โรงเรียนภัทรวิทยา อำเภอแม่สอด จังหวัดตาก จากนั้นได้เข้าศึกษาต่อในระดับมัธยมศึกษาตอนปลาย ที่โรงเรียนสรรพวิทยาคม อำเภอแม่สอด จังหวัดตาก จนสำเร็จการศึกษาในปีการศึกษา 2546 จากนั้นได้เข้าศึกษาต่อระดับปริญญาตรีในสาขาวิชาวิศวกรรมคอมพิวเตอร์ สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี และสำเร็จการศึกษาเมื่อปี พ.ศ. 2549 ภายหลังจากสำเร็จการศึกษาได้เข้าทำงานที่บริษัท VIZRT ในตำแหน่ง Broadcast System Engineer เมื่อปี พ.ศ. 2550 - 2551 จากนั้นได้เข้าทำงานที่บริษัท Hitachi Global Storage Technologies (Thailand) Ltd.ในตำแหน่ง Data Mining Expert Engineer เมื่อปี พ.ศ. 2551 - 2554 หลังจากนั้นได้เข้าทำงานที่บริษัท Seagate Technology (Thailand) Ltd. - Korat Plant ในตำแหน่ง Process Engineer เมื่อปี พ.ศ. 2554 - 2556

ในปีการศึกษา 2556 ได้เข้ารับการศึกษาต่อในระดับปริญญาโท สาขาวิชาวิศวกรรมคอมพิวเตอร์ สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี และสำเร็จการศึกษาในปี พ.ศ. 2558 หลังจากนั้นได้รับการบรรจุเข้าเป็นอาจารย์ ประจำคณะวิทยาการสารสนเทศ สาขาวิศวกรรมซอฟต์แวร์ มหาวิทยาลัยบูรพา จังหวัดชลบุรี จากนั้นจึงเข้ารับการศึกษาต่อในระดับปริญญาเอก สาขาวิชาวิศวกรรมคอมพิวเตอร์ สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารีจนถึงปัจจุบัน

ประสบการณ์ที่ได้รับจากการทำงานทั้งในส่วนของการเป็นอาจารย์ในระดับมหาวิทยาลัยและการทำงานในภาคส่วนของบริษัทเอกชนนั้น มีส่วนสำคัญในการช่วยให้ผู้วิจัยสามารถนำมาใช้ต่อยอดสำหรับการศึกษาค้นคว้าเพิ่มเติม อีกทั้งในระหว่างการศึกษาในระดับปริญญาเอกยังได้รับความไว้วางใจให้เป็นผู้ช่วยสอนปฏิบัติการ จากอาจารย์ประจำวิชา Computer Programming, Operating Systems และ Database Systems ซึ่งเป็นผลให้สามารถสร้างสรรค์งานวิจัยและได้รับการตีพิมพ์เผยแพร่บทความวิชาการ ซึ่งรายละเอียดสามารถดูได้ที่ภาคผนวก ข