การทำนายร่องรอยที่ไม่ดี ของโมเดลการออกแบบซอฟต์แวร์โดยใช้วิธีการเรียนรู้ของเครื่อง

นายนครินทร์ มณีรัตน์

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2553
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

BAD-SMELL PREDICTION FROM SOFTWARE DESIGN MODEL USING MACHINE LEARNING TECHNIQUES

Mr. Nakarin Maneerat

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science Program in Computer Science
Department of Computer Engineering
Faculty of Engineering
Chulalongkorn University
Academic Year 2010

Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์	การทำนายร่องรอยที่ไม่ดี ของโมเดลการออกแบบซอฟต์แวร์โดยใช้วิธีการเรียนรู้ของเครื่อง
โดย	นายนครินทร์ มณีรัตน์
สาขาวิชา	วิทยาศาสตร์คอมพิวเตอร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	รองศาสตราจารย์ ดร.พรศิริ หมื่นไชยศรี

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์
(รองศาสตราจารย์ ดร.บุญสม เลิศหิรัญวงศ์)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(รองศาสตราจารย์ ดร.ทวิตีย์ เสนีวงศ์ ณ อยุธยา)

..... อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(รองศาสตราจารย์ ดร.พรศิริ หมื่นไชยศรี)

..... กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.สุกรี สินธุภิญโญ)

..... กรรมการภายนอกมหาวิทยาลัย
(ผู้ช่วยศาสตราจารย์ ดร.ทรงศักดิ์ รองวิริยะพานิช)

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

นครินทร์ มณีรัตน์ : การทำนายร่องรอยที่ไม่ดี ของโมเดลการออกแบบซอฟต์แวร์โดยใช้วิธีการเรียนรู้ของเครื่อง. (BAD-SMELL PREDICTION FROM SOFTWARE DESIGN MODEL USING MACHINE LEARNING TECHNIQUES) อ. ที่ปรึกษาวิทยานิพนธ์หลัก: รศ.ดร.พรศิริ หมื่นไชยศรี, 121 หน้า.

การทำนายร่องรอยที่ไม่ดีในกระบวนการผลิตซอฟต์แวร์ เป็นหนึ่งวิธีการตรวจคุณภาพของซอฟต์แวร์ ในหลาย ๆ งานวิจัยก่อนหน้านี้ได้เสนอวิธีการทำนายร่องรอยที่ไม่ดีจากซอร์สโค้ดของโปรแกรม การทำนายร่องรอยที่ไม่ดีจะมีประโยชน์อย่างมาก ถ้าเราสามารถทำการทำนายร่องรอยที่ไม่ดีได้อย่างรวดเร็วที่สุดเท่าที่ทำได้ในกระบวนการผลิตซอฟต์แวร์ เป้าหมายของงานวิจัยนี้ คือ การนำเสนอวิธีการในการทำนายร่องรอยที่ไม่ดีของโมเดลการออกแบบซอฟต์แวร์โดยใช้วิธีการเรียนรู้ของเครื่อง โดยเราได้ทำการเก็บรวบรวมชุดข้อมูล 7 ชุดจากงานวิจัยที่ผ่านมา ชุดข้อมูลทั้ง 7 ชุดนั้น ประกอบด้วย 27 มาตรวัดการออกแบบ และ ร่องรอยที่ไม่ดีทั้ง 7 ชนิด ชุดข้อมูลทั้ง 7 ชุดนั้นจะถูกใช้ในการเรียนรู้ และ ทำนายร่องรอยที่ไม่ดีโดยวิธีการเรียนรู้ของเครื่อง งานวิจัยนี้ใช้วิธีการเลือกสุ่มข้อมูลแบบความเที่ยงตรงโดยการแบ่งชุดข้อมูลเป็น 10 ส่วนเพื่อป้องกันการเข้ากันของชุดข้อมูลมากเกินไป (Over-fitting) งานวิจัยนี้ทำการประเมินผลโดยการคำนวณประสิทธิภาพการทำนายร่องรอยที่ไม่ดีด้วย ค่าความถูกต้องการทำนายร่องรอยที่ไม่ดี ความไวและความจำเพาะ และ ค่าพยากรณ์ของการทดสอบแล้วเปรียบเทียบความสามารถในการทำนายของวิธีการเรียนรู้ของเครื่องแต่ละวิธี จากงานวิจัยนี้สามารถสรุปได้ว่าการทำนายร่องรอยที่ไม่ดีจากโมเดลการออกแบบโดยใช้วิธีการเรียนรู้ของเครื่องมีค่าใกล้เคียงกับจำนวนร่องรอยที่ไม่ดีจริงของคลาส

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ภาควิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อนิสิต นครินทร์ มณีรัตน์

สาขาวิชา วิทยาศาสตร์คอมพิวเตอร์ ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์หลัก

ปีการศึกษา 2553

5171417221 : MAJOR COMPUTER SCIENCE
KEYWORDS : BAD-SMELL / SOFTWARE DESIGN MODEL / RANDOM FOREST / UML / DESIGN DIAGRAM METRICS / PREDICTION MODELS / MACHINE LEARNERS

NAKARIN MANEERAT : BAD-SMELL PREDICTION FROM SOFTWARE DESIGN MODEL USING MACHINE LEARNING TECHNIQUES. ADVISOR : ASSOC. PROF. PORNSIRI MUENCHAISRI, Ph.D., 121 pp.

Bad-smell prediction in software development significantly impacts on the quality of the resulting software. Code metrics have been successfully used for predicting bad-smells in many literatures. It is more beneficial if bad-smell prediction can be performed as early as possible in the development life cycle. The goal of this paper is to present a methodology for predicting bad-smells from software design model and metrics using machine learning techniques. We collect 7 data sets from the previous literatures which offer 27 design model metrics and 7 bad-smells. The data sets are used as a learning set and a testing set to predict bad-smells using seven machine learning algorithms. We use 10-fold cross-validation for assessing the performance of a predictive model and for preventing over-fitting of the model. Prediction accuracy, hypothesis test, sensitivity and specificity, and predictive value of tests are used to evaluate and compare the performance of the machine learning algorithms. With the experiments, we can conclude that a bad-smell prediction from software design model using machine learning techniques have proximity to actual values

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

Department : Computer Engineering Student's Signature นครินทร์ มณีวัฒน์
Field of Study : Computer Science Advisor's Signature Pornsiri Muenchaisri
Academic Year : 2010

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยความช่วยเหลือจาก รศ.ดร.พรศิริ หมื่นไชยศรี อาจารย์ที่ปรึกษาวิทยานิพนธ์ของข้าพเจ้า ขอกราบขอบพระคุณอาจารย์ที่ให้คำแนะนำ และข้อเสนอแนะต่างๆ ตลอดระยะเวลาของการทำวิทยานิพนธ์ของข้าพเจ้า จนสำเร็จลุล่วงได้ด้วยดี

ขอกราบขอบพระคุณรศ.ดร.ทวิตีย์ เสนีวงศ์ ณ อยุธยา ที่เป็นประธานกรรมการ ผศ.ดร.สุกรี สินธุภิญโญ และผศ.ดร.ทรงศักดิ์ รองวิริยะพานิช เป็นกรรมการสอบวิทยานิพนธ์ ซึ่งได้สละเวลาและให้คำแนะนำต่างๆ ในการสอบวิทยานิพนธ์ของข้าพเจ้า

ขอกราบขอบพระคุณบิดา มารดา พี่ และขอบคุณเพื่อนๆ ทุกคนที่เป็นกำลังใจ และให้ความสนับสนุนมาโดยตลอด

สุดท้ายนี้ หากมีสิ่งใดขาดตกบกพร่องหรือข้อผิดพลาดประการใด ผู้วิจัยขออภัยเป็นอย่างสูงในข้อบกพร่องและความผิดพลาดนั้น และหวังเป็นอย่างยิ่งว่าวิทยานิพนธ์ฉบับนี้จะเป็นประโยชน์บ้างไม่มากก็น้อยสำหรับผู้ที่สนใจจะศึกษารายละเอียดต่อไป

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ง

บทคัดย่อภาษาอังกฤษ ..... จ

กิตติกรรมประกาศ ..... ฉ

สารบัญ ..... ช

สารบัญตาราง ..... ญ

สารบัญภาพ ..... ฐ

บทที่ 1 บทนำ 1

1.1 ความเป็นมาและความสำคัญของปัญหา 1

1.2 วัตถุประสงค์ของการวิจัย 3

1.3 ขอบเขตของการวิจัย 4

1.4 ลำดับขั้นตอนในการเสนอผลการวิจัย 4

1.5 ประโยชน์ที่คาดว่าจะได้รับ 5

บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง 6

2.1. ทฤษฎีที่เกี่ยวข้อง 6

2.1.1 ร่องรอยที่ไม่ดี 6

2.1.2 มาตวัดการออกแบบ 9

2.1.3 การเรียนรู้ของเครื่อง 13

2.1.4 การเลือกสุ่มข้อมูลแบบความเที่ยงตรง 15

2.1.5 การคำนวณประสิทธิภาพการทำนายร่องรอยที่ไม่ดี 16

2.1.6 การทดสอบสมมติฐานของงานวิจัย 17

2.2. งานวิจัยที่เกี่ยวข้อง 20

บทที่ 3 การออกแบบขั้นตอนการดำเนินงาน 28

บทที่ 4 วิธีการทดลอง ผลการทดลอง และ การประเมินผล 33

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

หน้า

4.1 กลุ่มตัวอย่างที่ใช้ในการทดลอง 33
4.2 เครื่องมือที่ใช้ในการวิจัย 35
4.3 ผลการทดลอง 36
4.3.1. Lazy Class 37
4.3.2. Feature Envy 44
4.3.3. Middle Man 51
4.3.4. Message Chains 58
4.3.5. Long Method 65
4.3.6. Long Parameter Lists 72
4.3.7. Switch Statement 79
4.4 การคำนวณประสิทธิภาพการทำนายร่องรอยที่ไม่ดี 86
4.5 การเปรียบเทียบประสิทธิภาพของวิธีการเรียนรู้เครื่อง 95
4.6 การประเมินผล 98
4.6.1. การเลือกวิธีการเรียนรู้เครื่องที่เหมาะสมกับร่องรอยที่ไม่ดีแบบทั่วไป 99
1) ความถูกต้องในทำนายร่องรอยที่ไม่ดี (Bad-smell prediction accuracy) 99
2) ความไว แล ะความจำเพาะ (Sensitivity & Specificity) 100
3) ค่าพยากรณ์ของการทดสอบ (Predictive value of tests) 101
4.6.2. การเลือกวิธีการเรียนรู้เครื่องที่เหมาะสมตามความร้ายแรงของร่องรอยที่ไม่ดี 102
1) การเลือกวิธีการเรียนรู้เครื่องที่เหมาะสมกับร่องรอยที่ไม่ดีที่มีความร้ายแรง 102
2) การเลือกวิธีการเรียนรู้เครื่องที่เหมาะสมกับร่องรอยที่ไม่ดีที่มีความร้ายแรงน้อย 103
4.6.3. การเลือกวิธีการเรียนรู้เครื่องที่เหมาะสมกับร่องรอยที่ไม่ดีทุกประเภท 104
บทที่ 5 สรุปผลการวิจัย และ ข้อเสนอแนะ 105
5.1. บทสรุป 105
5.2. ปัญหา และ ข้อจำกัด 106

หน้า

5.3. ข้อเสนอแนะ 106
รายการอ้างอิง 107
ภาคผนวก 109
ภาคผนวก ก การแปลงรหัสต้นฉบับเป็นแผนภาพคลาส การคำนวณค่ามาตรวัด 110
ภาคผนวก ข การจำแนกประเภทข้อมูลเพื่อใช้ในการทำนายร่องรอยที่ไม่ดี 117
ภาคผนวก ค ผลงานตีพิมพ์ 120
ประวัติผู้เขียนวิทยานิพนธ์ 121

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญตาราง

หน้า

ตารางที่ 2.1 ตารางมาตรวัดการออกแบบ 9
ตารางที่ 2.2 แสดงค่าที่เป็นไปได้ของการทำนาย 18

ตารางที่ 3. 1 ค่ามาตรวัด 27 มาตรวัดที่ได้จากการสกัดโมเดลการออกแบบ 31

ตารางที่ 4.1 ตัวอย่างตารางของผลการทดลอง..... 36
ตารางที่ 4.2 ผลการทำนายร่องรอยไม่ดีประเภท Lazy Class โดย Naive Bayes..... 37
ตารางที่ 4.3 ผลการทำนายร่องรอยไม่ดีประเภท Lazy Class โดย Logistic 38
ตารางที่ 4.4 ผลการทำนายร่องรอยไม่ดีประเภท Lazy Class โดย IB1..... 39
ตารางที่ 4.5 ผลการทำนายร่องรอยไม่ดีประเภท Lazy Class โดย IBk..... 40
ตารางที่ 4.6 ผลการทำนายร่องรอยไม่ดีประเภท Lazy Class โดย VFI..... 41
ตารางที่ 4.7 ผลการทำนายร่องรอยไม่ดีประเภท Lazy Class โดย J48 42
ตารางที่ 4.8 ผลการทำนายร่องรอยไม่ดีประเภท Lazy Class โดย Random forest 43
ตารางที่ 4.9 ผลการทำนายร่องรอยไม่ดีประเภท Feature Envy โดย Naive Bayes 44
ตารางที่ 4.10 ผลการทำนายร่องรอยไม่ดีประเภท Feature Envy โดย Logistic..... 45
ตารางที่ 4.11 ผลการทำนายร่องรอยไม่ดีประเภท Feature Envy โดย IB1 46
ตารางที่ 4.12 ผลการทำนายร่องรอยไม่ดีประเภท Feature Envy โดย IBk 47
ตารางที่ 4.13 ผลการทำนายร่องรอยไม่ดีประเภท Feature Envy โดย VFI 48
ตารางที่ 4.14 ผลการทำนายร่องรอยไม่ดีประเภท Feature Envy โดย J48..... 49
ตารางที่ 4.15 ผลการทำนายร่องรอยไม่ดีประเภท Feature Envy โดย Random forest..... 50
ตารางที่ 4.16 ผลการทำนายร่องรอยไม่ดีประเภท Middle Man โดย Naive Bayes 51
ตารางที่ 4.17 ผลการทำนายร่องรอยไม่ดีประเภท Middle Man โดย Logistic..... 52
ตารางที่ 4.18 ผลการทำนายร่องรอยไม่ดีประเภท Middle Man โดย IB1 53
ตารางที่ 4.19 ผลการทำนายร่องรอยไม่ดีประเภท Middle Man โดย IBk 54
ตารางที่ 4.20 ผลการทำนายร่องรอยไม่ดีประเภท Middle Man โดย VFI 55
ตารางที่ 4.21 ผลการทำนายร่องรอยไม่ดีประเภท Middle Man โดย J48..... 56
ตารางที่ 4.22 ผลการทำนายร่องรอยไม่ดีประเภท Middle Man โดย Random forest..... 57
ตารางที่ 4.23 ผลการทำนายร่องรอยไม่ดีประเภท Message Chains โดย Naive Bayes 58

หน้า

ตารางที่ 4.24 ผลการทำนายร่องรอยไม่ดีประเภท Message Chains โดย Logistic 59
ตารางที่ 4.25 ผลการทำนายร่องรอยไม่ดีประเภท Message Chains โดย IB1 60
ตารางที่ 4.26 ผลการทำนายร่องรอยไม่ดีประเภท Message Chains โดย IBk 61
ตารางที่ 4.27 ผลการทำนายร่องรอยไม่ดีประเภท Message Chains โดย VFI 62
ตารางที่ 4.28 ผลการทำนายร่องรอยไม่ดีประเภท Message Chains โดย J48..... 63
ตารางที่ 4.29 ผลการทำนายร่องรอยไม่ดีประเภท Message Chains โดย Random forest..... 64
ตารางที่ 4.30 ผลการทำนายร่องรอยไม่ดีประเภท Long Method โดย Naive Bayes..... 65
ตารางที่ 4.31 ผลการทำนายร่องรอยไม่ดีประเภท Long Method โดย Logistic 66
ตารางที่ 4.32 ผลการทำนายร่องรอยไม่ดีประเภท Long Method โดย IB1..... 67
ตารางที่ 4.33 ผลการทำนายร่องรอยไม่ดีประเภท Long Method โดย IBk..... 68
ตารางที่ 4.34 ผลการทำนายร่องรอยไม่ดีประเภท Long Method โดย VFI..... 69
ตารางที่ 4.35 ผลการทำนายร่องรอยไม่ดีประเภท Long Method โดย J48 70
ตารางที่ 4.36 ผลการทำนายร่องรอยไม่ดีประเภท Long Method โดย Random forest 71
ตารางที่ 4.37 ผลการทำนายร่องรอยไม่ดีประเภท Long Parameter Lists โดย Naive Bayes..... 72
ตารางที่ 4.38 ผลการทำนายร่องรอยไม่ดีประเภท Long Parameter Lists โดย Logistic 73
ตารางที่ 4.39 ผลการทำนายร่องรอยไม่ดีประเภท Long Parameter Lists โดย IB1..... 74
ตารางที่ 4.40 ผลการทำนายร่องรอยไม่ดีประเภท Long Parameter Lists โดย IBk..... 75
ตารางที่ 4.41 ผลการทำนายร่องรอยไม่ดีประเภท Long Parameter Lists โดย VFI..... 76
ตารางที่ 4.42 ผลการทำนายร่องรอยไม่ดีประเภท Long Parameter Lists โดย J48 77
ตารางที่ 4.43 ผลการทำนายร่องรอยไม่ดีประเภท Long Parameter Lists โดย Random forest ... 78
ตารางที่ 4.44 ผลการทำนายร่องรอยไม่ดีประเภท Switch Statement โดย Naive Bayes 79
ตารางที่ 4.45 ผลการทำนายร่องรอยไม่ดีประเภท Switch Statement โดย Logistic..... 80
ตารางที่ 4.46 ผลการทำนายร่องรอยไม่ดีประเภท Switch Statement โดย IB1 81
ตารางที่ 4.47 ผลการทำนายร่องรอยไม่ดีประเภท Switch Statement โดย IBk 82
ตารางที่ 4.48 ผลการทำนายร่องรอยไม่ดีประเภท Switch Statement โดย VFI 83
ตารางที่ 4.49 ผลการทำนายร่องรอยไม่ดีประเภท Switch Statement โดย J48..... 84
ตารางที่ 4.50 ผลการทำนายร่องรอยไม่ดีประเภท Switch Statement โดย Random forest..... 85
ตารางที่ 4.51 แสดงตัวอย่างวิธีการวิเคราะห์ผลลัพธ์ โดยแบ่งผลลัพธ์เป็นสองพวก จากผลการทำนายร่องรอยไม่ดีประเภท Lazy Class โดย Naive Bayes..... 87

หน้า

ตารางที่ 4.52 แสดงตัวอย่างวิธีการวิเคราะห์ผลลัพธ์ โดยแบ่งผลลัพธ์เป็นสี่พวก จากผลการทำนายร่องรอยไม่ดีประเภท Lazy Class โดย Naive Bayes 88
ตารางที่ 4.53 ความถูกต้องในทำนายร่องรอยที่ไม่ดี (Bad-smell prediction accuracy) 89
ตารางที่ 4.54 ความไวในการทำนายในการทำนายร่องรอยที่ไม่ดี 91
ตารางที่ 4.55 ความจำเพาะในการทำนายร่องรอยที่ไม่ดี 92
ตารางที่ 4.56 ค่าพยากรณ์ของการทดสอบทางบวกในการทำนายร่องรอยที่ไม่ดี 93
ตารางที่ 4.57 ค่าพยากรณ์ของการทดสอบทางลบในการทำนายร่องรอยที่ไม่ดี 94
ตารางที่ 4.58 ความถูกต้องในทำนายร่องรอยที่ไม่ดีของวิธีการเรียนรู้เครื่องประเภท Naive Bayes และ VFI 95
ตารางที่ 4.59 ความถูกต้องในทำนายร่องรอยที่ไม่ดีของวิธีการเรียนรู้เครื่องประเภท IB1 และ Random Forest 96
ตารางที่ 4.60 ค่า t ที่ได้จากการเปรียบเทียบวิธีการเรียนรู้เครื่องทุกวิธี 97
ตารางที่ 4.61 ตารางการยอมรับสมมติฐานหลัก หรือ ยอมรับสมมติฐานรอง ตามวิธีการเรียนรู้เครื่อง 98
ตารางที่ 4.62 วิธีการเรียนรู้เครื่องที่เหมาะสมในการทำนายร่องรอยที่ไม่ดีตามวิธีการคำนวณประสิทธิภาพการทำนายร่องรอยที่ไม่ดี 99
ตารางที่ 4.63 ความผิดพลาดแบบ I (Type I Error) หรือ False negative (FN) ของวิธีการเรียนรู้เครื่องที่ตามประเภทของร่องรอยที่ไม่ดี 103
ตารางที่ 4.63 ความผิดพลาดแบบที่ II (Type II Error) หรือ False positive (FP) ของวิธีการเรียนรู้เครื่องที่ตามประเภทของร่องรอยที่ไม่ดี 104

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญภาพ

หน้า

รูปที่ 1.1 กระบวนการผลิตซอฟต์แวร์โดยทั่วไปที่สามารถตรวจหาร่องรอยที่ไม่ดีจากซอร์สโค้ด 2
รูปที่ 1.2 กระบวนการผลิตซอฟต์แวร์ที่สามารถทำนายร่องรอยที่ไม่ดีของโมเดลการออกแบบ 3

รูปที่ 3. 1 ขั้นตอนของงานวิจัยทั้งหมด 28
รูปที่ 3. 2 ตัวอย่างรหัสต้นฉบับภาษาจาวาที่มีร่องรอยไม่ดีประเภท Message Chains..... 30
รูปที่ 3. 3 ตัวอย่างรหัสต้นฉบับภาษาจาวาถูกทำวิศวกรรมย้อนกลับ 31

รูปที่ ก-1หน้าจอหลักของโปรแกรม MagicDraw UML..... 110
รูปที่ ก-2 หน้าต่างการสร้าง Code engineering sets..... 111
รูปที่ ก-3 หน้าต่าง Code engineering sets ที่สร้างขึ้น..... 111
รูปที่ ก-4 หน้าต่างการเลือกไฟล์รหัสต้นฉบับนำเข้า..... 112
รูปที่ ก-5 หน้าต่างรายละเอียดของไฟล์รหัสต้นฉบับที่เลือก..... 112
รูปที่ ก-6 หน้าต่างการเลือกคำสั่ง Reverse เพื่อแปลงกลับแผนภาพ..... 113
รูปที่ ก-7 หน้าต่างการกำหนดค่าการแปลงกลับแผนภาพ..... 113
รูปที่ ก-8 หน้าต่างแผนภาพคลาสที่ได้จากการแปลงกลับ..... 114
รูปที่ ก-9 หน้าต่างการเลือกคำสั่ง Metrics เพื่อการคำนวณค่ามาตรวัด..... 115
รูปที่ ก-10 หน้าต่างการกำหนดค่าการคำนวณมาตรวัด..... 115
รูปที่ ก-11 ผลลัพธ์ที่ได้จากการคำนวณค่ามาตรวัด..... 115
รูปที่ ก-12 หน้าต่างตำแหน่งที่จะบันทึกข้อมูล และ ประเภทไฟล์ที่ต้องการบันทึก..... 116
รูปที่ ก-13 ข้อมูลค่ามาตรวัดที่ได้จากการบันทึก 116
รูปที่ ก-14 ตารางค่ามาตรวัด 116

รูปที่ ข-1หน้าจอการแสดงผล 117
รูปที่ ข-2 หน้าจอหลักของโปรแกรม 117
รูปที่ ข-3 หน้าต่างเลือกไฟล์ 118
รูปที่ ข-4 หน้าต่างเพื่อเลือก Classifier ที่ต้องการ 118
รูปที่ ข-5 วิธีการเลือกสุ่มข้อมูลแบบความเที่ยงตรง..... 119
รูปที่ ข-6 ระบบจะแสดงผลลัพธ์ที่ได้จากการทำนาย 119

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

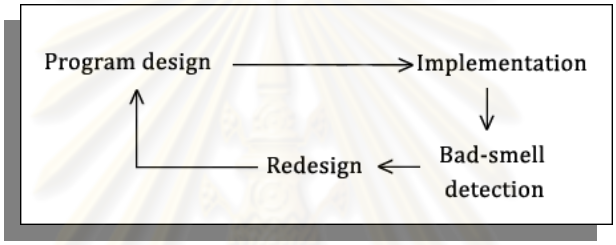
กระบวนการผลิตซอฟต์แวร์ในปัจจุบัน ประกอบด้วย การกำหนดความต้องการ การออกแบบ การเขียนซอร์สโค้ด การตรวจสอบ และ การบำรุงรักษา ถ้าสามารถตรวจสอบคุณภาพของซอฟต์แวร์ตั้งแต่การออกแบบในกระบวนการผลิตซอฟต์แวร์ จะช่วยลดเวลาในกระบวนการผลิตซอฟต์แวร์ ปัญหาหนึ่งที่มีผลต่อคุณภาพของซอฟต์แวร์ คือ การเกิดร่องรอยที่ไม่ดี (Bad-smell) ที่เกิดขึ้นจากขั้นตอนการออกแบบหรือการเขียนโปรแกรมไม่สมบูรณ์ วงจรในการผลิตซอฟต์แวร์จะสั้นลง และ เพิ่มประสิทธิภาพของกระบวนการพัฒนาซอฟต์แวร์ [1] ได้ถ้ามีวิธีการในการทำนายร่องรอยที่ไม่ดีตั้งแต่การออกแบบ

Fowler and Beck [2] ได้กำหนดร่องรอยที่ไม่ดี 22 ชนิดและได้นำเสนอวิธีการแก้ไขร่องรอยที่ไม่ดี ในหลายปีที่ผ่านมามีงานวิจัยจำนวนมากเสนอวิธีการตรวจจับร่องรอยที่ไม่ดีอย่างอัตโนมัติจากซอร์สโค้ด อาทิ Matthew James Munro [3], Radu Marinescu[4] และ Thisana Pienlert and Pornsiri Muenchaisri [5] ได้เสนอวิธีการตรวจจับร่องรอยที่ไม่ดีในซอร์สโค้ดโดยใช้มาตรวัดเช่นเดียวกัน แต่จะแตกต่างกันที่ชนิดของร่องรอยที่ไม่ดีที่ใช้ในการตรวจสอบ

Khanti Yeesoon [6] นำเสนอวิธีการค้นหาร่องรอยที่ไม่ดีสำหรับซอร์สโค้ดภาษาจาวาจากการพิจารณาองค์ประกอบของโปรแกรม และ ใช้วิธีการของโปรล็อกในการค้นหาโดยเลือกวิธีการแสดงความรู้ตรรกะเพรดิเคต ออกแบบกฎโปรล็อกจากร่องรอยที่ไม่ดี และข้อเท็จจริงจากซอร์สโค้ดต้นฉบับ ประเมินความสามารถจากการทดสอบกับโปรแกรมต้นฉบับภาษาจาวา 3 โปรแกรม เปรียบเทียบค่าที่ได้จากการใช้เครื่องมือกับการค้นหาด้วยบุคคล พร้อมทั้งพัฒนาเครื่องมือตามวิธีการที่ได้นำเสนอ ผลที่ได้พบว่า วิธีการนี้สามารถค้นหาร่องรอยที่ไม่ดีได้เป็นอย่างดี Sakorn Mekruksavanich [7] นำเสนอวิธีการค้นหาข้อบกพร่องในการออกแบบ (Design flaws) ในโปรแกรมเชิงวัตถุ ข้อบกพร่องในการออกแบบสามารถค้นหาได้ที่ระดับนิยาม (Meta-level) ของโปรแกรมในการเขียนโปรแกรมเชิงประกาศ (Declarative programming) เขาเลือกภาษาโปรล็อกในการเขียนโปรแกรมเชิงประกาศ โดยตรวจสอบความสามารถของการค้นหาข้อบกพร่องจากการ

ออกแบบประเภทความสัมพันธ์ (The couplers) เช่น Message Chains, Inappropriate Intimacy และ Middle Man

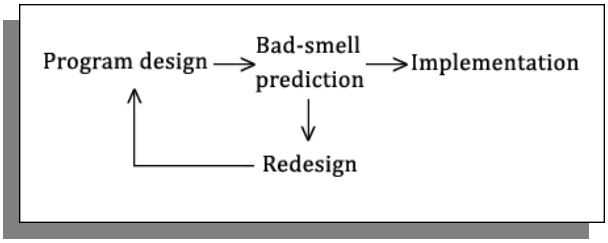
รูปที่ 1.1 แสดงกระบวนการผลิตซอฟต์แวร์โดยทั่วไปที่สามารถตรวจหาร่องรอยที่ไม่ดีจากซอร์สโค้ด คือต้องออกแบบโปรแกรมและนำไปเขียนซอร์สโค้ด จากนั้นใช้มาตรวัดในการตรวจจับว่าซอร์สโค้ดจะมีร่องรอยที่ไม่ดีหรือไม่ ถ้าซอร์สโค้ดมีร่องรอยที่ไม่ดีวงจรการพัฒนาจะกลับไปยังเฟสการออกแบบ



รูปที่ 1.1 กระบวนการผลิตซอฟต์แวร์โดยทั่วไปที่สามารถตรวจหาร่องรอยที่ไม่ดีจากซอร์สโค้ด

เนื่องจากงานวิจัยดังกล่าวข้างต้นนั้น[3] [4] [5] [6] [7] สามารถตรวจจับร่องรอยที่ไม่ดีได้จากซอร์สโค้ดของโปรแกรม ซึ่งจะต้องทำการเขียนซอร์สโค้ดให้เสร็จก่อนที่จะนำวิธีการตรวจจับข้างต้นมาใช้ได้ ส่งผลให้สามารถตรวจสอบคุณภาพซอฟต์แวร์ล่าช้าในกระบวนการผลิตซอฟต์แวร์ ผู้วิจัยจึงใช้แนวคิดของงานวิจัยที่ทำนายคุณสมบัติที่ไม่ดีของโมเดลการออกแบบซอฟต์แวร์ มาทำนายร่องรอยที่ไม่ดี รูปที่ 1.2 แสดงวิธีการทำนายร่องรอยที่ไม่ดีที่สามารถทำได้ก่อนนำไปเขียนซอร์สโค้ดในกระบวนการผลิตซอฟต์แวร์ นั่นคือ การทำนายร่องรอยที่ไม่ดีของโมเดลการออกแบบ

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 1.2 กระบวนการผลิตซอฟต์แวร์ที่สามารถทำนายร่องรอยที่ไม่ดีของโมเดลการออกแบบ

Sallie Henry and Calvin Selig [1] ทำนายคุณภาพของซอร์สโค้ดตั้งแต่เริ่มการออกแบบซอฟต์แวร์ โดยใช้มาตรวัดซอร์สโค้ดที่ใช้คือ Lines of code, Software science, Cyclomatic complexity และมาตรวัดของการออกแบบที่ใช้ คือ Fan-in และ Fan-out โดยวิธีการวิเคราะห์รีเกรสชัน (Regression Analysis) ในการทำนายความซับซ้อนของซอร์สโค้ด และประเมินความสามารถในการทำนายความซับซ้อนกับโปรแกรมตรวจสอบ

งานวิจัยฉบับนี้ได้นำเสนอวิธีการในการทำนายร่องรอยที่ไม่ดีของโมเดลการออกแบบซอฟต์แวร์โดยใช้วิธีการเรียนรู้ของเครื่อง โดยใช้ข้อมูลงานวิจัยที่ผ่านมา[5][6] โดยข้อมูลประกอบด้วยร่องรอยที่ไม่ดี 7 ประเภท คือ Lazy Class[5], Feature Envy[5], Middle Man[6], Message Chains[6], Long Method[5], Long Parameter Lists[5] และ Switch Statement[5] และ นำซอร์สโค้ดมาหามาตรวัดคลาสไดอะแกรมโดยใช้ Magic Draw UML 9.5 แล้วนำข้อมูลทั้งสองส่วนมาเป็นข้อมูลนำเข้าของโปรแกรมเวกา (Weka) ที่ใช้เป็นเครื่องมือที่ในการเรียนรู้ จำแนกประเภทข้อมูลและ ตรวจสอบความสามารถในการทำนายร่อยรอยที่ไม่ดี แล้วนำผลการทดลองมาประเมินความถูกต้องของการทำนายร่องรอยที่ไม่ดีโดยใช้ ความถูกต้องการทำนายของการทำนายร่องรอยที่ไม่ดี (Bad-smell prediction accuracy) ความไวและความจำเพาะ (Sensitivity and specificity) และ ค่าพยากรณ์ของการทดสอบ (Predictive value of tests) เป็นตัวชี้วัดความถูกต้อง

1.2 วัตถุประสงค์ของการวิจัย

งานวิจัยนี้มีวัตถุประสงค์เพื่อนำเสนอวิธีการในการทำนายร่องรอยไม่ดีของโมเดลการออกแบบซอฟต์แวร์โดยใช้วิธีการเรียนรู้ของเครื่อง

1.3 ขอบเขตของการวิจัย

1. นำงานวิจัยที่ตรวจจับร่องรอยที่ไม่ดีโดยใช้ทฤษฎีร่องรอยที่ไม่ดีของฟาวเลอร์ [2] มาใช้ในการทำนายร่องรอยไม่ดีของโมเดลการออกแบบซอฟต์แวร์ โดยร่องรอยที่ไม่ดีที่นำมาตรวจสอบนั้นจะประกอบด้วย Lazy Class[5], Feature Envy[5], Middle Man[6], Message Chains[6], Long Method[5], Long Parameter Lists[5] และ Switch Statement[5]
2. ใช้มาตรวัดโมเดลการออกแบบซอฟต์แวร์เพื่อตรวจหามาตรวัดการออกแบบ
3. จำนวนซอฟต์แวร์ที่ใช้ในการเรียนรู้จะต้องมีจำนวนไม่ต่ำกว่า 5 ซอฟต์แวร์ โดยในแต่ละซอฟต์แวร์จะต้องมีคลาสอย่างน้อย 5 คลาสที่มีร่องรอยที่ไม่ดีแต่ละประเภทอย่างน้อย 1 คลาสต่อหนึ่งประเภท
4. งานวิจัยนี้ใช้โปรแกรมเวกา (Weka) ในการเรียนรู้ และจำแนกประเภทข้อมูลเพื่อทำนายร่องรอยไม่ดีของโมเดลการออกแบบซอฟต์แวร์

1.4 ลำดับขั้นตอนในการเสนอผลการวิจัย

1. ศึกษาทฤษฎีพื้นฐาน ร่องรอยที่ไม่ดี และ วิธีการตรวจจับร่อยรอยที่ไม่ดีของงานวิจัยที่อ้างอิง
2. ศึกษาวิธีการคำนวณมาตรวัดโมเดลการออกแบบซอฟต์แวร์
3. ศึกษาการใช้เครื่องมือในการเรียนรู้ จำแนกประเภทข้อมูล และ ทดสอบความสามารถในการทำนายร่อยรอยที่ไม่ดี
4. นำซอฟต์แวร์ตัวอย่างมาคำนวณมาตรวัดโมเดลการออกแบบซอฟต์แวร์ และ รวบรวมข้อมูลร่องรอยที่ไม่ดีซอฟต์แวร์ตัวอย่างเพื่อเป็นข้อมูลนำเข้า
5. นำข้อมูลจากข้างต้นมาใช้ในการเรียนรู้ จำแนกประเภทข้อมูลและ ตรวจสอบความสามารถในการทำนายร่อยรอยที่ไม่ดี

6. ประเมินผลความถูกต้องของการทำนายของวิธีการต่าง ๆ

7. วิเคราะห์ผล

8. สรุปผลและเรียบเรียงวิทยานิพนธ์

1.5 ประโยชน์ที่คาดว่าจะได้รับ

สามารถนำวิธีการที่เสนอนี้เป็นทางเลือกในการทำนายร่องรอยไม่ดีของโมเดลการออกแบบซอฟต์แวร์โดยใช้วิธีการเรียนรู้ของเครื่อง ซึ่งสามารถตรวจสอบคุณภาพของซอฟต์แวร์ตั้งแต่การออกแบบในกระบวนการผลิตซอฟต์แวร์ จะช่วยลดเวลาในกระบวนการผลิตซอฟต์แวร์

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1. ทฤษฎีที่เกี่ยวข้อง

งานวิจัยนี้ ผู้วิจัยได้ค้นคว้าศึกษาเอกสาร แหล่งความรู้ทางอินเตอร์เนต งานวิจัย รวมทั้งแนวคิดทฤษฎีต่างๆ ที่เกี่ยวข้อง ได้ดังนี้

2.1.1 ร่องรอยที่ไม่ดี

ร่องรอยที่ไม่ดี (Bad-smell) หมายถึง ลักษณะของการออกแบบซอฟต์แวร์หรือการเขียนโปรแกรมที่ไม่ดี ทำให้การทำความเข้าใจซอฟต์แวร์ และการแก้ไขซอฟต์แวร์ทำได้ยาก [5] ฟาวเลอร์ และ เบค (Fowler and Beck) [2] ได้กำหนดร่องรอยที่ไม่ดี 22 ชนิด ดังนี้

a) The Bloaters

- Long Method

เมทธอดที่มีขนาดใหญ่สามารถอ่านได้ยาก สามารถเข้าใจได้ยาก และสามารถแก้ไขปัญหาได้ยาก ควรถูกแก้ไขให้เมทธอดมีขนาดเล็กลง หรือ แยกเป็นหลายๆ เมทธอด

- Large Class

คลาสที่มีเมทธอด และจำนวนตัวแปรมาก ทำให้คลาสมีหน้าที่จำนวนมากซึ่งส่งผลให้คลาสอ่านได้ยาก สามารถเข้าใจได้ยาก และสามารถแก้ไขปัญหาได้ยาก ควรถูกแก้ไขให้คลาสมีขนาดเล็กลง หรือ แยกเป็นหลายๆ คลาส

- Primitive Obsession

อย่าใช้ชนิดข้อมูลพื้นฐานที่ภาษานั้น ๆ จัดไว้ให้เท่านั้น ถ้าชนิดข้อมูลที่ต้องการใช้มีความแตกต่างไปจากชนิดข้อมูลที่มีมาให้อยู่แล้ว ก็ควรจะเขียนคลาสใหม่ตามความต้องการ

- Long Parameter List

เมื่อมีการส่งผ่านพารามิเตอร์จำนวนมาก ก็จะเกิดความซับซ้อน จึงมีความจำเป็นในการจำกัดจำนวนพารามิเตอร์ที่จะส่งผ่าน หรือ ทำการรวมพารามิเตอร์ที่ต้องการส่งผ่านเป็นออปเจก

- Data Clumps

ถ้ามีกลุ่มของข้อมูลอยู่ในหลายๆ ที่ อาจเป็นไปได้ที่ข้อมูลกลุ่มนั้นขึ้นต่อกัน เพราะฉะนั้นควรจัดกลุ่มของข้อมูลนั้นๆ เป็นคลาสเดียวกัน

b) The Object-Orientation Abusers

- Switch Statements

สวิทช์สเตตเมนต์ (Switch statement) เป็นเหตุของความซ้ำซ้อน ถ้ามีสวิทช์สเตตเมนต์ที่มีความเหมือนๆ กันกระจายอยู่ทั่วไปทั้งโปรแกรม ควรจะเปลี่ยนสวิทช์สเตตเมนต์นั้นๆเป็น หลายซับคลาสที่มีการสืบทอดคุณสมบัติจากคลาสแม่ แล้วเขียนทับเมทธอดที่สืบทอดมา

- Temporary Field

ถ้าพบคลาสที่ประกอบด้วยตัวแปรที่ไม่จำเป็นจำนวนมากซึ่งเก็บค่าชั่วคราวในซอร์สโค้ด ควรกำจัดตัวแปรที่ไม่จำเป็นนั้นๆ ออก

- Refused Bequest

ถ้าคลาสที่ทำการสืบทอดคุณสมบัติของคลาสอื่นมาแล้ว แต่ไม่ได้ใช้คุณสมบัติที่ได้รับการสืบทอดมา ควรจะพิจารณาว่าควรจะต้องทำการสืบทอดหรือไม่

- Alternative Classes with Different Interfaces

ถ้าสองคลาสใดๆ มีความคล้ายคลึงกันภายใน แต่มีความแตกต่างกันภายนอก บางทีเราสามารถแก้ไขปรับปรุงสองคลาสนั้นๆ ให้มาใช้อินเทอเฟส (Interface) ร่วมกันได้

c) The Change Preventers

- Divergent Change

เมื่อมีคลาสที่รองรับการเปลี่ยนแปลงอยู่เรื่อยๆ ควรจะมีการพิจารณาบางส่วนของคลาสแยกออกไปเป็นอีกคลาส

- Shotgun Surgery

ถ้ามีการเปลี่ยนแปลงในคลาสใดๆ แล้วส่งผลให้เกิดการเปลี่ยนแปลงกับคลาสที่มีความสัมพันธ์กันเรื่อยๆ ควรจะมีการพิจารณาให้การเปลี่ยนแปลงไปอยู่ที่คลาสใดคลาสหนึ่ง

- Parallel Inheritance Hierarchies

ทุกครั้งเมื่อมีการสร้างคลาสลูก จะต้องมีการสร้างให้คลาสนั้นเป็นคลาสลูกของสองคลาสอยู่เสมอ เมื่ออีกคลาสแม่ทั้ง 2 คลาสถูกสืบทอด จะทำความเข้าใจคลาสนั้นผิดพลาด เพราะฉนั้นควรพิจารณาให้มีการสืบทอดคุณสมบัติจากคลาสๆ หนึ่งเท่านั้นโดยการย้ายเมทธอด หรือ ตัวแปรมาไว้ที่คลาสแม่เพียงคลาสเดียว

d) The Dispensables

- Lazy Class

ถ้ามีคลาสที่ไม่ได้ทำหน้าที่ของตัวเองเพียงพอควรจะมีการรวมไปอยู่ที่คลาสอื่นๆ

- Data Class

ควรมีการหลีกเลี่ยงคลาสที่มีหน้าที่เพียงแค่เก็บข้อมูลเท่านั้น คลาสควรมีทั้งข้อมูลและหน้าที่

- Duplicated code

ถ้ามีซอร์สโค้ดที่เหมือนๆ กันอยู่มากๆ ซึ่งหมายความว่าได้มีการทำงานที่ซ้ำซ้อนกัน ควรรวมความซ้ำซ้อนเป็นเมทธอด หรือ คลาส

- Incomplete Library Class

เมื่อมีคลาสไลบรารี่ (Library Class) ที่ไม่สมบูรณ์ทำให้การเข้าใจคลาสนั้นทำได้ยากซึ่งส่งผลให้ คลาสนั้นนำกลับมาใช้ใหม่ได้ยากขึ้น

- Speculative Generality

มีการเขียนซอร์สโค้ดที่แก้ปัญหาปัจจุบันโดยที่ไม่ได้คำนึงถึงอนาคต จะทำให้การบำรุงดูแลรักษาทำได้ยาก ควรใช้ดีไซน์แพตเทิร์น (Design pattern) เข้ามาช่วยในการเขียนซอร์สโค้ด

e) The Couplers

- Feature Envy

คลาสที่มีการเรียกใช้คลาสใดคลาสหนึ่ง ทำให้มีการขึ้นต่อคลาสนั้นมาก ควรจะมีการพิจารณาให้มีการย้ายเมทธอดที่อยู่ในคลาสที่ถูกเรียกใช้บ่อยๆ มาอยู่ในคลาสที่เรียกใช้

- Inappropriate Intimacy

คลาสที่มีการเรียกใช้คลาสอื่นๆ หลายๆ ครั้ง หรือ คลาสที่มีการเรียกใช้ข้อมูลของอีกฝ่ายโดยไม่จำเป็นหลายๆ ครั้ง ซึ่งคลาสในแต่ละคลาสควรเรียกใช้กันเพียงเล็กน้อยตามความจำเป็นเท่านั้น

- Middle Man

ถ้ามีคลาสที่เป็นตัวกลางในการทำงาน ระหว่างสอง หรือ หลายคลาส ควรกำจัดคลาสที่เป็นตัวกลางนั้นทิ้งไป

- Message Chains

เมื่อมีการเรียกใช้คลาสอื่นๆเป็นทอดๆ ซึ่งทำให้ต้องเสียเวลาในการเรียกคลาสอื่นๆ โดยไม่จำเป็น ซึ่งส่งผลให้เกิดความผิดพลาดได้ง่าย ควรมีการเขียน หรือ ย้ายเมทธอด เพื่อลดการเรียกใช้คลาสอื่นเป็นทอดๆ

f) Another

- Comments

ถ้ามีการเขียนคอมเมนต์มากเกินไปเกินความจำเป็น ควรมีการเขียนคอมเมนต์ตามเท่าที่จำเป็นเท่านั้น

2.1.2 มาตวัดการออกแบบ

หนึ่งในมาตรวัดการออกแบบ (Design diagram metric) ที่ได้รับความนิยมคือคลาสไดอะแกรม ซึ่งคลาสไดอะแกรมแสดงให้เห็นโครงสร้างของคลาสของโปรแกรม รวมทั้งตัวแปรและคำสั่งของแต่ละคลาส อีกทั้งยังแสดงความเกี่ยวข้องต่างๆระหว่างคลาส เช่นความสัมพันธ์ของการสืบทอด หรือความสัมพันธ์กันระหว่างคลาส

ในงานวิจัยนี้ได้ใช้มาตรวัดการออกแบบทั้งหมด 27 มาตรวัด ซึ่งประกอบด้วย 9 กลุ่ม คือ กลุ่มทั่วไป (Basic) กลุ่มการใช้งานของคลาส (Class Employment) กลุ่มความซับซ้อน (Complexity) กลุ่มไดอะแกรม (Diagrams) กลุ่มการสืบทอด (Inheritance) กลุ่มมูด (MOOD)[10] กลุ่มขนาดโมเดล (Model size) กลุ่มอื่น ๆ (Other) และ กลุ่มความสัมพันธ์ (Relationship)

ตารางที่ 2.1 ตารางมาตรวัดการออกแบบ

กลุ่ม : ทั่วไป	
NA	คำอธิบาย : Number of attributes คำอธิบาย : จำนวนแอทริบิวต์ ความหมาย : มาตรวัดนี้นับจำนวนแอทริบิวต์ในคลาส และอินเตอร์เฟส
NC	คำอธิบาย : Number of Classes คำอธิบาย : จำนวนคลาส ความหมาย : มาตรวัดนี้นับจำนวนคลาส และ อินเตอร์เฟสที่อยู่ในระบบนั้น
NM	คำอธิบาย : Number of Members คำอธิบาย : จำนวนสมาชิก ความหมาย : มาตรวัดนี้นับจำนวนแอทริบิวต์ และ โอเปอเรชั่น ที่มีอยู่ในคลาสนั้น

NO	คำอธิบาย : Number of Operation คำอธิบาย : จำนวนการดำเนินการ ความหมาย : มาตรวัดนี้นับจำนวนโอเปอเรชั่น ที่มีอยู่ในคลาสนั้น
NP	คำอธิบาย : Number of Parameters คำอธิบาย : จำนวนพารามิเตอร์ ความหมาย : มาตรวัดนี้นับจำนวนพารามิเตอร์ ที่มีอยู่ในคลาสนั้น
กลุ่ม : การใช้งานของคลาส	
C_PARAM	คำอธิบาย : Number of times class is used as parameter type คำอธิบาย : จำนวนครั้งที่คลาสถูกใช้เป็นพารามิเตอร์ ความหมาย : มาตรวัดนี้นับจำนวนครั้งที่คลาสถูกใช้เป็น พารามิเตอร์
กลุ่ม :ความซับซ้อน	
RFC	คำอธิบาย : Response for a class คำอธิบาย : จำนวนการตอบสนองต่อคลาส ความหมาย : จำนวนเมทธอดที่ถูกเรียกใช้ ไม่ว่าจะเป็นเมทธอด ของคลาสตัวเอง หรือ เมทธอดที่สืบทอดมา
WAC	คำอธิบาย : Weighted attributes per class คำอธิบาย : จำนวนแอทริบิวต์ต่อคลาส ความหมาย : มาตรวัดนี้นับจำนวนแอทริบิวต์ต่อคลาส
WMA	คำอธิบาย : Weighted methods per class คำอธิบาย : จำนวนเมทธอดต่อคลาส ความหมาย : มาตรวัดนี้นับจำนวนเมทธอดต่อคลาส
กลุ่ม :ไดอะแกรม	
D_APPEAR	คำอธิบาย : Appearance in diagrams คำอธิบาย : ลักษณะที่ปรากฏในไดอะแกรม ความหมาย : มาตรวัดนี้นับจำนวนเครื่องหมายของไดอะแกรม ทั้งหมด เช่น เครื่องหมายคลาส หรือ เครื่องหมายอินเตอร์เฟส
กลุ่ม : การสืบทอด	
DIT	คำอธิบาย : Depth of inheritance tree คำอธิบาย : ระดับชั้นของการสืบทอด

	ความหมาย : มาตรวัดนี้นับจำนวนระยะทางมากที่สุดระหว่างโหนดนั้นถึงโหนดที่สืบทอดคุณสมบัติไป
NOC	คำอธิบาย : Number of children คำอธิบาย : จำนวนลูก ความหมาย : มาตรวัดนี้นับจำนวนคลาสลูกทั้งหมด
NAI	คำอธิบาย : Number of inherited attributes คำอธิบาย : จำนวนของคุณลักษณะที่สืบทอด ความหมาย : มาตรวัดนี้นับจำนวนของคุณลักษณะที่สืบทอด
NOI	คำอธิบาย : Number of inherited operations คำอธิบาย : จำนวนการดำเนินงานที่ถูกรับช่วง ความหมาย : มาตรวัดนี้นับจำนวนการดำเนินงานที่ถูกรับช่วง
กลุ่ม : มูด	
AHF	คำอธิบาย : Attribute hiding factor คำอธิบาย : ปัจจัยการซ่อนแอททริบิวต์ ความหมาย : เปอร์เซ็นต์ของคุณลักษณะที่มองไม่เห็นในทุกๆ คลาส
AIF	คำอธิบาย : Attribute inheritance factor คำอธิบาย : ปัจจัยการสืบทอดคุณสมบัติ ความหมาย : ปัจจัยการสืบทอดคุณลักษณะนั้น เป็นการหารกันระหว่างผลรวมของคุณลักษณะในคลาส หารด้วยผลรวมของคุณลักษณะของคลาสทั้งหมดในระบบ
CF	คำอธิบาย : Coupling factor คำอธิบาย : ปัจจัยความสัมพันธ์ระหว่างคลาส ความหมาย : จำนวนความสัมพันธ์ระหว่างคลาส หารด้วยความสัมพันธ์ระหว่างคลาสของทุกคลาสในระบบ
MHF	คำอธิบาย : Method hiding factor คำอธิบาย : ปัจจัยการซ่อนเมทธอด ความหมาย : เปอร์เซ็นต์ของเมทธอดที่มองไม่เห็นทั้งหมดที่อยู่ในคลาส
MIF	คำอธิบาย : Method inheritance factor

	คำอธิบาย : ปัจจัยการสืบทอดเมทธอด ความหมาย : ปัจจัยการสืบทอดเมทธอดนั้น เป็นการหารกันระหว่างผลรวมของเมทธอดในคลาสหนึ่งๆ ด้วย ผลรวมของเมทธอดของคลาสทั้งหมด
PF	คำอธิบาย : Polymorphism factor คำอธิบาย : ปัจจัยความหลากหลาย ความหมาย : ปัจจัยการสืบทอดคุณสมบัติทั้งหมดนั้น เป็นการหารกันระหว่างผลรวมของจำนวนเมทธอดที่ถูกเขียนทับในคลาสทั้งหมดที่สืบทอดไปทั้งหมดจริงๆ หารด้วยจำนวนเมทธอดที่ถูกเขียนทับสูงสุด
กลุ่ม : ขนาดโมเดล	
ACT	คำอธิบาย : Number of actors คำอธิบาย : จำนวนของผู้กระทำ ความหมาย : นับจำนวนของผู้กระทำ
COMP	คำอธิบาย : Number of components คำอธิบาย : จำนวนคอมโพเนนต์ ความหมาย : นับจำนวนคอมโพเนนต์ทั้งหมด
NS	คำอธิบาย : Number of namespaces คำอธิบาย : จำนวนเนมสเปส ความหมาย : นับจำนวนเนมสเปสทั้งหมด
กลุ่ม : อื่น ๆ	
CBC	คำอธิบาย : Coupling between objects คำอธิบาย : ความสัมพันธ์ระหว่างออปเจค ความหมาย : ความสัมพันธ์กัน การที่จะมีความสัมพันธ์กันนั้นจะมีกันหลายวิธีไม่ว่าจะเป็น การเรียกเมทธอด การเข้าถึงตัวแปร การสืบทอดคุณสมบัติ การส่งผ่านตัวแปรผ่านเมทธอด การคืนค่า (Return) และ การดักจับข้อผิดพลาด
กลุ่ม : ความสัมพันธ์	
ABSTR_R	คำอธิบาย : Number of abstractions คำอธิบาย : จำนวนแอ๊บแสตก

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

	ความหมาย : นับจำนวนความสัมพันธ์แบบแอ๊บแสตก
ASSOC_R	คำอธิบาย : Number of associations คำอธิบาย : จำนวนความเกี่ยวข้องกัน ความหมาย : นับจำนวนความสัมพันธ์แบบแอสโซซิเอชั่น
DEPEND_R	คำอธิบาย : Number of dependencies คำอธิบาย : จำนวนการขึ้นต่อกัน ความหมาย : นับจำนวนความสัมพันธ์ทุกชนิดกับคลาสอื่น

2.1.3 การเรียนรู้ของเครื่อง

การเรียนรู้ของเครื่องเป็นการสร้างฟังก์ชันจากข้อมูลสอน โดยข้อมูลสอนจะประกอบด้วยข้อมูลนำเข้า ซึ่งมักจะเป็นเวกเตอร์ และ ผลของการเรียนรู้ที่ต้องการ หลังจากการเรียนรู้จะได้ข้อมูลเป็นฟังก์ชันที่อาจจะให้ค่าต่อเนื่อง หรือ ค่าทำนายประเภทของวัตถุ โดยหน้าที่ของการเรียนของเครื่อง คือ การทำนายค่าของฟังก์ชันจากวัตถุเข้าที่ถูกต้องโดยใช้ตัวอย่างสอนจำนวนน้อย การเรียนรู้ของเครื่องมีข้อดีดังต่อไปนี้

a) วิธีการเรียนรู้ของเครื่องสามารถถอดความสัมพันธ์ และ ความเกี่ยวข้องที่สำคัญ ในกลุ่มของข้อมูลใหญ่
b) จำนวนของข้อมูลเรียนรู้มีจำนวนมากเกินความสามารถในการคำนวณโดยมนุษย์ เครื่องสามารถที่จะเรียนรู้ความรู้นี้ทีละเล็กทีละน้อย และ สามารถจับประเด็นได้ตรงตามที่ต้องการออกมา ได้มากกว่ามนุษย์
c) ข้อมูลมีการเปลี่ยนแปลงและเพิ่มเติมได้ตลอดเวลา เครื่องสามารถเรียนรู้ได้อย่างต่อเนื่องเมื่อมีข้อมูลเข้ามาใหม่

การเรียนรู้ของเครื่องที่ใช้ในงานวิจัยนี้ประกอบด้วย

a) **Random Forest [9]**

แรนดอมฟอเรสต์ ประกอบด้วยแผนผังต้นไม้ช่วยในการตัดสินใจ (Decision trees) จำนวนมาก โดยที่ต้นไม้แต่ละต้นจะได้รับการจำแนก และ ต้นไม้จำทำการ “เลือกโหวต” คลาสที่มีประสิทธิภาพ ซึ่งลักษณะดีของแรนดอมฟอเรสต์คือ ดำเนินการได้อย่างมีประสิทธิภาพต่อฐานข้อมูลขนาดใหญ่ สามารถนำตัวแปรนำเข้าปริมาณหลายพันตัวได้โดยไม่ต้องลบตัวแปร

ออกไปบางส่วน และ มีประสิทธิภาพในการประมาณค่าข้อมูลที่หายไปและมีความถูกต้องขณะที่ข้อมูลส่วนใหญ่หายไป

b) Naive Bayes

นาอีฟเบย์ (Naive-Bayes) เป็นเทคนิคที่ถูกตั้งชื่อตามโทมัส เบย์ (Thomas Bayes) เทคนิคแบบ Naive-Bayes โมเดลการคัดแยกประเภทข้อมูลที่ใช้หลักความน่าจะเป็น ที่สามารถคาดการณ์ผลลัพธ์ได้ เทคนิคนี้จะทำการวิเคราะห์ความสัมพันธ์ระหว่างตัวแปรอิสระ (Independent Variable) ที่มีต่อตัวแปรตาม (Dependent Variable) เพื่อใช้ในการสร้างเงื่อนไขความน่าจะเป็นสำหรับแต่ละความสัมพันธ์ Naive-Bayes เป็นเทคนิคในการแก้ปัญหาแบบการจำแนกประเภทซึ่งสามารถให้ผลลัพธ์ที่ดี และ รวดเร็ว ทำให้เทคนิคนี้เป็นเครื่องมือที่ดีในการสร้างแบบจำลอง และ หารูปแบบความสัมพันธ์ที่ไม่ซับซ้อน

c) Logistic regression

วิธีการทางสถิติที่ใช้ในการสร้างสมการ หรือแบบจำลองถดถอย ที่แสดงความสัมพันธ์ระหว่างตัวแปรอิสระ และ ตัวแปรทำนาย แล้วสามารถนำแบบจำลองถดถอยดังกล่าวไปใช้ประมาณค่าตัวแปรตามได้ ซึ่งมีความสามารถคล้ายกับการวิเคราะห์การถดถอย ที่สามารถใช้ทำนายตัวแปรหลายชนิด ไม่ว่าจะเป็นตัวเลขหรือหมวดหมู่

d) IB1

เป็นหนึ่งในวิธีการแยกประเภทโดยใช้ขั้นตอนวิธีการเพื่อนบ้านที่ใกล้ที่สุด (Nearest Neighbor algorithm) แยกประเภทจะใช้การวัดระยะทางง่ายๆ ในการหาตัวแปรที่จะนำมาใช้ในการเรียนรู้ และใช้ตัวแปรนั้นๆ ในการทำนายเช่นเดียวกัน หากกรณีมีหลายตัวแปรที่มีระยะทางที่ใกล้ที่สุดเท่าๆ กันเราจะเลือกตัวแปรตัวแรกที่พบมาใช้

e) IBk

เป็นหนึ่งในวิธีการแยกประเภทโดยใช้ขั้นตอนวิธีการเพื่อนบ้านที่ใกล้ที่สุด (Nearest Neighbor algorithm) แยกประเภทจะใช้การวัดระยะทางง่าย ๆ ในการหากลุ่มของตัวแปรที่จะนำมาใช้ในการเรียนรู้ และใช้ตัวแปรนั้นๆ ในการทำนาย โดย k คือ จำนวนเพื่อนบ้านที่ใกล้ที่สุดที่จะใช้ในการทำนาย

f) VFI

VFI (Voting Feature Intervals) จะมีการสร้างขอบเขตบน และ ล่างระหว่างแต่ละตัวแปรอิสระ และ ตัวแปรทำนาย แล้วนำไปทำการทำนาย ซึ่งในแต่ละรอบนั้นจะมีการลงคะแนนในการเลือกตัวแปรอิสระที่ดีที่สุด อัลกอริทึมนี้มีความเรียบง่ายเมื่อเทียบกับวิธีอื่น ๆ และมีความรวดเร็วอย่างมาก

g) J48

เป็นต้นไม้การตัดสินใจในการเรียนรู้การทำนายของเครื่อง ที่ตัดสินใจค่าเป้าหมาย (ตัวแปรตาม) ของกลุ่มตัวอย่างใหม่ขึ้นอยู่กับค่ารายละเอียดต่าง ๆ ของข้อมูลที่มีอยู่ โหนดภายในของต้นไม้การตัดสินใจที่แตกต่างกันจะแสดงถึงคุณลักษณะ กิ่งระหว่างโหนดบอกให้เราทราบค่าที่เป็นไปได้ ซึ่งลักษณะดีของ J48 คือ สามารถใช้กับชุดข้อมูลทดลอง ที่ไม่มีค่าตัวแปรอิสระบางตัวได้ และ สามารถใช้งานได้กับข้อมูลที่มีความผิดปกติหรือมีความเสียหาย

2.1.4 การเลือกสุ่มข้อมูลแบบความเที่ยงตรง

การเลือกสุ่มข้อมูลแบบความเที่ยงตรง (Cross Validation) คือ วิธีการแบ่งชุดข้อมูลในการทดลองเพื่อลดความผิดพลาดของโมเดลหรือวิธีการที่เรานำเสนอ โดยพื้นฐานของวิธีเลือกสุ่มข้อมูลแบบความเที่ยงตรงคือการสุ่มตัวอย่าง (Resampling) โดยเริ่มจากแบ่งชุดข้อมูลออกเป็นส่วน ๆ และนำบางส่วนจากชุดข้อมูลนั้นมาตรวจสอบโดยที่งานวิจัยนี้จะใช้วิธีการเลือกสุ่มข้อมูลแบบความเที่ยงตรงโดยแบ่งข้อมูลเป็น 10 ส่วน (10 - fold cross-validation)

วิธีเลือกสุ่มข้อมูลแบบความเที่ยงตรงโดยการแบ่งข้อมูลเป็น 10 ส่วน

ในการเลือกสุ่มข้อมูลแบบความเที่ยงตรงโดยแบ่งข้อมูลเราจะแบ่งข้อมูลออกเป็น 10 ชุดเท่าๆกัน และคำนวณค่าความผิดพลาด 10 รอบ โดยแต่ละรอบการคำนวณ ข้อมูลชุดหนึ่งจากข้อมูล 10 ชุดจะถูกเลือกออกมาเพื่อเป็นข้อมูลทดสอบ และข้อมูลอีก 9 ชุดจะถูกใช้เป็นข้อมูลสำหรับการเรียนรู้ และทำสลับอย่างนี้ไปเรื่อยๆ จนครบ 10 รอบ ในขั้นตอนสุดท้ายจะหาค่าเฉลี่ยของค่าความถูกต้องในแต่ละกลุ่ม วิธีการนี้ข้อมูลทุกชุดตัวอย่างจะได้เป็นทั้งชุดทดสอบและชุดเรียนรู้

จากงานวิจัยของ Kohavi [12] เขาได้ทำการทดลองเปรียบเทียบความสามารถของการเลือกสุ่มข้อมูลแบบความเที่ยงตรงหลายชนิดไม่ว่าจะเป็น การเลือกสุ่มข้อมูลแบบความเที่ยงตรงแบบทั่วไป (Regular cross-validation) การเลือกสุ่มข้อมูลแบบความเที่ยงตรงแบบเลือกออกหนึ่ง (Leave-one-out cross-validation) การเลือกสุ่มข้อมูลแบบความเที่ยงตรงแบบแบ่งข้อมูลเป็น K ชุด (K-fold cross-validation) และ การเลือกสุ่มข้อมูลแบบความเที่ยงตรงแบบบูตแสตร็ป เขาสรุปได้ว่าในการเลือกสุ่มข้อมูลแบบความเที่ยงตรงโดยแบ่งข้อมูลเราจะแบ่งข้อมูลออกเป็น 10 ชุด ชุดละเท่าๆ กันมีประสิทธิภาพมากที่สุด

2.1.5 การคำนวณประสิทธิภาพการทำนายร่องรอยที่ไม่ดี

ในงานวิจัยนี้ได้ใช้สถิติ 4 ชนิด มาคำนวณประสิทธิภาพการทำนายร่องรอยที่ไม่ดีด้วย ความถูกต้องการทำนายของการทำนายร่องรอยที่ไม่ดี (Bad-smell prediction accuracy) ความไว และ ความจำเพาะ (Sensitivity & Specificity) และ ค่าพยากรณ์ของการทดสอบ (Predictive value of tests)

2.1.5.1 ความถูกต้องในการทำนายร่องรอยที่ไม่ดี

ความถูกต้องในการทำนายร่องรอยที่ไม่ดี (Bad-smell prediction accuracy) เป็นผลลัพธ์ที่ได้จากการคำนวณของโปรแกรมเวกา (Weka) ซึ่งบ่งบอกความสามารถของความถูกต้องในการทำนายจาก

$$\text{ความถูกต้องในการทำนายร่องรอยที่ไม่ดี} = \frac{\text{จำนวนคลาสที่สามารถทำนายถูกต้อง}}{\text{จำนวนคลาสทั้งหมด}}$$

2.1.5.2 ความไว และ ความจำเพาะ

ความไว (Sensitivity) เป็นหนึ่งในวิธีการคำนวณประสิทธิภาพการทำนายร่องรอยที่ไม่ดี แสดงความสามารถของวิธีการทดสอบในการทำนาย ว่าคลาสใดมีร่องรอยที่ไม่ดีได้อย่างถูกต้อง การทดสอบที่มีค่าความไวมาก ก็ยิ่งมีความสามารถมากในการค้นหาคลาสที่มีร่องรอยที่ไม่ดีมากขึ้นด้วย มีสูตรการคำนวนดังนี้

$$\text{Sensitivity} = \frac{TN}{TN + FP}$$

ความจำเพาะ (Specificity) เป็นความสามารถของวิธีการทดสอบในการทำนายว่าคลาสใดไม่มีร่องรอยที่ไม่ดีได้อย่างถูกต้อง การทดสอบที่มีค่าความจำเพาะมากก็ยิ่งมีความสามารถมากในการค้นหาคลาสที่ไม่มีร่องรอยที่ไม่ดีมากขึ้นด้วย มีสูตรการคำนวนดังนี้

$$\text{Specificity} = \frac{TP}{TP + FN}$$

2.1.5.3 ค่าพยากรณ์ของการทดสอบ

ค่าพยากรณ์ของการทดสอบ (Predictive value of tests) เป็นความสามารถในการทำนายร่องรอยไม่ดีในคลาสว่ามี หรือ ไม่มีได้อย่างถูกต้อง

ค่าพยากรณ์ของการทดสอบทางบวก (Positive predictive value (PPV) เป็นสัดส่วนระหว่างคลาสที่ทำการทดสอบได้ผลเป็นการทำนายเป็นบวก หรือ ถูกทำนายเป็นคลาสไม่ดี ต่อด้วยจำนวนคลาสทั้งหมดที่เป็นคลาสที่มีร่องรอยที่ไม่ดี มีสูตรการคำนวนดังนี้

$$PPV = \frac{TP}{TP + FP}$$

ค่าพยากรณ์ของการทดสอบทางลบ Negative predictive value (NPV) เป็นสัดส่วนระหว่างคลาสที่ทำการทดสอบได้ผลเป็นการทำนายเป็นลบ หรือ ถูกทำนายเป็นคลาสดี ต่อด้วยจำนวนคลาสทั้งหมดที่เป็นคลาสที่ไม่มีร่องรอยที่ไม่ดี มีสูตรการคำนวนดังนี้

$$NPV = \frac{TN}{TN + FN}$$

2.1.6 การทดสอบสมมติฐานของงานวิจัย

การตั้งสมมติฐานทางสถิติ (Hypothesis) การตั้งข้อสงสัยของคำตอบ ของปัญหาการวิจัยอย่างมีเหตุผล โดยอาศัยแนวคิด หลักการ ประสบการณ์หรือทฤษฎีที่เกี่ยวข้อง ซึ่งในวิชาสถิติ เราเรียกข้อสงสัยดังกล่าวว่า สมมติฐาน และการพิสูจน์ข้อสงสัยเราก็เรียกว่า การทดสอบสมมติฐาน เช่นเดียวกัน การทดลอง เก็บข้อมูล แล้ววิเคราะห์ทางคณิตศาสตร์ และตัดสินใจที่จะยอมรับหรือปฏิเสธสมมติฐาน โดยอ้างอิงตามผลการวิเคราะห์ดังกล่าว ซึ่งเราเรียกกระบวนการทั้งหมดนี้ว่า การทดสอบสมมติฐาน (Hypothesis Testing) ซึ่งจัดว่าเป็นการประยุกต์ใช้หลักวิชาสถิติ ที่สำคัญ มีวิธีการดังนี้

1. ตั้งสมมติฐาน

สมมติฐานฐานหลัก (Null Hypothesis) เขียนสัญลักษณ์ด้วย (H0) เป็นสมมติฐานเบื้องต้นในสิ่งที่เราสนใจหาคำตอบ ถ้าหากไม่มีเหตุผลหรือหลักฐานเพียงพอ เราจะยอมรับสมมติฐานหลักไว้ก่อน

$$H0: \mu 1 = \mu 2$$

เมื่อ μ คือ ค่าเฉลี่ยของกลุ่มประชากร

สมมติฐานรอง หรือสมมติฐานทางเลือก (Alternative Hypothesis) เขียนสัญลักษณ์ด้วย (H1) เป็นสมมติฐานทางเลือกที่จะเป็นไปได้หากสมมติฐานหลักถูกปฏิเสธ ซึ่งแบ่งได้เป็น 2 ประเภท คือ

- สมมติฐานแบบไม่มีทิศทาง (non - directional hypothesis) ได้แก่ สมมติฐานที่ระบุความสัมพันธ์กันของกลุ่มสองกลุ่มตัวอย่างไม่เท่ากัน

$$H1: \mu 1 \neq \mu 2$$

เมื่อ μ คือ ค่าเฉลี่ยของกลุ่มประชากร

- สมมติฐานแบบมีทิศทาง (Directional hypothesis) ได้แก่ สมมติฐานที่ระบุความสัมพันธ์กันของกลุ่มสองกลุ่มตัวอย่างมากกว่าหรือน้อยกว่า ค่าใดค่าหนึ่ง หรือ ความสัมพันธ์ทางบวก ความสัมพันธ์ทางลบ

$$H1: \mu 1 > \mu 2 \text{ หรือ } H1: \mu 1 < \mu 2$$

เมื่อ μ คือ ค่าเฉลี่ยของกลุ่มประชากร

ซึ่งการตัดสินใจในการทดสอบสมมติฐานทางสถิตินั้นจะมี 4 ประเภท ประกอบด้วย

a) ยอมรับ H0 เมื่อสมมติฐานนั้นเป็นจริง ถือว่าตัดสินใจถูกต้อง หรือ True positive (TP)
b) ไม่ยอมรับ H0 ทั้งที่สมมติฐานนั้นเป็นจริง ถือว่าเป็นความผิดพลาดแบบ I (Type I Error) หรือ False negative (FN)
c) ยอมรับ H0 ทั้งที่สมมติฐานนั้นไม่เป็นจริง ถือว่าตัดสินใจเกิดความผิดพลาดแบบที่ II (Type II Error) หรือ False positive (FP)
d) ไม่ยอมรับ H0 เมื่อสมมติฐานนั้นไม่จริง ถือว่าตัดสินใจถูกต้อง หรือ True negative (TN)

ในตารางที่ 2.2 แสดงค่าความสัมพันธ์กันระหว่างค่าจริง และ ค่าที่ได้จากการทำนาย

ตารางที่ 2.2 แสดงค่าที่เป็นไปได้ของการทำนาย

		ค่าจริง	
		H0	H1
ค่าที่ตัดสินใจ	H0	TP	FP
	H1	FN	TN

2. กำหนดระดับนัยสำคัญ (Level of significance)

ระดับนัยสำคัญ (Level of significance) หรือ α หมายถึง ระดับความน่าจะเป็นในการที่จะปฏิเสธสมมติฐานหลัก (H0) มากน้อยเพียงใด ซึ่งจะสะท้อนถึงความเชื่อมั่นในการสรุปตามผลการทดสอบ หรือ เป็นการแสดงว่าสมมติฐานนั้นเชื่อถือได้มากน้อยเพียงใดนั่นเอง โดยทั่วไประดับนัยสำคัญทางสถิติจะกำหนดไว้ที่ 0.05 หรือ 0.01 ระดับนัยสำคัญที่ 0.05 หมายถึง โอกาสที่ไม่เป็นไปตามสมมติฐานมีเพียง 5 ส่วนใน 100 ส่วน หรือ คลาดเคลื่อนไม่เกิน 5%

3. คำนวณค่าสถิติ

ในงานวิจัยนี้ใช้การวิเคราะห์ผลแบบ Pair Sample T-test เป็นการเปรียบเทียบค่าเฉลี่ยระหว่างสองตัวแปร 2 กลุ่มที่ไม่เป็นอิสระต่อกันว่ามีความแตกต่างกันหรือไม่ เช่น ผลการทดสอบก่อนและหลังเรียน เมื่อแบบทดสอบเป็นฉบับเดียวกัน มีสูตรคือ

$$t = \frac{\sum D}{\sqrt{\frac{N\sum D^2 - (\sum D)^2}{(N-1)}}}$$

เมื่อ D คือ ความแตกต่างระหว่างคะแนนแต่ละคู่

N คือ จำนวนคู่

4. การตัดสินสมมติฐาน

เมื่อได้ค่า t แล้วให้เปรียบเทียบกับค่า t วิกฤตในตารางที่ DOF (Degree of freedom) ตามจำนวนตัวแปร และ ระดับความเชื่อมั่นที่ต้องการ ถ้าค่า |t| ที่คำนวณได้มีค่าน้อยกว่าค่าในตารางแปลว่า สมมติฐานหลักที่ตั้งมาถูกต้อง แต่ถ้าค่า t มากกว่าค่าในตารางแปลว่า ปฏิเสธสมมติฐานฐานหลัก หรือ ยอมรับสมมติฐานรอง

2.2. งานวิจัยที่เกี่ยวข้อง

2.2.1 งานวิจัย "Product Metrics for Automatic Identification of "Bad Smell" Design Problems in Java Source-Code" [3]

Matthew James Munro ได้เสนอวิธีการตรวจสอบร่องรอยที่ไม่ดีของโปรแกรมภาษาจาวาโดยอัติโนมัติ โดยที่วิธีการที่ตรวจสอบร่องรอยที่ไม่ดีของ Matthew นั้นมีสามขั้นตอนคือ เลือกร่องรอยที่ไม่ดี (Bad Smell Name) ที่ได้อธิบายโดย ฟาวเลอร์มาแล้ว ในขั้นตอนที่สองทำโดยการอธิบายคุณลักษณะ (Characteristics) ของร่องรอยที่ไม่ดีนั้นๆ โดยอ้างอิงจากงานวิจัยเก่าๆ ที่เคยทำมาแล้ว และในขั้นตอนสุดท้ายคือการออกแบบวิธีการที่ใช้ในการตรวจจับร่องรอยที่ไม่ดี (Design Heuristics) โดยอ้างอิงจากงานวิจัยเก่าๆ ที่เคยทำมาแล้วเช่นเดียวกัน โดยที่ Matthew ได้ทำการเสนอร่องรอยที่ไม่ดีที่จะใช้ในการตรวจสอบ สองร่องรอยคือ

a) Lazy Class

ร่องรอยที่ไม่ดี: คลาสที่ไม่ได้ทำอะไรมากต้องถูกกำจัด

คุณลักษณะ: คลาสที่คาดว่าในอนาคตจะไม่มีการเพิ่มงานในอนาคต

วิธีการที่ใช้ในการตรวจจับ: มีคลาสจำนวนมากที่ไม่ได้ทำงานมาก

มาตรวัด: จำนวนเมธอด number of methods (NOM) จำนวนเมธอดต่อคลาส Weighted Methods per Class (WMC) และ จำนวนบรรทัด Line of Code (LOC)

b) Temporary Field

ร่องรอยที่ไม่ดี: ออบเจคที่มีตัวแปรที่ถูกประกาศปรากฏเพียงชั่วคราว

คุณลักษณะ: ตัวแปรของคลาสที่ปรากฏ และ ถูกใช้ ณ เวลาหนึ่งๆ เท่านั้น

วิธีการที่ใช้ในการตรวจจับ: ร่องรอยที่ไม่ดีของ Temporary Field มีเกิดขึ้นเมื่อมีตัวแปรชั่วคราวที่ถูกสร้างขึ้นในคลาสที่มีร่องรอยไม่ดีของ Large Class

มาตรวัด: จำนวนเมธอดที่เรียกใช้ตัวแปรชั่วคราวที่ถูกสร้างในคลาส Instance variable defined in a class (IVMC)

จากนั้นเขาได้สร้างเครื่องมือสำหรับการตรวจจับร่องรอยที่ไม่ดีขึ้นในภาษาจาวา โดยใช้เครื่องมือในการคำนวณเพื่อที่จะลดความผิดพลาด และ ช่วยให้สามารถวิเคราะห์ระบบขนาดใหญ่ได้อย่างรวดเร็ว และ สามารถนำข้อมูลกลับมาใช้ใหม่ได้ สถาปัตยกรรมทั่วไปของเครื่องมือ คือ นำซอร์สโค้ดภาษาจาวาไปคำนวณมาตรวัด แล้วนำค่ามาตารวัดไปใช้เป็นข้อมูลของการตรวจจับร่องรอยที่ไม่ดี ในขั้นตอนสุดท้ายเขาได้นำเครื่องมือที่เขาสร้างไปวัดผล โดยการนำเครื่องมือที่เขาได้สร้างขึ้นมาทดลองด้วยระบบจองห้องพักขนาดเล็ก และ ระบบการเขียนกราฟขนาดกลาง

ผลลัพธ์ที่ได้จากงานวิจัยนี้คือ มาตรวัด และ เครื่องมือสำหรับการตรวจจับร่องรอยที่ไม่ดี 2 ชนิดคือ Lazy Class และ Temporary Field ที่สามารถทำงานได้อย่างอัตโนมัติ

2.2.2 งานวิจัย "Detecting Design Flaws via Metrics in Object-Oriented Systems" [4]

งานวิจัยนี้ [4] นำเสนอมาตรวัดสำหรับตรวจจับร่องรอยที่ไม่ดีในระบบเชิงวัตถุ และนำเสนอขั้นตอนการออกแบบวิธีการสำหรับการตรวจจับร่องรอยไม่ดีบนซอร์สโค้ด 5 ขั้นตอน ซึ่ง 3 ขั้นตอนแรก คือ กำหนดแรงจูงใจ (Motivation) กำหนดวิธีการวัด (Strategy) การออกแบบมาตรวัด (Metrics) จะอธิบายถึงนิยามของเทคนิคการตรวจจับนี้ ส่วน 2 ขั้นตอนสุดท้าย คือ การคำนวณค่า (Measurement) และการตรวจสอบผลที่พบ (Finding) รวมทั้งการนำเทคนิคนี้ไปประยุกต์ใช้กับกรณีศึกษา โดยผู้วิจัยได้นำวิธีการนี้มาประยุกต์ใช้กับร่องรอยที่ไม่ดี 2 วิธีคือ Data Class และ God Class ซึ่งมีรายละเอียดดังนี้

a) Data Class

คือ คลาสที่มีทำหน้าที่หลักใน การกำหนดค่าให้กับ

แรงจูงใจ : คลาสขาดความมีหน้าที่ของเมทธอด ทำให้บำรุงรักษาซอฟต์แวร์ทำได้ยาก เกิดซอร์สโค้ดที่ซ้ำซ้อนขึ้น และ ทำความเข้าใจคลาสยากขึ้น

วิธีการวัด : หาคลาสที่เรียกว่า ไลท์เวทคลาส (Lightweight classes)

มาตรวัด : จำนวนของเมทธอดต่อคลาสหารด้วยจำนวนสมาชิกของอินเตอร์เฟสทั้งหมดที่ไม่ได้สืบทอดมา (Weight of a class (WOC)) จำนวนคุณลักษณะที่ไม่ได้ถูกสืบทอด (Number of public attribute (NOPA)) และ จำนวนเอคเซสเซอร์เมทธอดที่ไม่ได้ถูกสืบทอดมา (Number of accessor method (NOAM))

b) God Class

คลาสที่ทำหน้าที่ควบคุมการทำงานหลักทั้งหมดในระบบ

แรงจูงใจ : คลาสมีการทำงานหลักหลายหน้าที่ ส่งผลทำให้สามารถนำคลาสกลับมาใช้ใหม่ได้น้อย และ เกิดความยากในการทำความเข้าใจคลาส

วิธีการวัด : จำนวนการเข้าถึงข้อมูลไลท์เวทคลาส ทั้งเข้าถึงโดยตรงและผ่านเอคเซสเซอร์เมทธอด และ มีพฤติกรรมที่ไม่มีการสื่อสารมาก แสดงว่าไม่มีการกระจายการทำงานระหว่างคลาส

มาตรวัด : จำนวนการเข้าถึงข้อมูลคลาสภายนอก (Access of foreign data (AOFD)) ค่าจำนวนของเมทธอดในคลาส (Weighted method count (WMC)) และ การเรียกเมทธอดที่เข้าถึงข้อมูลโดยตรง (Tight class cohesion (TCC))

จากนั้นได้สร้างเครื่องมือสำหรับตรวจจับร่องรอยที่ไม่ดีขึ้น ประกอบด้วย 2 ส่วนคือ TableGen สำหรับเก็บข้อมูลการออกแบบที่ต้องการจากโปรแกรมภาษาซีพลัสพลัส (C++) เช่น คลาส เมทธอด ความสัมพันธ์ระหว่างคลาส การเรียกใช้เมทธอด มาเก็บในรูปแบบของตาราง และนำมาตรวัดมาคำนวณโดยใช้เครื่องมือฐานข้อมูลเชิงสัมพันธ์ (Relational database engine) ซึ่งในงานวิจัยนี้เลือกใช้ ฐานข้อมูลออราเคิล (Oracle) เพื่อสร้างชุดคำสั่งเอสคิวแอล (SQL) สำหรับคำนวณมาตรวัดที่เลือกใช้ โดยดึงข้อมูลจากตารางเหล่านั้น

ผลลัพธ์ที่ได้จากงานวิจัยนี้ คือ มาตรวัดสำหรับการตรวจจับร่องรอยที่ไม่ดี 2 วิธี คือ Data Class และGod Class ซึ่งสามารถนำมาตรวัดนี้ไปใช้กับกรณีศึกษาอื่นๆ เพื่อตรวจสอบความน่าเชื่อถือของมาตรวัดนี้ และสามารถนำขั้นตอนการออกแบบวิธีการสำหรับการตรวจจับร่องรอยที่ไม่ดีนี้ไปประยุกต์ใช้ เพื่อออกแบบวิธีการสำหรับการตรวจจับร่องรอยที่ไม่ดีอื่นๆ

2.2.3 **งานวิจัย** “Bad-smell Detection using Object-Oriented Software Metrics” [5]

งานวิจัยนี้ [5] ได้ออกแบบมาตรวัดซอฟต์แวร์เชิงวัตถุในการตรวจจับร่องรอยที่ไม่ดี และได้แนะนำวิธีในการทำรีแฟคทอริงที่ใช้กับร่องรอยไม่ดีทั้ง 6 ประเภท คือ Long Method, Large Class, Long Parameter List, Switch Statement, Lazy Class และ Feature Envy โดยการกำหนดนิยามของมาตรวัดสำหรับการตรวจจับ และ การหาตำแหน่งของร่องรอยที่ไม่ดี โดยจะมีขั้นตอนทั้งหมด 6 ขั้นตอนดังต่อไปนี้

1. นิยามร่องรอยที่ไม่ดี และ อธิบายลักษณะของร่องรอยที่ไม่ดีแต่ละประภท
2. อธิบายแรงจูงใจ รายละเอียดผลกระทบของร่องรอยที่ไม่ดีต่อซอฟต์แวร์ และอธิบายวิธีการสังเกตถึงวิธีการพอร่อยรอยที่ไม่ดีได้อย่างไร
3. อธิบายวิธีการวัดที่ใช้ในการหาร่องรอยที่ไม่ดี
4. กำหนดมาตรวัดสำหรับการตรวจจับร่องรอยที่ไม่ดี โดยใช้มาตรวัดที่มีอยู่แล้ว และ สร้างมาตรวัดขึ้นมาใหม่
5. การกำหนดค่าของการพิจารณา หรือ การกำหนดช่วงของค่ามาตรวัดที่ระบุได้ว่าเป็นร่องรอยที่ไม่ดี
6. การประยุกต์วิธีรีแฟคทอริง คือ ข้อเสนอแนะวิธีการรีแฟคทอริงสำหรับปรับปรุงซอฟต์แวร์เพื่อใช้ในการกำจัดร่องรอยที่ไม่ดี

ผู้วิจัยได้สร้างเครื่องมือในการตรวจจับร่องรอยที่ไม่ดีสำหรับภาษาจาวาด้วยมาตารวัดที่ได้ออบแบบในงานวิจัยนี้ และได้นำเครื่องมือที่ได้มาประยุกต์ใช้กับโปรแกรมตัวอย่าง ผลการทดสอบผู้ทำการวิจัย สรุปได้ว่าหลังจากการตรวจสอบร่องรอยที่ไม่ดี และประยุกต์ใช้วิธีรีแฟคทอริงที่ได้นำเสนอ ทำให้ร่องรอยที่ไม่ดีที่มีอยู่ในโปรแกรมได้ถูกแก้ไข

2.2.4 งานวิจัย “Design and implementation of a tool for detecting bad smells in Java program” [6]

งานวิจัยนี้ [6] นำเสนอวิธีการค้นหาร่องรอยที่ไม่ดีสำหรับซอร์สโค้ดภาษาจาวา จากการพิจารณาองค์ประกอบของโปรแกรม และ ใช้วิธีการของโปรล็อกในการค้นหาโดยเลือกวิธีการแสดงความรู้ตรรกะเพรดิเคต ซึ่งแบ่งขั้นตอนได้ 4 ขั้นตอนดังต่อไปนี้

1. เลือกร่องรอยที่ไม่ดี

ในงานวิจัยนี้จะนำเสนอวิธีการค้นหาร่องรอยที่ไม่ดี 4 ประเภทคือ Feature Envy, Message Chains, Middle Man และ Inappropriate Intimacy (General Form) ซึ่งร่องรอยที่ไม่ดีทั้ง 4 ประเภทนี้เหมาะสมกับหลักการของภาษาโปรล็อก

2. ออกแบบวิธีการหาร่องรอยที่ไม่ดี

ออกแบบเพรดิเคตเพื่อใช้แทนความรู้ของความสัมพันธ์และองค์ประกอบของร่องรอยที่ไม่ดีในแต่ชนิด ของภาษาจาวา แล้วนำเพรดิเคตของร่องรอยที่ไม่ดีในแต่ชนิดที่ได้ออกแบบไว้มาสร้างเป็นกฎโปรล็อก ใช้ภาษาโปรล็อกในการค้นหาร่อยรอยที่ไม่ดี

3. ออกแบบ และ พัฒนาเครื่องมือค้นหาร่องรอยที่ไม่ดี

เครื่องมือที่ใช้ในการค้นหาร่องรอยที่ไม่ดีนั้นจะใช้แผนภาพยูสเคส (Use case Diagram) แผนภาพคลาส (Class Diagram) และ แผนภาพซีเควนซ์ (Sequence Diagram) แปลงเป็นเพรดิเคตแล้วใช้กฎโปรล็อกในการค้นหาร่อยรอยที่ไม่ดี

4. ทดสอบความน่าเชื่อถือของเครื่องมือ

หลังจากนั้นได้นำเครื่องมือที่ได้ออกแบบ และ พัฒนา มาประเมินความสามารถจากการทดสอบกับโปรแกรมต้นฉบับภาษาจาวา 3 โปรแกรม เปรียบเทียบค่าที่ได้จากการใช้เครื่องมือกับการค้นหาด้วยบุคคล เขาได้สรุปผลการทดสอบพบว่า วิธีการนี้สามารถค้นหาร่องรอยที่ไม่ดีได้ผลเป็นอย่างดี

2.2.5 งานวิจัย “Using Declarative Meta Programming for Design Flaws Detection in Object-Oriented Software” [7]

งานวิจัยนี้ [7] มีเป้าหมายเพื่อที่จะแนะนำวิธีการตรวจสอบข้อบกพร่องของระบบการออกแบบเชิงวัตถุแบบอัตโนมัติในการออกแบบภาษาจาวาซอร์สโค้ด และนำเสนอวิธีการค้นหาร่องรอยที่ไม่ดีสำหรับโปรแกรมแบบเชิงวัตถุที่ระดับเมต้า (Meta-level) โดยวิธีการนี้จะช่วยให้นักพัฒนาและ ผู้ดูแลระบบสามารถหาข้อบกพร่องในการออกแบบระบบซอฟต์แวร์เพื่อที่จะหลีกเลี่ยงวิธีการออกแบบโปรแกรมที่ไม่ดี ซึ่งวิธีการตรวจร่องรอยที่ไม่ดีที่งานวิจัยนี้เสนอจะประกอบด้วยร่องรอยที่ไม่ดี 3 ประเภท คือ Message Chains, Inappropriate Intimacy, Middle Man โดยงานวิจัยนี้แบ่งขั้นตอนได้ดังต่อไปนี้

1. สร้างกฎและความจริง

ใช้จาวาคอมไพเลอร์ในการแยกซอร์สโค้ดภาษาจาวา และสร้างโครงสร้างต้นไม้ ซึ่งในขั้นตอนนี้ข้อมูลการออกแบบ และ ข้อบกพร่องของการออกแบบจะถูกสกัดออกมาจากโครงสร้างต้นไม้ กฎการอนุมานโปรล็อกที่ใช้ในการตรวจจับร่องรอยที่ไม่ดีจะถูกกำหนดสร้างขึ้นจากกรณีศึกษา และ กำหนดความหมายโดเมนเพื่อที่อธิบายขอบเขตในการตรวจจับร่องรอยที่ไม่ดีในงานที่เขียนด้วยภาษาจาวา ในขั้นตอนการตรวจสอบนั้นเขาใช้ซอร์สโค้ด JHotdraw มาแปลงให้เป็นตรรกะข้อเท็จจริงตามกฎที่กำหนดไว้

2.วิธีการตรวจสอบข้อบกพร่อง

จากกรณีศึกษาแสดงให้เห็นว่าการออกแบบข้อบกพร่องสามารถตรวจพบตามกฎที่กำหนดไว้ จากการทดลองกฎตรรกะและกฎการอนุมานกับซอร์สโค้ดหลาย ๆ โปรแกรมแล้วนำมาเปรียบเทียบกับลักษณะของข้อบกพร่องออกแบบแสดงให้เห็นว่าเทคนิคที่นำเสนอสามารถตรวจหาข้อบกพร่องในการออกแบบภาษาจาวาซอร์สโค้ดได้เป็นอย่างดี นอกจากนี้วิธีการตรวจสอบในระดับเมตาสามารถสร้างกฎที่เรียบง่าย และ ตรงไปตรงมาค้นหาในการค้นหาระบบที่มีความซับซ้อนสูง

จากการทดสอบสรุปได้ว่าตัวอย่างการออกแบบการตรวจสอบข้อบกพร่อง แสดงให้เห็นว่าวิธีการที่เสนอสามารถตรวจพบข้อบกพร่องการออกแบบ ความคิดที่นำเสนอสามารถที่จะตรวจจับข้อบกพร่องในการออกแบบซอฟต์แวร์เชิงวัตถุ

2.2.6 งานวิจัย "Predicting Source-Code Complexity at the Design Stage" [1]

Sallie Henry and Calvin Selig ทำนายคุณภาพของซอร์สโค้ดตั้งแต่เริ่มการออกแบบซอฟต์แวร์ เพื่อที่จะลดขนาดลูปของกระบวนการผลิตซอฟต์แวร์ และ เพื่อเพิ่มประสิทธิภาพของกระบวนการพัฒนา โดยใช้มาตรวัดซอร์สโค้ดที่ใช้ คือ Lines of code, Software science, Cyclomatic complexity และมาตรวัดของการออกแบบ Fan-in และ Fan-out โดยวิธีการวิเคราะห์รีเกรสชัน (Regression Analysis) ในการทำนายความซับซ้อนของซอร์สโค้ด

มาตรวัดเชิงปริมาณ :

1.มาตรวัดซอร์สโค้ด

-จำนวนบรรทัด (LOC : Lines of Code)

เป็นวิธีการนับจำนวนบรรทัดของซอร์สโค้ด ในงานวิจัยนี้จะนับจำนวนบรรทัดระหว่างเครื่องหมายอัฒภาค (semicolons (;)) โดยจะรวมทั้งบรรทัดที่ไม่สามารถคำนวนได้ ส่วนหัวของโปรแกรมและการประกาศตัวแปร

- มาตรวัดซอฟต์แวร์วิทยาศาสตร์ (Halstead's Software Science)

ในงานวิจัยนี้ใช้สามในเก้าของมาตรวัดซอฟต์แวร์วิทยาศาสตร์

1. N จำนวนตัวถูกดำเนินการ (operands) กับ ตัวดำเนินการ (operators)
2. V จำนวนบิทที่ถูกเก็บในเมโมรี่คำนวณได้จากสูตร

$$V = N \log_2 (n)$$

เมื่อ N จำนวนตัวถูกดำเนินการ กับ ตัวดำเนินการ
n จำนวนตัวถูกดำเนินการ กับ ตัวดำเนินการที่ไม่ซ้ำกัน

3. E แสดงถึงความยากง่ายของขั้นตอนวิธีการดำเนินการคำนวณได้จากสูตร

$$E = V \times D$$

เมื่อ V จำนวนบิทที่ถูกเก็บในเมโมรี่
D คำนวณได้จากสูตร $D = (n1/2) \times (N2/n2)$

เมื่อ n1 จำนวนตัวถูกดำเนินการที่ไม่ซ้ำกัน
n2 จำนวนตัวดำเนินการที่ไม่ซ้ำกัน

N2 จำนวนตัวถูกดำเนินการ กับ ตัวดำเนินการ

- Cyclomatic complexity

เป็นมาตรวัดบ่งชี้ถึงความซับซ้อนของโปรแกรม คำนวณจากจำนวนเส้นทางการทำงานของซอร์สโค้ด

2.มาตรวัดการออกแบบ

-Fan-in/Fan-out

เป็นมาตรวัดที่ถูกสร้างโดย Sallie Henry และ Dennis Kafura เพื่อนับจำนวนข้อมูลที่ส่งผ่านกันไปมา

เมื่อ Fan-in คือ จำนวนข้อมูลที่ถูกส่งผ่านภายในโพซิเยอร์ (Procedure) และ จำนวนการเรียกข้อมูลโกบอล

Fan-out คือ จำนวนข้อมูลที่ถูกส่งผ่านระหว่างโพซิเยอร์ (Procedure) และ จำนวนการแก้ไขข้อมูลโกบอล

การทดลอง :

ในการทำนายคุณภาพของซอร์สโค้ด ในงานวิจัยนี้จะประกอบด้วยขั้นตอนดังต่อไปนี้

1.การเก็บข้อมูล

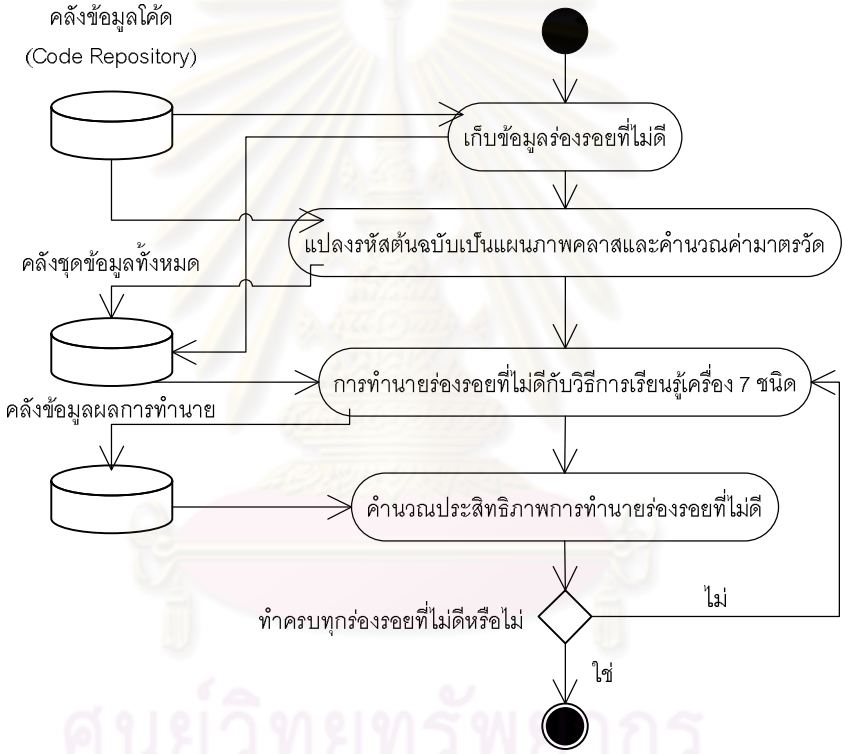
เขาได้เก็บรวบรวมซอร์สโค้ดโปรแกรมที่เขียนด้วยภาษา Ada-like และซอร์สโค้ดปาสคาลที่ได้มาจากการเรียนการสอนระดับปริญญาตรี และ หลักสูตรวิศวกรรมซอฟต์แวร์ที่ทั้งสถาบันเวอร์จิเนียโปลีเทคนิคและมหาวิทยาลัยวิสคอนซินที่เมืองลาครอส ซึ่งหลักสูตรได้รับการออกแบบเพื่อมุ่งเน้นในการสอนนักศึกษาเกี่ยวกับพื้นฐานของวิศวกรรมซอฟต์แวร์ โปรเจคที่เสร็จสมบูรณ์มีความหลากหลาย ซึ่งขนาดของซอร์สโค้ดมีขนาดตั้งแต่ 2,000 ถึง 8,000 บรรทัด หลังจากนั้นก็คำนวณค่ามาตรวัดของการออกแบบและคำนวณมาตรวัดในแต่ละโปรเจค

2.การวิเคราะห์ข้อมูล

การวิเคราะห์ข้อมูลจะใช้ McCabes cyclomatic-complexity เป็นหลักโดยใช้การวิเคราะห์การถดถอยแบบเชิงเส้น (Linear regression) ผลการวิจัยของบทความระบุว่าการออกแบบที่มีการปรับขอบเขตบน และ ขอบเขตล่างให้มีความกว้างสูง ความสัมพันธ์กันระหว่างมาตรวัดของการออกแบบ และ มาตรวัดของซอร์สโค้ดจะต่ำ แต่เมื่อปรับขอบเขตบน และ ขอบเขตล่างให้มีความกว้างน้อยลง ความสัมพันธ์กันระหว่างมาตรวัดของการออกแบบ และ มาตรวัดของซอร์สโค้ดก็จะมากขึ้น

บทที่ 3
การออกแบบขั้นตอนการดำเนินงาน

งานวิจัยนี้สามารถแบ่งออกเป็น 4 ขั้นตอน ดังแสดงด้วยแผนภาพแอคทิวิตี (Activity diagram) ใน รูปที่ 3.1 ได้แก่ การเก็บข้อมูลร่องรอยที่ไม่ดีจากซอฟต์แวร์ที่ได้จากงานวิจัยที่ผ่านมา แปลงรหัสต้นฉบับเป็นแผนภาพคลาสและคำนวณค่ามาตรวัด การทำนายร่องรอยที่ไม่ดี และ การคำนวณประสิทธิภาพการทำนายร่องรอยที่ไม่ดี ในบทนี้จะอธิบายถึง ขั้นตอนที่ 1 ถึง 3 ซึ่งเป็นรายละเอียดเกี่ยวกับวิธีการทำนายร่องรอยที่ไม่ดี ส่วนรายละเอียดขั้นตอนที่ 4 อยู่ในบทที่ 4



รูปที่ 3. 1 ขั้นตอนของงานวิจัยทั้งหมด

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

3.1 การเก็บข้อมูลร่องรอยที่ไม่ดี

ในงานวิจัยนี้ เก็บรวบรวมข้อมูลจากชุดซอฟต์แวร์ของงานวิจัยก่อนหน้านี้ [5][6] เข้าไปในคลังข้อมูลโค้ด ซอฟต์แวร์ของงานวิจัยก่อนหน้าจะประกอบด้วยซอร์สโค้ด และ จำนวนร่องรอยที่ไม่ดี ซึ่งตัวแปรทำนายที่เราสนใจ คือ จำนวนร่องรอยที่ไม่ดีในแต่ละคลาส ร่องรอยที่ไม่ดีที่เราสนใจจะประกอบด้วย Lazy Class, Feature Envy, Middle Man, Message Chains, Long Method, Long Parameter Lists และ Switch Statement

จำนวนร่องรอยที่ไม่ดีทั้ง 7 ชนิดในแต่ละคลาสจะถูกเก็บรวบรวมเข้าไปในคลังชุดข้อมูลทั้งหมด ส่วนคลังข้อมูลโค้ดทั้งหมดจะถูกนำไปสกัดจำนวนมาตรวัดการออกแบบในขั้นตอนถัดไป

3.2 การแปลงรหัสต้นฉบับแผนภาพคลาสและคำนวณค่ามาตรวัด

เราใช้ค่ามาตรวัดแผนภาพคลาสที่เราอธิบายในส่วนก่อนหน้านี้ เป็นตัวแปรนำเข้าในการทำนายร่องรอยที่ไม่ดี ดังนั้นข้อมูลแต่ละชุดจากงานวิจัยก่อนหน้านี้จะถูกสกัดได้ค่ามาตรวัดการออกแบบทั้ง 27 มาตรวัดที่ได้อธิบายไว้ในบทที่แล้ว โดยใช้โปรแกรมที่ชื่อว่า MagicDraw 9.5 SP1.1 ซึ่งเป็นเครื่องมือสำหรับย้อนรอยทางวิศวกรรม (Reverse engineering tool) ซึ่งผลที่ได้จากการใช้เครื่องมือนี้ คือ โมเดลการออกแบบ และ ค่ามาตรวัด รายละเอียดของการคำนวณค่ามาตรวัดของคลาส ได้แสดงในภาคผนวก ก

ขั้นตอนวิธีของการคำนวณค่ามาตรวัดของคลาส เริ่มด้วยการนำเข้าข้อมูล แล้วใช้โปรแกรมในการทำวิศวกรรมย้อนกลับ เพื่อแปลงซอร์สโค้ดให้เป็นโมเดลการออกแบบ จากนั้นใช้โปรแกรมตัวเดิมในการสกัดค่ามาตรวัดการออกแบบ

ตัวอย่างการคำนวณค่ามาตรวัดของคลาส รูปที่ 3.2 ตัวอย่างรหัสต้นฉบับซอฟต์แวร์ภาษาจาวาที่มีร่องรอยไม่ดีประเภท Message Chains ประกอบด้วย 3 คลาสคือ Person, TelephoneNumber และ Martin [2] ซอฟต์แวร์จะถูกทำวิศวกรรมย้อนกลับโดยใช้โปรแกรมที่ชื่อว่า MagicDraw 9.5 SP1.1 ซึ่งจะได้โมเดลการออกแบบแสดงใน รูปที่ 3.3 หลังจากนั้นนำโมเดลการออกแบบมาทำการสกัดค่ามาตรวัดการออกแบบ ซึ่งจะได้ค่ามาตรวัดทั้ง 27 มาตรวัด แสดงในตารางที่ 3.1

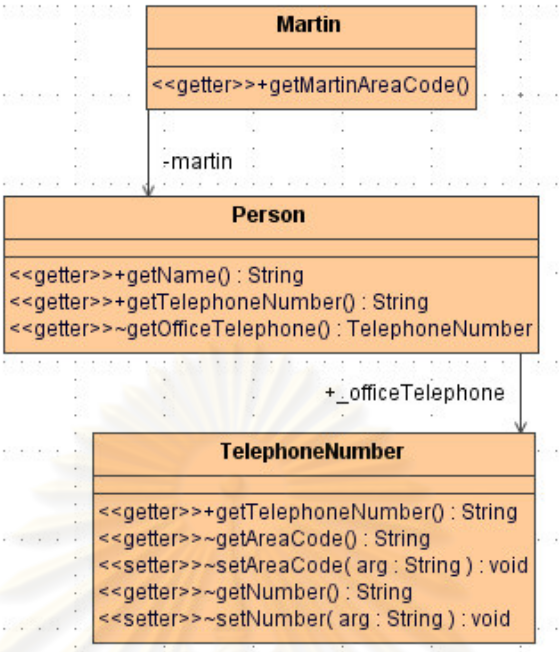
```
public class TelephoneNumber{
	private String _number;
	private String _areaCode;
	public String getTelephoneNumber() {
		return ("(" + _areaCode + ") " + _number);
	}
	String getAreaCode() {
		return _areaCode;
	}
	void setAreaCode(String arg) {
		_areaCode = arg;
	}
	String getNumber() {
		return _number;
	}
	void setNumber(String arg) {
		_number = arg;
	}
}

public class Person{

	public String _name;
	public TelephoneNumber _officeTelephone = new
TelephoneNumber();

	public String getName() {
		return _name;
	}
	public String getTelephoneNumber(){
		return _officeTelephone.getTelephoneNumber();
	}
	TelephoneNumber getOfficeTelephone() {
		return _officeTelephone;
	}
}
public class Martin{
	private Person martin = new Person();
	public getMartinAreaCode(){
		martin.getOfficeTelephone().setAreaCode ("781");
		martin.getTelephoneNumber();
	}
}
```

รูปที่ 3. 2 ตัวอย่างรหัสต้นฉบับภาษาจาวาที่มีร่องรอยไม่ดีประเภท Message Chains



รูปที่ 3. 3 ตัวอย่างรหัสต้นฉบับภาษาจาวาถูกทำวิศวกรรมย้อนกลับ

ตารางที่ 3. 1 ค่ามาตรวัด 27 มาตรวัดที่ได้จากการสกัดโมเดลการออกแบบ

Element	NA	NC	NM	NO	NP	C_PARAM	RFC	WAC	WMC	D_APPEAR	DIT	NOC	NAI	NOI	AHF	AIF	CF	MHF	MIF	PF	ACT	COMP	NS	CBC	ABSTR_R	ASSOC_R	DEPEND_R
Class Martin()	0	0	1	1	0	0	1	0	0	3	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
Class Person()	0	0	3	3	0	0	3	0	3	3	1	0	0	0	0	0	0	0	0	0	0	0	0	2	0	2	0
Class TelephoneNumber()	0	0	5	5	1	1	5	0	7	3	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	2	0

การใช้เครื่องมือสำหรับย้อนรอยทางวิศวกรรมจากซอร์สโค้ด เพื่อที่จะนำโมเดลการออกแบบไปใช้ในการทดลองไม่ใช่เรื่องที่แปลกใหม่ เพราะวิธีการนี้ถูกใช้ในงานวิจัย [8], [13] และ [14] โดยวิธีการสกัดค่ามาตรวัดแผนภาพคลาส ได้อธิบายไว้ในภาคผนวก ก หลังจากการสกัดค่ามาตรวัดแล้วจะรวบรวมชุดข้อมูลเป็น 7 ชุดไว้ในคลังชุดข้อมูลทั้งหมด แยกไปตามแต่ละชนิดของร่องรอยที่ไม่ดี ซึ่งแต่ละชุดข้อมูลจะประกอบด้วยจำนวนร่องรอยที่ไม่ดี และ ค่ามาตรวัดที่ได้ในแต่ละคลาส

3.3 การทำนายร่องรอยที่ไม่ดีกับวิธีการเรียนรู้เครื่อง 7 ชนิด

ในงานวิจัยนี้ ใช้วิธีการเรียนรู้เครื่อง 7 ชนิดที่อยู่ในโปรแกรมเวกา (Weka) ในการทำนายร่องรอยที่ไม่ดีในแต่ละชุดข้อมูล วิธีการเรียนรู้เครื่องประกอบด้วย Naive Bayes, Logistic, IB1, IBk, VFI, J48 และ Random forest โดยวิธีการทำนายร่องรอยไม่ดีมีรายละเอียดดังต่อไปนี้

-แต่ละชุดข้อมูลที่ประกอบด้วยจำนวนร่องรอยที่ไม่ดีแต่ละชนิด และ ค่ามาตรวัดในแต่ละคลาสจะถูกเรียนรู้ และ ทำนายโดยวิธีการเรียนรู้เครื่อง 7 ชนิด โดยการระบุคุณสมบัติของวิธีการเรียนรู้เครื่องตามค่าเริ่มต้น ซึ่งวิธีการทำนายนั้นได้อธิบายไว้ในภาคผนวก ข

-ใช้การเลือกสุ่มข้อมูลแบบความเที่ยงตรง (Cross Validation) โดยการแบ่งชุดข้อมูลเป็น 10 ส่วนเพื่อลดความผิดพลาดของวิธีการที่เรานำเสนอ ซึ่งวิธีการนี้เป็นคุณสมบัติของโปรแกรมเวกา (Weka) ที่แบ่งข้อมูลเป็น 10 กลุ่ม โดย 9 กลุ่มจะใช้ในการเรียนรู้ แล้วใช้ชุดข้อมูลที่เหลือในการตรวจสอบความสามารถในการทำนาย ในการทำนายแต่ละครั้งจะได้จำนวนผลการทำนายเท่ากับจำนวนสมาชิกในชุดข้อมูลที่ใช้ในการทำนายแล้ว ผลการทำนายในแต่ละครั้งจะถูกนับไว้ในตาราง Confusion matrix ซึ่งแนวตั้งจะบอกค่าที่ได้จากการทำนาย แนวนอนจะแสดงค่าความจริง โดยแต่ละชุดข้อมูลจะต้องมีการตรวจสอบโดยการสลับชุดข้อมูลที่ใช้ในการเรียนรู้ และ ทดสอบไปเรื่อยๆ จนครบทั้ง 10 รอบ

จากวิธีการข้างต้นนั้น 7 ชุดข้อมูลจะถูกเรียนรู้ และ ทำนายโดยวิธีการเรียนรู้เครื่อง 7 ชนิด ดังนั้นหลังจากการทดลองจะได้ผลการทดลองทั้งหมด 49 (7X7) ชุด

หลังจากนั้น ทำการเปรียบเทียบความสามารถในการทำนายร่องรอยไม่ดีของของโมเดลการออกแบบซอฟต์แวร์ระหว่างผลการทำนายกับค่าจริง แล้วนำมาคำนวณประสิทธิภาพการทำนายร่องรอยที่ไม่ดี นอกจากนี้เรายังจะเปรียบเทียบความสามารถของวิธีการเรียนรู้เครื่องทั้ง 7 ชนิด ซึ่งมีรายละเอียด และ ผลการประเมินผลในบทถัดไป

บทที่ 4
วิธีการทดลอง ผลการทดลอง และ การประเมินผล

จากขั้นตอนการดำเนินงานที่ได้นำเสนอในบทที่ 3 มาแล้ว ผู้เสนอวิทยานิพนธ์อธิบายวิธีการทดลอง ซึ่งประกอบด้วย กลุ่มตัวอย่างที่ใช้ในการทดลอง เครื่องมือที่ใช้ในการวิจัย ผลการทำนายร่องรอยที่ไม่ดี วิธีการคำนวณประสิทธิภาพการทำนายร่องรอยที่ไม่ดี และ การประเมินผล

4.1 กลุ่มตัวอย่างที่ใช้ในการทดลอง

งานวิจัยนี้ทำการรวบรวมข้อมูลจากชุดซอฟต์แวร์ของงานวิจัยก่อนหน้านี้ [5][6] เพื่อใช้ในการทำนายร่องรอยที่ไม่ดี โปรแกรมชุดที่ 1-19 ซึ่งเป็นโปรแกรมที่ได้จากงานวิจัยที่ [5] จะประกอบด้วย 191 คลาส ซึ่งชุดข้อมูลทั้งหมดจะประกอบด้วยร่องรอยที่ไม่ดี 5 ชนิด คือ Lazy Class, Feature Envy, Long Method, Long Parameter Lists และ Switch Statement

1) โปรแกรมที่ใช้ในการทดสอบโปรแกรมที่ 1 ได้มาจากตัวอย่างโปรแกรมในหนังสือ Refactoring: Improving the design of existing code เขียนโดย Martin Fowler [1] เป็นโปรแกรมของระบบเช่าภาพยนตร์ ที่มี 5 คลาส
2) โปรแกรมที่ใช้ในการทดสอบโปรแกรมที่ 2 ได้มาจากงานวิจัย [5] โดยรีแฟคทอริงโปรแกรมที่ 1 โดยวิธี Move Attribute โดยการย้ายคุณลักษณะ CHILDREN, REGULAR และ NEW_RELEASE จากคลาส Movie ไปยังคลาส Customer ที่มี 5 คลาส
3) โปรแกรมที่ใช้ในการทดสอบโปรแกรมที่ 3 ได้มาจากงานวิจัย [5] โดยรีแฟคทอริงโปรแกรมที่ 1 โดยวิธี Extract Method โดยการย้ายสวิทช์สเตทเมนต์ในเมทธอด statement มาสร้างเป็นเมทธอดใหม่คือ เมทธอด amountFor ที่มี 5 คลาส
4) โปรแกรมที่ใช้ในการทดสอบโปรแกรมที่ 4 ได้มาจากงานวิจัย [5] โดยรีแฟคทอริงโปรแกรมที่ 1 โดยวิธี Extract Method โดยการแยกส่วนของเมทธอด statement ของคลาส Movie และ Rental มาสร้างเป็นเมทธอดใหม่ ที่มี 5 คลาส
5) โปรแกรมที่ใช้ในการทดสอบโปรแกรมที่ 5 ได้มาจากงานวิจัย [5] โดยรีแฟคทอริงโปรแกรมที่ 1 โดยวิธี Replace Temp with Query กับเมทธอด statement ที่มี 5 คลาส
6) โปรแกรมที่ใช้ในการทดสอบโปรแกรมที่ 6 ได้มาจากตัวอย่างโปรแกรมในหนังสือ Refactoring: Improving the design of existing code เขียนโดย Martin Fowler [1] มี 7 คลาส
7) โปรแกรมที่ใช้ในการทดสอบโปรแกรมที่ 7 ได้มาจากงานวิจัย [5] โดยรีแฟคทอริงโปรแกรมที่ 6 โดยวิธี Extract Method กับเมทธอด charge ในคลาส DisabilitySite ที่มี 7 คลาส

8) โปรแกรมที่ใช้ในการทดสอบโปรแกรมที่ 8 ได้มาจากงานวิจัย [5] โดยรีแฟคทอริงโปรแกรมที่ 6 โดยวิธี Extract Method กับเมทธอด charge ในคลาส ResidentialSite ที่มี 7 คลาส
9) โปรแกรมที่ใช้ในการทดสอบโปรแกรมที่ 9 ได้มาจากงานวิจัย [5] โดยรีแฟคทอริงโปรแกรมที่ 6 โดยวิธี Replace Parameters with Method กับเมทธอด charge ในคลาส DisabilitySite ที่มี 7 คลาส
10) โปรแกรมที่ใช้ในการทดสอบโปรแกรมที่ 10 ได้มาจากงานวิจัย [5] โดยรีแฟคทอริงโปรแกรมที่ 6 โดยวิธี Replace Parameters with Method กับเมทธอด charge ในคลาส ResidentialSite ที่มี 7 คลาส
11) โปรแกรมที่ใช้ในการทดสอบโปรแกรมที่ 11 ได้มาจากงานวิจัย [5] โดยรีแฟคทอริงโปรแกรมที่ 6 โดยวิธี Extract Method กับเมทธอด charge ในคลาส BusinessSite ที่มี 7 คลาส
12) โปรแกรมที่ใช้ในการทดสอบโปรแกรมที่ 12 ได้มาจากงานวิจัย [5] โดยรีแฟคทอริงโปรแกรมที่ 6 โดยวิธี Extract Method เพื่อแก้ไขร่องรอยที่ไม่ดี Long Method สำหรับเมทธอด dayOfYear ในคลาส DisabilitySite ที่มี 7 คลาส
13) โปรแกรมที่ใช้ในการทดสอบโปรแกรมที่ 13 ได้มาจากงานวิจัย [5] โดยรีแฟคทอริงโปรแกรมที่ 6 โดยวิธี Extract Method เพื่อแก้ไขร่องรอยที่ไม่ดี Long Method สำหรับเมทธอด dayOfYear ในคลาส ResidentialSite ที่มี 7 คลาส
14) โปรแกรมที่ใช้ในการทดสอบโปรแกรมที่ 14 ได้มาจากงานวิจัย [5] โดยรีแฟคทอริงโปรแกรมที่ 6 โดยวิธี Extract Method เพื่อแก้ไขร่องรอยที่ไม่ดี Switch Statement สำหรับเมทธอด dayOfYear ภายในคลาส DisabilitySite ที่มี 7 คลาส
15) โปรแกรมที่ใช้ในการทดสอบโปรแกรมที่ 15 ได้มาจากงานวิจัย [5] โดยรีแฟคทอริงโปรแกรมที่ 6 โดยวิธี Extract Method เพื่อแก้ไขร่องรอยที่ไม่ดี Switch Statement สำหรับเมทธอด dayOfYear ภายในคลาส ResidentialSite ที่มี 7 คลาส
16) โปรแกรมที่ใช้ในการทดสอบโปรแกรมที่ 16 ได้มาจากตัวอย่างโปรแกรมในหนังสือ Java:How to program [15] เป็นโปรแกรมสำหรับจำลองการทำงานของลิฟต์ มีจำนวน 24 คลาส
17) โปรแกรมที่ใช้ในการทดสอบโปรแกรมที่ 17 ได้มาจากงานวิจัย [5] โดยรีแฟคทอริงโปรแกรมที่ 16 โดยวิธี Extract Class กับคลาส ElevatorSimulation ที่มี 24 คลาส
18) โปรแกรมที่ใช้ในการทดสอบโปรแกรมที่ 18 ได้มาจากงานวิจัย [5] โดยรีแฟคทอริงโปรแกรมที่ 17 โดยวิธี Collapse Hierarchy กับคลาส BellEvent, ButtonEvent, DoorEvent, ElevatorEvent และ LightEvent ที่มี 24 คลาส

19) โปรแกรมที่ใช้ในการทดสอบโปรแกรมที่ 19 ได้มาจากงานวิจัย [5] โดยรีแฟคทอริงโปรแกรมที่ 16 โดยวิธี Collapse Hierarchy กับคลาส BellEvent, ButtonEvent, DoorEvent, ElevatorEvent และ LightEvent ที่มี 24 คลาส

โปรแกรมชุดที่ 20-22 จะประกอบด้วย 27 คลาส เป็นโปรแกรมที่ได้จากงานวิจัยที่ [6] ซึ่งชุดข้อมูลทั้งหมดจะประกอบด้วยร่องรอยที่ไม่ดี 2 ชนิด คือ Middle Man และ Message Chains

20) โปรแกรมที่ใช้ในการทดสอบโปรแกรมที่ 20 ได้มาจากตัวอย่างโปรแกรมในหนังสือ Refactoring: Improving the design of existing code เขียนโดย Martin Fowler [1] เป็นโปรแกรมของระบบเซ่าภาพยนตร์ ที่มี 12 คลาส
21) โปรแกรมที่ใช้ในการทดสอบโปรแกรมที่ 21 ได้มาจากตัวอย่างโปรแกรมในหนังสือ Refactoring Workbook เขียนโดย William C. Wake [16] เป็น โปรแกรมจัดการข้อมูล ที่มี 10 คลาส
22) โปรแกรมที่ใช้ในการทดสอบโปรแกรมที่ 22 ได้มาจากตัวอย่างโปรแกรมในหนังสือ Refactoring Workbook เขียนโดย William C. Wake [16] เป็น โปรแกรมของระบบจำลองการสร้างหุ่นยนต์ ที่มี 5 คลาส

4.2 เครื่องมือที่ใช้ในการวิจัย

1) ฮาร์ดแวร์ (Hardware)

- เครื่องคอมพิวเตอร์ส่วนบุคคลหน่วยประมวลผล อินเทลคอร์ทูดูโอ 2.0 กิกะเฮิร์ต (Intel Core 2 Duo 2.0 GHz.)
- หน่วยความจำ (RAM) 2 กิกะไบต์ (2 GB)
- ฮาร์ดดิสก์ (Hard Disk) 250 กิกะไบต์ (250 GB)

2) ซอฟต์แวร์ (Software) และไลบรารี (Library)

- ระบบปฏิบัติการ วินโดวส์เอ็กซ์พี โปรเฟสชั่นนอล (Windows XP Professional)
- MagicDraw UML เวอร์ชัน 9.5 ในการแปลงรหัสต้นฉบับเป็นแผนภาพคลาส และคำนวณค่ามาตรวัด
- โปรแกรมเวกา (Weka) 3.6 ในการทำนายร่องรอยที่ไม่ดี

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

4.3 ผลการทดลอง

จากการนำข้อมูล 7 ชุด ในขั้นตอนที่ 4.1 มาเรียนรู้กับวิธีการเรียนรู้เครื่อง 7 ชนิด ตามขั้นตอนในบทที่ 3 เราจะได้ผลการทดลองทั้งหมด 49 ชุด ซึ่งผลการทดลองที่ได้จะแสดงเป็นตาราง Confusion matrix ซึ่งแนวตั้งจะบอกค่าที่ได้จากการทำนาย แนวนอนจะแสดงค่าความจริง และข้อมูลในตารางจะบอกจำนวนคลาสที่ได้จากการทำนาย ตารางที่ 4.1 จะแสดงตัวอย่างตารางของผลการทดลอง ซึ่งสามารถอ่านค่าได้ดังนี้

- มี 121 คลาสที่ไม่มีร่องรอยที่ไม่ดี และ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 48 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 1 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ แต่ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 19 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่

ตารางที่ 4.1 ตัวอย่างตารางของผลการทดลอง

		ค่าที่ได้จากการทำนาย					
		0	1	2	3	4	5
ค่าจริง	0	121	48	0	0	0	0
	1	1	19	0	0	0	0
	2	0	0	0	0	0	0
	3	0	0	0	0	0	0
	4	0	0	0	0	0	0
	5	0	0	0	0	0	0

ผลการทดลองได้ถูกจัดเรียงตามชนิดของร่องรอยที่ไม่ดี และ วิธีการเรียนรู้ของเครื่องดังต่อไปนี้

4.3.1. Lazy Class

1) ผลการทำนายร่องรอยไม่ดีประเภท Lazy Class โดย Naive Bayes ในตารางที่ 4.2 สามารถอ่านผลได้ว่า

- มี 121 คลาสที่ไม่มีร่องรอยที่ไม่ดี และ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 48 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 2 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 2 ที่
- มี 1 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ แต่ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 19 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่

ตารางที่ 4.2 ผลการทำนายร่องรอยไม่ดีประเภท Lazy Class โดย Naive Bayes

		ค่าที่ได้จากการทำนาย					
		0	1	2	3	4	5
ค่าจริง	0	121	48	2	0	0	0
	1	1	19	0	0	0	0
	2	0	0	0	0	0	0
	3	0	0	0	0	0	0
	4	0	0	0	0	0	0
	5	0	0	0	0	0	0

2) ผลการทำนายร่องรอยไม่ดีประเภท Lazy Class โดย Logistic ในตารางที่ 4.3 สามารถอ่านผลได้ว่า

- มี 168 คลาสที่ไม่มีร่องรอยที่ไม่ดี และ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 3 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 5 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ แต่ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 15 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่

ตารางที่ 4.3 ผลการทำนายร่องรอยไม่ดีประเภท Lazy Class โดย Logistic

		ค่าที่ได้จากการทำนาย					
		0	1	2	3	4	5
ค่าจริง	0	168	3	0	0	0	0
	1	5	15	0	0	0	0
	2	0	0	0	0	0	0
	3	0	0	0	0	0	0
	4	0	0	0	0	0	0
	5	0	0	0	0	0	0

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

3) ผลการทำนายร่องรอยไม่ดีประเภท Lazy Class โดย IB1 ในตารางที่ 4.4 สามารถอ่านผลได้ว่า

- มี 165 คลาสที่ไม่มีร่องรอยที่ไม่ดี และ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 4 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 4 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ แต่ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 16 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่

ตารางที่ 4.4 ผลการทำนายร่องรอยไม่ดีประเภท Lazy Class โดย IB1

		ค่าที่ได้จากการทำนาย					
		0	1	2	3	4	5
ค่าจริง	0	167	4	0	0	0	0
	1	4	16	0	0	0	0
	2	0	0	0	0	0	0
	3	0	0	0	0	0	0
	4	0	0	0	0	0	0
	5	0	0	0	0	0	0

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

4) ผลการทำนายร่องรอยไม่ดีประเภท Lazy Class โดย IBk ในตารางที่ 4.5 สามารถอ่านผลได้ว่า

- มี 170 คลาสที่ไม่มีร่องรอยที่ไม่ดี และ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 1 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 5 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ แต่ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 15 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่

ตารางที่ 4.5 ผลการทำนายร่องรอยไม่ดีประเภท Lazy Class โดย IBk

		ค่าที่ได้จากการทำนาย					
		0	1	2	3	4	5
ค่าจริง	0	170	1	0	0	0	0
	1	5	15	0	0	0	0
	2	0	0	0	0	0	0
	3	0	0	0	0	0	0
	4	0	0	0	0	0	0
	5	0	0	0	0	0	0

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

5) ผลการทำนายร่องรอยไม่ดีประเภท Lazy Class โดย VFI ในตารางที่ 4.6 สามารถอ่านผลได้ว่า

- มี 115 คลาสที่ไม่มีร่องรอยที่ไม่ดี และ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 56 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 20 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่

ตารางที่ 4.6 ผลการทำนายร่องรอยไม่ดีประเภท Lazy Class โดย VFI

		ค่าที่ได้จากการทำนาย					
		0	1	2	3	4	5
ค่าจริง	0	115	56	0	0	0	0
	1	0	20	0	0	0	0
	2	0	0	0	0	0	0
	3	0	0	0	0	0	0
	4	0	0	0	0	0	0
	5	0	0	0	0	0	0

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

6) ผลการทำนายร่องรอยไม่ดีประเภท Lazy Class โดย J48 ในตารางที่ 4.7 สามารถอ่านผลได้ว่า

- มี 171 คลาสที่ไม่มีร่องรอยที่ไม่ดี และ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 5 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ แต่ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 15 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่

ตารางที่ 4.7 ผลการทำนายร่องรอยไม่ดีประเภท Lazy Class โดย J48

		ค่าที่ได้จากการทำนาย					
		0	1	2	3	4	5
ค่าจริง	0	171	0	0	0	0	0
	1	5	15	0	0	0	0
	2	0	0	0	0	0	0
	3	0	0	0	0	0	0
	4	0	0	0	0	0	0
	5	0	0	0	0	0	0

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

1) ผลการทำนายร่องรอยไม่ดีประเภท Lazy Class โดย Random forest ในตารางที่ 4.8 สามารถอ่านผลได้ว่า

- มี 170 คลาสที่ไม่มีร่องรอยที่ไม่ดี และ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 1 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 5 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ แต่ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 15 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่

ตารางที่ 4.8 ผลการทำนายร่องรอยไม่ดีประเภท Lazy Class โดย Random forest

		ค่าที่ได้จากการทำนาย					
		0	1	2	3	4	5
ค่าจริง	0	170	1	0	0	0	0
	1	5	15	0	0	0	0
	2	0	0	0	0	0	0
	3	0	0	0	0	0	0
	4	0	0	0	0	0	0
	5	0	0	0	0	0	0

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

4.3.2. Feature Envy

1) ผลการทำนายร่องรอยไม่ดีประเภท Feature Envy โดย Naive Bayes ในตารางที่ 4.9 สามารถอ่านผลได้ว่า

- มี 134 คลาสที่ไม่มีร่องรอยที่ไม่ดี และ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 45 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 2 ที่
- มี 1 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 3 ที่
- มี 2 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 4 ที่
- มี 3 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 1 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 1 คลาสที่มีร่องรอยที่ไม่ดี 2 ที่ แต่ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 1 คลาสที่มีร่องรอยที่ไม่ดี 2 ที่ แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 1 คลาสที่มีร่องรอยที่ไม่ดี 2 ที่ แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 3 ที่
- มี 2 คลาสที่มีร่องรอยที่ไม่ดี 3 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 3 ที่

ตารางที่ 4.9 ผลการทำนายร่องรอยไม่ดีประเภท Feature Envy โดย Naive Bayes

		ค่าที่ได้จากการทำนาย					
		0	1	2	3	4	5
ค่าจริง	0	134	0	45	1	2	0
	1	0	3	1	0	0	0
	2	1	1	0	1	0	0
	3	0	0	0	2	0	0
	4	0	0	0	0	0	0
	5	0	0	0	0	0	0

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

2) ผลการทำนายร่องรอยไม่ดีประเภท Feature Envy โดย Logistic ในตารางที่ 4.10 สามารถอ่านผลได้ว่า

- มี 177 คลาสที่ไม่มีร่องรอยที่ไม่ดี และ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 3 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 1 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 2 ที่
- มี 1 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 3 ที่
- มี 1 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ แต่ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 3 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 1 คลาสที่มีร่องรอยที่ไม่ดี 2 ที่ แต่ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 1 คลาสที่มีร่องรอยที่ไม่ดี 2 ที่ แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 1 คลาสที่มีร่องรอยที่ไม่ดี 2 ที่ แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 3 ที่
- มี 2 คลาสที่มีร่องรอยที่ไม่ดี 3 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 3 ที่

ตารางที่ 4.10 ผลการทำนายร่องรอยไม่ดีประเภท Feature Envy โดย Logistic

		ค่าที่ได้จากการทำนาย					
		0	1	2	3	4	5
ค่าจริง	0	177	3	1	1	0	0
	1	1	3	0	0	0	0
	2	1	1	0	1	0	0
	3	0	0	0	2	0	0
	4	0	0	0	0	0	0
	5	0	0	0	0	0	0

3) ผลการทำนายร่องรอยไม่ดีประเภท Feature Envy โดย IB1ในตารางที่ 4.11 สามารถอ่านผลได้ว่า

- มี 179 คลาสที่ไม่มีร่องรอยที่ไม่ดี และ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 1 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 1 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 2 ที่
- มี 1 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 3 ที่
- มี 3 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 1 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 2 ที่
- มี 1 คลาสที่มีร่องรอยที่ไม่ดี 2 ที่ แต่ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 1 คลาสที่มีร่องรอยที่ไม่ดี 2 ที่ แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 1 คลาสที่มีร่องรอยที่ไม่ดี 2 ที่ แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 3 ที่
- มี 2 คลาสที่มีร่องรอยที่ไม่ดี 3 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 3 ที่
-

ตารางที่ 4.11 ผลการทำนายร่องรอยไม่ดีประเภท Feature Envy โดย IB1

		ค่าที่ได้จากการทำนาย					
		0	1	2	3	4	5
ค่าจริง	0	179	1	1	1	0	0
	1	0	3	1	0	0	0
	2	1	1	0	1	0	0
	3	0	0	0	2	0	0
	4	0	0	0	0	0	0
	5	0	0	0	0	0	0

4) ผลการทำนายร่องรอยไม่ดีประเภท Feature Envy โดย IBk ในตารางที่ 4.12 สามารถอ่านผลได้ว่า

- มี 179 คลาสที่ไม่มีร่องรอยที่ไม่ดี และ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 1 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 1 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 2 ที่
- มี 1 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 3 ที่
- มี 2 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 1 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 2 ที่
- มี 1 คลาสที่มีร่องรอยที่ไม่ดี 2 ที่ แต่ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 1 คลาสที่มีร่องรอยที่ไม่ดี 2 ที่ แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 1 คลาสที่มีร่องรอยที่ไม่ดี 2 ที่ แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 3 ที่
- มี 2 คลาสที่มีร่องรอยที่ไม่ดี 3 ที่ แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 2 ที่

ตารางที่ 4.12 ผลการทำนายร่องรอยไม่ดีประเภท Feature Envy โดย IBk

		ค่าที่ได้จากการทำนาย					
		0	1	2	3	4	5
ค่าจริง	0	179	1	1	1	0	0
	1	0	2	1	0	0	0
	2	1	1	0	1	0	0
	3	0	0	2	0	0	0
	4	0	0	0	0	0	0
	5	0	0	0	0	0	0

5) ผลการทำนายร่องรอยไม่ดีประเภท Feature Envy โดย VFI ในตารางที่ 4.13 สามารถอ่านผลได้ว่า

- มี 133 คลาสที่ไม่มีร่องรอยที่ไม่ดี และ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 25 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 13 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 2 ที่
- มี 11 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 3 ที่
- มี 3 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 1 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 2 ที่
- มี 1 คลาสที่มีร่องรอยที่ไม่ดี 2 ที่ แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 2 คลาสที่มีร่องรอยที่ไม่ดี 2 ที่ แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 3 ที่
- มี 2 คลาสที่มีร่องรอยที่ไม่ดี 3 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 3 ที่

ตารางที่ 4.13 ผลการทำนายร่องรอยไม่ดีประเภท Feature Envy โดย VFI

		ค่าที่ได้จากการทำนาย					
		0	1	2	3	4	5
ค่าจริง	0	133	25	13	11	0	0
	1	0	3	1	0	0	0
	2	0	1	0	2	0	0
	3	0	0	0	2	0	0
	4	0	0	0	0	0	0
	5	0	0	0	0	0	0

6) ผลการทำนายร่องรอยไม่ดีประเภท Feature Envy โดย J48 ในตารางที่ 4.14 สามารถอ่านผลได้ว่า

- มี 180 คลาสที่ไม่มีร่องรอยที่ไม่ดี และ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 2 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 4 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ แต่ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 3 คลาสที่มีร่องรอยที่ไม่ดี 2 ที่ แต่ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 2 คลาสที่มีร่องรอยที่ไม่ดี 3 ที่ แต่ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี

ตารางที่ 4.14 ผลการทำนายร่องรอยไม่ดีประเภท Feature Envy โดย J48

		ค่าที่ได้จากการทำนาย					
		0	1	2	3	4	5
ค่าจริง	0	180	2	0	0	0	0
	1	4	0	0	0	0	0
	2	3	0	0	0	0	0
	3	2	0	0	0	0	0
	4	0	0	0	0	0	0
	5	0	0	0	0	0	0

7) ผลการทำนายร่องรอยไม่ดีประเภท Feature Envy โดย Random forest ในตารางที่ 4.15 สามารถอ่านผลได้ว่า

- มี 181 คลาสที่ไม่มีร่องรอยที่ไม่ดี และ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 1 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 2 ที่
- มี 3 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 1 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 2 ที่
- มี 1 คลาสที่มีร่องรอยที่ไม่ดี 2 ที่ แต่ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 1 คลาสที่มีร่องรอยที่ไม่ดี 2 ที่ แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 1 คลาสที่มีร่องรอยที่ไม่ดี 2 ที่ แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 3 ที่
- มี 1 คลาสที่มีร่องรอยที่ไม่ดี 3 ที่ แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 2 ที่
- มี 1 คลาสที่มีร่องรอยที่ไม่ดี 3 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 3 ที่

ตารางที่ 4.15 ผลการทำนายร่องรอยไม่ดีประเภท Feature Envy โดย Random forest

		ค่าที่ได้จากการทำนาย					
		0	1	2	3	4	5
ค่าจริง	0	181	0	1	0	0	0
	1	0	3	1	0	0	0
	2	1	1	0	1	0	0
	3	0	0	1	1	0	0
	4	0	0	0	0	0	0
	5	0	0	0	0	0	0

4.3.3. Middle Man

1) ผลการทำนายร่องรอยไม่ดีประเภท Middle Man โดย Naive Bayes ในตารางที่ 4.16 สามารถอ่านผลได้ว่า

- มี 7 คลาสที่ไม่มีร่องรอยที่ไม่ดี และ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 3 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 2 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 2 ที่
- มี 2 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ แต่ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 4 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 2 คลาสที่มีร่องรอยที่ไม่ดี 2 ที่ แต่ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 4 คลาสที่มีร่องรอยที่ไม่ดี 2 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 2 ที่
- มี 1 คลาสที่มีร่องรอยที่ไม่ดี 3 ที่ แต่ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 1 คลาสที่มีร่องรอยที่ไม่ดี 3 ที่ แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 1 คลาสที่มีร่องรอยที่ไม่ดี 3 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 3 ที่

ตารางที่ 4.16 ผลการทำนายร่องรอยไม่ดีประเภท Middle Man โดย Naive Bayes

		ค่าที่ได้จากการทำนาย					
		0	1	2	3	4	5
ค่าจริง	0	7	3	2	0	0	0
	1	2	4	0	0	0	0
	2	2	0	4	0	0	0
	3	1	1	0	1	0	0
	4	0	0	0	0	0	0
	5	0	0	0	0	0	0

2) ผลการทำนายร่องรอยไม่ดีประเภท Middle Man โดย Logistic ในตารางที่ 4.17 สามารถอ่านผลได้ว่า

- มี 10 คลาสที่ไม่มีร่องรอยที่ไม่ดี และ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 1 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 1 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 3 ที่
- มี 6 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 6 คลาสที่มีร่องรอยที่ไม่ดี 2 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 2 ที่
- มี 1 คลาสที่มีร่องรอยที่ไม่ดี 3 ที่ แต่ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 2 คลาสที่มีร่องรอยที่ไม่ดี 3 ที่ แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่

ตารางที่ 4.17 ผลการทำนายร่องรอยไม่ดีประเภท Middle Man โดย Logistic

		ค่าที่ได้จากการทำนาย					
		0	1	2	3	4	5
ค่าจริง	0	10	1	0	1	0	0
	1	0	6	0	0	0	0
	2	0	0	6	0	0	0
	3	1	2	0	0	0	0
	4	0	0	0	0	0	0
	5	0	0	0	0	0	0

3) ผลการทำนายร่องรอยไม่ดีประเภท Middle Man โดย IB1ในตารางที่ 4.18 สามารถอ่านผลได้ว่า

- มี 10 คลาสที่ไม่มีร่องรอยที่ไม่ดี และ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 1 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 1 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 3 ที่
- มี 6 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 6 คลาสที่มีร่องรอยที่ไม่ดี 2 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 2 ที่
- มี 1 คลาสที่มีร่องรอยที่ไม่ดี 3 ที่ แต่ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 1 คลาสที่มีร่องรอยที่ไม่ดี 3 ที่ แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 1 คลาสที่มีร่องรอยที่ไม่ดี 3 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 3 ที่

ตารางที่ 4.18 ผลการทำนายร่องรอยไม่ดีประเภท Middle Man โดย IB1

		ค่าที่ได้จากการทำนาย					
		0	1	2	3	4	5
ค่าจริง	0	10	1	0	1	0	0
	1	0	6	0	0	0	0
	2	0	0	6	0	0	0
	3	1	1	0	1	0	0
	4	0	0	0	0	0	0
	5	0	0	0	0	0	0

4) ผลการทำนายร่องรอยไม่ดีประเภท Middle Man โดย IBk ในตารางที่ 4.19 สามารถอ่านผลได้ว่า

- มี 10 คลาสที่ไม่มีร่องรอยที่ไม่ดี และ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 1 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 1 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 3 ที่
- มี 6 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 6 คลาสที่มีร่องรอยที่ไม่ดี 2 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 2 ที่
- มี 1 คลาสที่มีร่องรอยที่ไม่ดี 3 ที่ แต่ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 1 คลาสที่มีร่องรอยที่ไม่ดี 3 ที่ แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 1 คลาสที่มีร่องรอยที่ไม่ดี 3 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 3 ที่

ตารางที่ 4.19 ผลการทำนายร่องรอยไม่ดีประเภท Middle Man โดย IBk

		ค่าที่ได้จากการทำนาย					
		0	1	2	3	4	5
ค่าจริง	0	10	1	0	1	0	0
	1	0	6	0	0	0	0
	2	0	0	6	0	0	0
	3	1	1	0	1	0	0
	4	0	0	0	0	0	0
	5	0	0	0	0	0	0

5) ผลการทำนายร่องรอยไม่ดีประเภท Middle Man โดย VFI ในตารางที่ 4.20 สามารถอ่านผลได้ว่า

- มี 1 คลาสที่ไม่มีร่องรอยที่ไม่ดี และ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 7 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 3 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 2 ที่
- มี 1 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 3 ที่
- มี 6 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 2 คลาสที่มีร่องรอยที่ไม่ดี 2 ที่ แต่ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 4 คลาสที่มีร่องรอยที่ไม่ดี 2 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 2 ที่
- มี 1 คลาสที่มีร่องรอยที่ไม่ดี 3 ที่ แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 2 คลาสที่มีร่องรอยที่ไม่ดี 3 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 3 ที่

ตารางที่ 4.20 ผลการทำนายร่องรอยไม่ดีประเภท Middle Man โดย VFI

		ค่าที่ได้จากการทำนาย					
		0	1	2	3	4	5
ค่าจริง	0	1	7	3	1	0	0
	1	0	6	0	0	0	0
	2	2	0	4	0	0	0
	3	0	1	0	2	0	0
	4	0	0	0	0	0	0
	5	0	0	0	0	0	0

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

6) ผลการทำนายร่องรอยไม่ดีประเภท Middle Man โดย J48 ในตารางที่ 4.21 สามารถอ่านผลได้ว่า

- มี 11 คลาสที่ไม่มีร่องรอยที่ไม่ดี และ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 1 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 2 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ แต่ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 4 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 4 คลาสที่มีร่องรอยที่ไม่ดี 2 ที่ แต่ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 2 คลาสที่มีร่องรอยที่ไม่ดี 2 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 2 ที่
- มี 2 คลาสที่มีร่องรอยที่ไม่ดี 3 ที่ แต่ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 1 คลาสที่มีร่องรอยที่ไม่ดี 3 ที่ แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่

ตารางที่ 4.21 ผลการทำนายร่องรอยไม่ดีประเภท Middle Man โดย J48

		ค่าที่ได้จากการทำนาย					
		0	1	2	3	4	5
ค่าจริง	0	11	1	0	0	0	0
	1	2	4	0	0	0	0
	2	4	0	2	0	0	0
	3	2	1	0	0	0	0
	4	0	0	0	0	0	0
	5	0	0	0	0	0	0

7) ผลการทำนายร่องรอยไม่ดีประเภท Middle Man โดย Random forest ในตารางที่ 4.22 สามารถอ่านผลได้ว่า

- มี 10 คลาสที่ไม่มีร่องรอยที่ไม่ดี และ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 1 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 1 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 3 ที่
- มี 1 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ แต่ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 5 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 6 คลาสที่มีร่องรอยที่ไม่ดี 2 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 2 ที่
- มี 1 คลาสที่มีร่องรอยที่ไม่ดี 3 ที่ แต่ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 1 คลาสที่มีร่องรอยที่ไม่ดี 3 ที่ แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 1 คลาสที่มีร่องรอยที่ไม่ดี 3 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 3 ที่

ตารางที่ 4.22 ผลการทำนายร่องรอยไม่ดีประเภท Middle Man โดย Random forest

		ค่าที่ได้จากการทำนาย					
		0	1	2	3	4	5
ค่าจริง	0	10	1	0	1	0	0
	1	1	5	0	0	0	0
	2	0	0	6	0	0	0
	3	1	1	0	1	0	0
	4	0	0	0	0	0	0
	5	0	0	0	0	0	0

4.3.4. Message Chains

1) ผลการทำนายร่องรอยไม่ดีประเภท Message Chains โดย Naive Bayes ในตารางที่ 4.23 สามารถอ่านผลได้ว่า

- มี 12 คลาสที่ไม่มีร่องรอยที่ไม่ดี และ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 3 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 2 ที่
- มี 2 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ แต่ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 4 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 2 คลาสที่มีร่องรอยที่ไม่ดี 2 ที่ แต่ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 4 คลาสที่มีร่องรอยที่ไม่ดี 2 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 2 ที่

ตารางที่ 4.23 ผลการทำนายร่องรอยไม่ดีประเภท Message Chains โดย Naive Bayes

		ค่าที่ได้จากการทำนาย					
		0	1	2	3	4	5
ค่าจริง	0	12	0	3	0	0	0
	1	2	4	0	0	0	0
	2	2	0	4	0	0	0
	3	0	0	0	0	0	0
	4	0	0	0	0	0	0
	5	0	0	0	0	0	0

2) ผลการทำนายร่องรอยไม่ดีประเภท Message Chains โดย Logistic ในตารางที่ 4.24 สามารถอ่านผลได้ว่า

- มี 13 คลาสที่ไม่มีร่องรอยที่ไม่ดี และ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 1 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 1 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 2 ที่
- มี 6 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 6 คลาสที่มีร่องรอยที่ไม่ดี 2 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 2 ที่

ตารางที่ 4.24 ผลการทำนายร่องรอยไม่ดีประเภท Message Chains โดย Logistic

		ค่าที่ได้จากการทำนาย					
		0	1	2	3	4	5
ค่าจริง	0	13	1	1	0	0	0
	1	0	6	0	0	0	0
	2	0	0	6	0	0	0
	3	0	0	0	0	0	0
	4	0	0	0	0	0	0
	5	0	0	0	0	0	0

3) ผลการทำนายร่องรอยไม่ดีประเภท Message Chains โดย IB1 ในตารางที่ 4.25 สามารถอ่านผลได้ว่า

- มี 13 คลาสที่ไม่มีร่องรอยที่ไม่ดี และ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 1 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 1 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 2 ที่
- มี 6 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 6 คลาสที่มีร่องรอยที่ไม่ดี 2 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 2 ที่

ตารางที่ 4.25 ผลการทำนายร่องรอยไม่ดีประเภท Message Chains โดย IB1

		ค่าที่ได้จากการทำนาย					
		0	1	2	3	4	5
ค่าจริง	0	13	1	1	0	0	0
	1	0	6	0	0	0	0
	2	0	0	6	0	0	0
	3	0	0	0	0	0	0
	4	0	0	0	0	0	0
	5	0	0	0	0	0	0

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

4) ผลการทำนายร่องรอยไม่ดีประเภท Message Chains โดย IBk ในตารางที่ 4.26 สามารถอ่านผลได้ว่า

- มี 13 คลาสที่ไม่มีร่องรอยที่ไม่ดี และ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 1 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 1 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 2 ที่
- มี 6 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 6 คลาสที่มีร่องรอยที่ไม่ดี 2 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 2 ที่

ตารางที่ 4.26 ผลการทำนายร่องรอยไม่ดีประเภท Message Chains โดย IBk

		ค่าที่ได้จากการทำนาย					
		0	1	2	3	4	5
ค่าจริง	0	13	1	1	0	0	0
	1	0	6	0	0	0	0
	2	0	0	6	0	0	0
	3	0	0	0	0	0	0
	4	0	0	0	0	0	0
	5	0	0	0	0	0	0

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

5) ผลการทำนายร่องรอยไม่ดีประเภท Message Chains โดย VFI ในตารางที่ 4.27 สามารถอ่านผลได้ว่า

- มี 10 คลาสที่ไม่มีร่องรอยที่ไม่ดี และ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 5 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 2 ที่
- มี 2 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ แต่ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 4 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 6 คลาสที่มีร่องรอยที่ไม่ดี 2 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 2 ที่

ตารางที่ 4.27 ผลการทำนายร่องรอยไม่ดีประเภท Message Chains โดย VFI

		ค่าที่ได้จากการทำนาย					
		0	1	2	3	4	5
ค่าจริง	0	10	0	5	0	0	0
	1	2	4	0	0	0	0
	2	0	0	6	0	0	0
	3	0	0	0	0	0	0
	4	0	0	0	0	0	0
	5	0	0	0	0	0	0

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

6) ผลการทำนายร่องรอยไม่ดีประเภท Message Chains โดย J48 ในตารางที่ 4.28 สามารถอ่านผลได้ว่า

- มี 11 คลาสที่ไม่มีร่องรอยที่ไม่ดี และ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 1 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 3 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 2 ที่
- มี 6 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 6 คลาสที่มีร่องรอยที่ไม่ดี 2 ที่ แต่ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี

ตารางที่ 4.28 ผลการทำนายร่องรอยไม่ดีประเภท Message Chains โดย J48

		ค่าที่ได้จากการทำนาย					
		0	1	2	3	4	5
ค่าจริง	0	11	1	3	0	0	0
	1	0	6	0	0	0	0
	2	6	0	0	0	0	0
	3	0	0	0	0	0	0
	4	0	0	0	0	0	0
	5	0	0	0	0	0	0

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

7) ผลการทำนายร่องรอยไม่ดีประเภท Message Chains โดย Random forest ในตารางที่ 4.29 สามารถอ่านผลได้ว่า

- มี 13 คลาสที่ไม่มีร่องรอยที่ไม่ดี และ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 2 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 6 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 6 คลาสที่มีร่องรอยที่ไม่ดี 2 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 2 ที่

ตารางที่ 4.29 ผลการทำนายร่องรอยไม่ดีประเภท Message Chains โดย Random forest

		ค่าที่ได้จากการทำนาย					
		0	1	2	3	4	5
ค่าจริง	0	13	2	0	0	0	0
	1	0	6	0	0	0	0
	2	0	0	6	0	0	0
	3	0	0	0	0	0	0
	4	0	0	0	0	0	0
	5	0	0	0	0	0	0

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

4.3.5. Long Method

1) ผลการทำนายร่องรอยไม่ดีประเภท Long Method โดย Naive Bayes ในตารางที่ 4.30 สามารถอ่านผลได้ว่า

- มี 142 คลาสที่ไม่มีร่องรอยที่ไม่ดี และ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 25 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 2 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 3 ที่
- มี 19 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 1 คลาสที่มีร่องรอยที่ไม่ดี 2 ที่ แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 2 คลาสที่มีร่องรอยที่ไม่ดี 2 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 2 ที่

ตารางที่ 4.30 ผลการทำนายร่องรอยไม่ดีประเภท Long Method โดย Naive Bayes

		ค่าที่ได้จากการทำนาย					
		0	1	2	3	4	5
ค่าจริง	0	142	25	0	2	0	0
	1	0	19	0	0	0	0
	2	0	1	2	0	0	0
	3	0	0	0	0	0	0
	4	0	0	0	0	0	0
	5	0	0	0	0	0	0

2) ผลการทำนายร่องรอยไม่ดีประเภท Long Method โดย Logistic ในตารางที่ 4.31 สามารถอ่านผลได้ว่า

- มี 163 คลาสที่ไม่มีร่องรอยที่ไม่ดี และ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 25 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 1 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 2 ที่
- มี 5 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ แต่ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 13 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 1 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 2 ที่
- มี 1 คลาสที่มีร่องรอยที่ไม่ดี 2 ที่ แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 2 คลาสที่มีร่องรอยที่ไม่ดี 2 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 2 ที่

ตารางที่ 4.31 ผลการทำนายร่องรอยไม่ดีประเภท Long Method โดย Logistic

		ค่าที่ได้จากการทำนาย					
		0	1	2	3	4	5
ค่าจริง	0	163	5	1	0	0	0
	1	5	13	1	0	0	0
	2	0	1	2	0	0	0
	3	0	0	0	0	0	0
	4	0	0	0	0	0	0
	5	0	0	0	0	0	0

3) ผลการทำนายร่องรอยไม่ดีประเภท Long Method โดย IB1ในตารางที่ 4.32 สามารถอ่านผลได้ว่า

- มี 165 คลาสที่ไม่มีร่องรอยที่ไม่ดี และ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 4 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 3 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ แต่ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 16 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 1 คลาสที่มีร่องรอยที่ไม่ดี 2 ที่ แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 2 คลาสที่มีร่องรอยที่ไม่ดี 2 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 2 ที่

ตารางที่ 4.32 ผลการทำนายร่องรอยไม่ดีประเภท Long Method โดย IB1

		ค่าที่ได้จากการทำนาย					
		0	1	2	3	4	5
ค่าจริง	0	165	4	0	0	0	0
	1	3	16	0	0	0	0
	2	0	1	2	0	0	0
	3	0	0	0	0	0	0
	4	0	0	0	0	0	0
	5	0	0	0	0	0	0

4) ผลการทำนายร่องรอยไม่ดีประเภท Long Method โดย IBk ในตารางที่ 4.33 สามารถอ่านผลได้ว่า

- มี 165 คลาสที่ไม่มีร่องรอยที่ไม่ดี และ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 4 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 2 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ แต่ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 17 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 1 คลาสที่มีร่องรอยที่ไม่ดี 2 ที่ แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 2 คลาสที่มีร่องรอยที่ไม่ดี 2 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 2 ที่

ตารางที่ 4.33 ผลการทำนายร่องรอยไม่ดีประเภท Long Method โดย IBk

		ค่าที่ได้จากการทำนาย					
		0	1	2	3	4	5
ค่าจริง	0	165	4	0	0	0	0
	1	2	17	0	0	0	0
	2	0	1	2	0	0	0
	3	0	0	0	0	0	0
	4	0	0	0	0	0	0
	5	0	0	0	0	0	0

5) ผลการทำนายร่องรอยไม่ดีประเภท Long Method โดย VFI ในตารางที่ 4.34 สามารถอ่านผลได้ว่า

- มี 139 คลาสที่ไม่มีร่องรอยที่ไม่ดี และ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 29 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 1 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 2 ที่
- มี 2 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ แต่ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 17 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 1 คลาสที่มีร่องรอยที่ไม่ดี 2 ที่ แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 2 คลาสที่มีร่องรอยที่ไม่ดี 2 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 2 ที่

ตารางที่ 4.34 ผลการทำนายร่องรอยไม่ดีประเภท Long Method โดย VFI

		ค่าที่ได้จากการทำนาย					
		0	1	2	3	4	5
ค่าจริง	0	139	29	1	0	0	0
	1	2	17	0	0	0	0
	2	0	1	2	0	0	0
	3	0	0	0	0	0	0
	4	0	0	0	0	0	0
	5	0	0	0	0	0	0

6) ผลการทำนายร่องรอยไม่ดีประเภท Long Method โดย J48 ในตารางที่ 4.35 สามารถอ่านผลได้ว่า

- มี 164 คลาสที่ไม่มีร่องรอยที่ไม่ดี และ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 5 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 5 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ แต่ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 13 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 1 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 2 ที่
- มี 1 คลาสที่มีร่องรอยที่ไม่ดี 2 ที่ แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 2 คลาสที่มีร่องรอยที่ไม่ดี 2 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 2 ที่

ตารางที่ 4.35 ผลการทำนายร่องรอยไม่ดีประเภท Long Method โดย J48

		ค่าที่ได้จากการทำนาย					
		0	1	2	3	4	5
ค่าจริง	0	164	5	0	0	0	0
	1	5	13	1	0	0	0
	2	0	1	2	0	0	0
	3	0	0	0	0	0	0
	4	0	0	0	0	0	0
	5	0	0	0	0	0	0

7) ผลการทำนายร่องรอยไม่ดีประเภท Long Method โดย Random forest ในตารางที่ 4.36 สามารถอ่านผลได้ว่า

- มี 163 คลาสที่ไม่มีร่องรอยที่ไม่ดี และ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 6 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 2 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ แต่ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 17 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 1 คลาสที่มีร่องรอยที่ไม่ดี 2 ที่ แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 2 คลาสที่มีร่องรอยที่ไม่ดี 2 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 2 ที่

ตารางที่ 4.36 ผลการทำนายร่องรอยไม่ดีประเภท Long Method โดย Random forest

		ค่าที่ได้จากการทำนาย					
		0	1	2	3	4	5
ค่าจริง	0	163	6	0	0	0	0
	1	2	17	0	0	0	0
	2	0	1	2	0	0	0
	3	0	0	0	0	0	0
	4	0	0	0	0	0	0
	5	0	0	0	0	0	0

4.3.6. Long Parameter Lists

1) ผลการทำนายร่องรอยไม่ดีประเภท Long Parameter Lists โดย Naive Bayes ในตารางที่ 4.37 สามารถอ่านผลได้ว่า

- มี 99 คลาสที่ไม่มีร่องรอยที่ไม่ดี และ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 68 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 2 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 2 ที่
- มี 22 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่

ตารางที่ 4.37 ผลการทำนายร่องรอยไม่ดีประเภท Long Parameter Lists โดย Naive Bayes

		ค่าที่ได้จากการทำนาย					
		0	1	2	3	4	5
ค่าจริง	0	99	68	2	0	0	0
	1	0	22	0	0	0	0
	2	0	0	0	0	0	0
	3	0	0	0	0	0	0
	4	0	0	0	0	0	0
	5	0	0	0	0	0	0

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

2) ผลการทำนายร่องรอยไม่ดีประเภท Long Parameter Lists โดย Logistic ในตารางที่ 4.38 สามารถอ่านผลได้ว่า

- มี 167 คลาสที่ไม่มีร่องรอยที่ไม่ดี และ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 1 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 1 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 5 ที่
- มี 11 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ แต่ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 11 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่

ตารางที่ 4.38 ผลการทำนายร่องรอยไม่ดีประเภท Long Parameter Lists โดย Logistic

		ค่าที่ได้จากการทำนาย					
		0	1	2	3	4	5
ค่าจริง	0	167	1	0	0	0	1
	1	11	11	0	0	0	0
	2	0	0	0	0	0	0
	3	0	0	0	0	0	0
	4	0	0	0	0	0	0
	5	0	0	0	0	0	0

3) ผลการทำนายร่องรอยไม่ดีประเภท Long Parameter Lists โดย IB1 ในตารางที่ 4.39 สามารถอ่านผลได้ว่า

- มี 166 คลาสที่ไม่มีร่องรอยที่ไม่ดี และ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 3 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 2 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ แต่ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 20 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่

ตารางที่ 4.39 ผลการทำนายร่องรอยไม่ดีประเภท Long Parameter Lists โดย IB1

		ค่าที่ได้จากการทำนาย					
		0	1	2	3	4	5
ค่าจริง	0	166	3	0	0	0	0
	1	2	20	0	0	0	0
	2	0	0	0	0	0	0
	3	0	0	0	0	0	0
	4	0	0	0	0	0	0
	5	0	0	0	0	0	0

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

4) ผลการทำนายร่องรอยไม่ดีประเภท Long Parameter Lists โดย IBk ในตารางที่ 4.40 สามารถอ่านผลได้ว่า

- มี 166 คลาสที่ไม่มีร่องรอยที่ไม่ดี และ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 3 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 2 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ แต่ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 20 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่

ตารางที่ 4.40 ผลการทำนายร่องรอยไม่ดีประเภท Long Parameter Lists โดย IBk

		ค่าที่ได้จากการทำนาย					
		0	1	2	3	4	5
ค่าจริง	0	166	3	0	0	0	0
	1	2	20	0	0	0	0
	2	0	0	0	0	0	0
	3	0	0	0	0	0	0
	4	0	0	0	0	0	0
	5	0	0	0	0	0	0

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

5) ผลการทำนายร่องรอยไม่ดีประเภท Long Parameter Lists โดย VFI ในตารางที่ 4.41 สามารถอ่านผลได้ว่า

- มี 92 คลาสที่ไม่มีร่องรอยที่ไม่ดี และ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 77 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 1 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ แต่ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 21 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่

ตารางที่ 4.41 ผลการทำนายร่องรอยไม่ดีประเภท Long Parameter Lists โดย VFI

		ค่าที่ได้จากการทำนาย					
		0	1	2	3	4	5
ค่าจริง	0	92	77	0	0	0	0
	1	1	21	0	0	0	0
	2	0	0	0	0	0	0
	3	0	0	0	0	0	0
	4	0	0	0	0	0	0
	5	0	0	0	0	0	0

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

6) ผลการทำนายร่องรอยไม่ดีประเภท Long Parameter Lists โดย J48 ในตารางที่ 4.42 สามารถอ่านผลได้ว่า

- มี 166 คลาสที่ไม่มีร่องรอยที่ไม่ดี และ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 3 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 3 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ แต่ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 19 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่

ตารางที่ 4.42 ผลการทำนายร่องรอยไม่ดีประเภท Long Parameter Lists โดย J48

		ค่าที่ได้จากการทำนาย					
		0	1	2	3	4	5
ค่าจริง	0	166	3	0	0	0	0
	1	3	19	0	0	0	0
	2	0	0	0	0	0	0
	3	0	0	0	0	0	0
	4	0	0	0	0	0	0
	5	0	0	0	0	0	0

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

7) ผลการทำนายร่องรอยไม่ดีประเภท Long Parameter Lists โดย Random forest ในตารางที่ 4.43 สามารถอ่านผลได้ว่า

- มี 168 คลาสที่ไม่มีร่องรอยที่ไม่ดี และ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 1 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 1 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ แต่ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 21 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่

ตารางที่ 4.43 ผลการทำนายร่องรอยไม่ดีประเภท Long Parameter Lists โดย Random forest

		ค่าที่ได้จากการทำนาย					
		0	1	2	3	4	5
ค่าจริง	0	168	1	0	0	0	0
	1	1	21	0	0	0	0
	2	0	0	0	0	0	0
	3	0	0	0	0	0	0
	4	0	0	0	0	0	0
	5	0	0	0	0	0	0

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

4.3.7. Switch Statement

1) ผลการทำนายร่องรอยไม่ดีประเภท Switch Statement โดย Naive Bayes ในตารางที่ 4.44 สามารถอ่านผลได้ว่า

- มี 158 คลาสที่ไม่มีร่องรอยที่ไม่ดี และ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 12 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 2 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 2 ที่
- มี 1 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ แต่ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 18 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่

ตารางที่ 4.44 ผลการทำนายร่องรอยไม่ดีประเภท Switch Statement โดย Naive Bayes

		ค่าที่ได้จากการทำนาย					
		0	1	2	3	4	5
ค่าจริง	0	158	12	2	0	0	0
	1	1	18	0	0	0	0
	2	0	0	0	0	0	0
	3	0	0	0	0	0	0
	4	0	0	0	0	0	0
	5	0	0	0	0	0	0

2) ผลการทำนายร่องรอยไม่ดีประเภท Switch Statement โดย Logistic ในตารางที่ 4.45 สามารถอ่านผลได้ว่า

- มี 168 คลาสที่ไม่มีร่องรอยที่ไม่ดี และ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 4 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 2 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ แต่ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 17 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่

ตารางที่ 4.45 ผลการทำนายร่องรอยไม่ดีประเภท Switch Statement โดย Logistic

		ค่าที่ได้จากการทำนาย					
		0	1	2	3	4	5
ค่าจริง	0	168	4	0	0	0	0
	1	2	17	0	0	0	0
	2	0	0	0	0	0	0
	3	0	0	0	0	0	0
	4	0	0	0	0	0	0
	5	0	0	0	0	0	0

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

3) ผลการทำนายร่องรอยไม่ดีประเภท Switch Statement โดย IB1 ในตารางที่ 4.46 สามารถอ่านผลได้ว่า

- มี 168 คลาสที่ไม่มีร่องรอยที่ไม่ดี และ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 4 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 2 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ แต่ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 17 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่

ตารางที่ 4.46 ผลการทำนายร่องรอยไม่ดีประเภท Switch Statement โดย IB1

		ค่าที่ได้จากการทำนาย					
		0	1	2	3	4	5
ค่าจริง	0	168	4	0	0	0	0
	1	2	17	0	0	0	0
	2	0	0	0	0	0	0
	3	0	0	0	0	0	0
	4	0	0	0	0	0	0
	5	0	0	0	0	0	0

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

4) ผลการทำนายร่องรอยไม่ดีประเภท Switch Statement โดย IBk ในตารางที่ 4.47 สามารถอ่านผลได้ว่า

- มี 169 คลาสที่ไม่มีร่องรอยที่ไม่ดี และ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 3 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 2 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ แต่ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 17 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่

ตารางที่ 4.47 ผลการทำนายร่องรอยไม่ดีประเภท Switch Statement โดย IBk

		ค่าที่ได้จากการทำนาย					
		0	1	2	3	4	5
ค่าจริง	0	169	3	0	0	0	0
	1	2	17	0	0	0	0
	2	0	0	0	0	0	0
	3	0	0	0	0	0	0
	4	0	0	0	0	0	0
	5	0	0	0	0	0	0

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

5) ผลการทำนายร่องรอยไม่ดีประเภท Switch Statement โดย VFI ในตารางที่ 4.48 สามารถอ่านผลได้ว่า

- มี 143 คลาสที่ไม่มีร่องรอยที่ไม่ดี และ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 29 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 1 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ แต่ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 18 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่

ตารางที่ 4.48 ผลการทำนายร่องรอยไม่ดีประเภท Switch Statement โดย VFI

		ค่าที่ได้จากการทำนาย					
		0	1	2	3	4	5
ค่าจริง	0	143	29	0	0	0	0
	1	1	18	0	0	0	0
	2	0	0	0	0	0	0
	3	0	0	0	0	0	0
	4	0	0	0	0	0	0
	5	0	0	0	0	0	0

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

6) ผลการทำนายร่องรอยไม่ดีประเภท Switch Statement โดย J48 ในตารางที่ 4.49 สามารถอ่านผลได้ว่า

- มี 171 คลาสที่ไม่มีร่องรอยที่ไม่ดี และ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 1 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 2 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ แต่ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 17 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่

ตารางที่ 4.49 ผลการทำนายร่องรอยไม่ดีประเภท Switch Statement โดย J48

		ค่าที่ได้จากการทำนาย					
		0	1	2	3	4	5
ค่าจริง	0	171	1	0	0	0	0
	1	2	17	0	0	0	0
	2	0	0	0	0	0	0
	3	0	0	0	0	0	0
	4	0	0	0	0	0	0
	5	0	0	0	0	0	0

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

7) ผลการทำนายร่องรอยไม่ดีประเภท Switch Statement โดย Random forest ในตารางที่ 4.50 สามารถอ่านผลได้ว่า

- มี 168 คลาสที่ไม่มีร่องรอยที่ไม่ดี และ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 4 คลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่
- มี 2 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ แต่ ถูกทำนายให้เป็นคลาสที่ไม่มีร่องรอยไม่ดี
- มี 17 คลาสที่มีร่องรอยที่ไม่ดี 1 ที่ และ ถูกทำนายให้เป็นคลาสที่มีร่องรอยไม่ดี 1 ที่

ตารางที่ 4.50 ผลการทำนายร่องรอยไม่ดีประเภท Switch Statement โดย Random forest

		ค่าที่ได้จากการทำนาย					
		0	1	2	3	4	5
ค่าจริง	0	168	4	0	0	0	0
	1	2	17	0	0	0	0
	2	0	0	0	0	0	0
	3	0	0	0	0	0	0
	4	0	0	0	0	0	0
	5	0	0	0	0	0	0

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

4.4 การคำนวณประสิทธิภาพการทำนายร่องรอยที่ไม่ดี

จากผลการทดลอง ผู้วิจัยได้นำผลการทดลองทั้ง 49 ชุด มาคำนวณประสิทธิภาพการทำนายร่องรอยที่ไม่ดีด้วย ความถูกต้องการทำนายของการทำนายร่องรอยที่ไม่ดี (Bad-smell prediction accuracy) ความไว และ ความจำเพาะ (Sensitivity & Specificity) และ ค่าพยากรณ์ของการทดสอบ (Predictive value of tests) วิธีการวิเคาระห์ผลลัพธ์การทดลอง และผลการคำนวณประสิทธิภาพการทำนายนั้น แสดงอยู่ในหัวข้อถัดไป

4.4.1. การวิเคราะห์ผลลัพธ์การทดลอง

จากผลการทดลองสามารถอ่านผลลัพธ์ได้สองแบบ สำหรับการคำนวณประสิทธิภาพการทำนายร่องรอยที่ไม่ดี ซึ่งได้อธิบายไว้แล้วใน บทที่ 2 ประกอบด้วยวิธีการอ่านสองแบบ คือ

1. ความถูกต้องในการทำนายร่องรอยที่ไม่ดี ในตารางที่ 4.51 แสดงตัวอย่างวิธีการวิเคราะห์ผลลัพธ์ โดยแบ่งผลลัพธ์เป็นสองพวก
   - ผลรวมของช่องสีเขียว คือ จำนวนคลาสที่สามารถทำนายถูกต้อง
   - ผลรวมของช่องสีฟ้า คือ จำนวนคลาสที่ทำนายผิดพลาด
   - ผลรวมของทุกช่อง คือ จำนวนคลาสทั้งหมด

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ 4.51 แสดงตัวอย่างวิธีการวิเคราะห์ผลลัพธ์ โดยแบ่งผลลัพธ์เป็นสองพวก จากผลการทำนายร่องรอยไม่ดีประเภท Lazy Class โดย Naive Bayes

		ค่าที่ได้จากการทำนาย					
		0	1	2	3	4	5
ค่าจริง	0	121	48	2	0	0	0
	1	1	19	0	0	0	0
	2	0	0	0	0	0	0
	3	0	0	0	0	0	0
	4	0	0	0	0	0	0
	5	0	0	0	0	0	0

2. ความไว ความจำเพาะ และ ค่าพยากรณ์ของการทดสอบ ในตารางที่ 4.52 แสดงตัวอย่างวิธีการวิเคราะห์ผลลัพธ์ โดยแบ่งผลลัพธ์เป็นสี่พวก

   - ผลรวมของช่องสีฟ้า คือ True positive (TP) หมายความว่าคลาสที่มีร่องรอยที่ไม่ดี และ ถูกทำนายว่าเป็นคลาสที่มีร่องรอยไม่ดีด้วย

   - ผลรวมของช่องสีส้ม คือ False positive (FP) หมายความว่าคลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ถูกทำนายว่าเป็นคลาสที่มีร่องรอยไม่ดี

   - ผลรวมของช่องสีเขียว คือ True negative (TN) หมายความว่าคลาสที่ไม่มีร่องรอยที่ไม่ดี และ ถูกทำนายว่าเป็นคลาสที่ไม่มีร่องรอยที่ไม่ดี

   - ผลรวมของช่องสีน้ำตาล คือ False negative (FN) หมายความว่าคลาสที่มีร่องรอยที่ไม่ดี แต่ถูกทำนายว่าเป็นคลาสที่ไม่มีร่องรอยที่ไม่ดี

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ 4.52 แสดงตัวอย่างวิธีการวิเคราะห์ผลลัพธ์ โดยแบ่งผลลัพธ์เป็นสี่พวก จากผลการทำนายร่องรอยไม่ดีประเภท Lazy Class โดย Naive Bayes

		ค่าที่ได้จากการทำนาย					
		0	1	2	3	4	5
ค่าจริง	0	121	48	2	0	0	0
	1	1	19	0	0	0	0
	2	0	0	0	0	0	0
	3	0	0	0	0	0	0
	4	0	0	0	0	0	0
	5	0	0	0	0	0	0

ผลการคำนวณประสิทธิภาพการทำนายร่องรอยที่ไม่ดี ได้ถูกจัดเรียงตามวิธีการคำนวณประสิทธิภาพ ดังต่อไปนี้

4.4.2. ความถูกต้องในทำนายร่องรอยที่ไม่ดี (Bad-smell prediction accuracy)

ผู้วิจัยได้ทำการคำนวณประสิทธิภาพการทำนายร่องรอยที่ไม่ดี โดยวิธีทดสอบความถูกต้องในทำนายร่องรอยที่ไม่ดีกับผลการทดลองทั้ง 49 ชุด ผู้วิจัยได้นำผลการทำนายร่องรอยที่ไม่ดีใน ตารางที่ 4.2 - 4.50 มาวิเคราะห์ผลลัพธ์การทดลองตามหัวข้อที่ 4.4.1. แล้วนำผลการวิเคราะห์มาคำนวณค่าความถูกต้องในทำนายร่องรอยที่ไม่ดี ตามสูตรที่ได้อธิบายไว้ในหัวข้อที่ 2.1.5.1

ตัวอย่าง การคำนวณค่าความถูกต้องในทำนายร่องรอยที่ไม่ดี จากผลการทำนายร่องรอยไม่ดีประเภท Lazy Class โดย Naive Bayes จาก ตารางที่ 4.51 สามารถวิเคราะห์ผลลัพธ์การทดลองได้ว่า

- มีจำนวนคลาสที่สามารถทำนายถูกต้อง 140 คลาส
- มีจำนวนคลาสทั้งหมด 191 คลาส

จากสูตรความถูกต้องในการทำนายร่องรอยที่ไม่ดี สามารถคำนวณค่าความถูกต้องในการทำนายร่องรอยที่ไม่ดีได้ 73.30%

ความถูกต้องในการทำนายร่องรอยที่ไม่ดี = จำนวนคลาสที่สามารถทำนายถูกต้อง / จำนวนคลาสทั้งหมด

ความถูกต้องในการทำนายร่องรอยที่ไม่ดี = 140 / 191 = 0.7330 = 73.30%

ตารางที่ 4.53 แสดงความถูกต้องในทำนายร่องรอยที่ไม่ดีในรูปแบบตารางความสัมพันธ์ระหว่างร่อยรอยที่ไม่ดี และ วิธีการเรียนรู้เครื่อง

ตารางที่ 4.53 ความถูกต้องในทำนายร่องรอยที่ไม่ดี (Bad-smell prediction accuracy)

	Feature Envy (%)	Long Method (%)	Long Parameter Lists (%)	Lazy Class (%)	Message Chains (%)	Middle Man (%)	Switch Statement (%)	Average (%)
Naive Bayes	72.77	85.34	73.30	73.30	74.07	59.26	92.15	78.38
Logistic regression	95.29	93.19	95.81	95.81	92.59	81.48	96.86	93.81
IB1	96.34	95.81	95.81	95.81	92.59	85.19	96.86	94.74
IBk	95.29	96.34	96.86	96.86	92.59	85.19	97.38	95.00
VFI	72.25	82.72	70.68	70.68	74.07	48.15	84.29	72.65
J48	94.24	93.72	97.38	97.38	62.96	62.96	98.43	88.32
Random Forest	96.86	95.29	96.86	96.86	92.59	81.48	96.86	94.53

4.4.3. ความไวและความจำเพาะ (Sensitivity&Specificity)

ในหัวข้อนี้ผู้วิจัยได้ทำการคำนวณประสิทธิภาพการทำนายร่องรอยที่ไม่ดี โดยวิธีทดสอบความไว และ ความจำเพาะกับผลการทดลองทั้ง 49 ชุด ผู้วิจัยได้นำผลการทำนายร่องรอยที่ไม่ดีในตารางที่ 4.2 - 4.50 มาวิเคราะห์ผลลัพธ์การทดลองตามหัวข้อที่ 4.4.1. แล้วนำผลการวิเคราะห์มาคำนวณค่าความไว และ ความจำเพาะ ตามสูตรที่ได้อธิบายไว้ในหัวข้อที่ 2.1.5.2 ตารางที่ 4.54-

4.55 แสดงความไว และ ความจำเพาะในการทำนายในการทำนายร่องรอยที่ไม่ดีในรูปแบบตาราง ความสัมพันธ์ระหว่างร่อยรอยที่ไม่ดี และ วิธีการเรียนรู้เครื่อง

ตัวอย่าง การคำนวณค่าความไว จากผลการทำนายร่องรอยไม่ดีประเภท Lazy Class โดย Naive Bayes จาก ตารางที่ 4.52 สามารถวิเคราะห์ผลลัพธ์การทดลองได้ว่า

- มีคลาสที่มีร่องรอยที่ไม่ดี และ ถูกทำนายว่าเป็นคลาสที่มีร่องรอยไม่ดีด้วย (TP) 19 คลาส
- มีคลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ถูกทำนายว่าเป็นคลาสที่มีร่องรอยไม่ดี (FP) 50 คลาส
- มีคลาสที่ไม่มีร่องรอยที่ไม่ดี และ ถูกทำนายว่าเป็นคลาสที่ไม่มีร่องรอยที่ไม่ดี (TN) 121 คลาส
- มีคลาสที่มีร่องรอยที่ไม่ดี แต่ถูกทำนายว่าเป็นคลาสที่ไม่มีร่องรอยที่ไม่ดี (FN) 1 คลาส

จากสูตรการคำนวณค่าความไว สามารถคำนวณค่าความไวในการทำนายร่องรอยที่ไม่ดี ได้ 70.46%

$$\text{Sensitivity} = \frac{\text{TN}}{\text{TN} + \text{FP}} = \frac{121}{121 + 50} = 0.7076 = 70.76\%$$

ตารางที่ 4.54 แสดงความไวในการทำนายร่องรอยที่ไม่ดี ของวิธีการเรียนรู้เครื่องประเภท Logistic regression, IB1, IBk, J48 และ Random Forest ที่มีความสามารถในการทำนายมากกว่า Naive Bayes และ VFI เพราะวิธีการเรียนรู้เครื่องกลุ่มนี้มีความสามารถในการทำนายโดยเฉลี่ยมากกว่า 90%

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ 4.54 ความไวในการทำนายในการทำนายร่องรอยที่ไม่ดี

	Feature Envy (%)	Long Method (%)	Long Parameter Lists (%)	Lazy Class (%)	Message Chains (%)	Middle Man (%)	Switch Statement (%)	Average (%)
Naive Bayes	73.63	84.02	58.58	70.76	80.00	58.33	91.86	73.88
Logistic regression	97.25	96.45	98.82	98.25	86.67	83.33	97.67	94.06
IB1	98.35	97.63	98.22	97.66	86.67	83.33	97.67	94.22
IBk	98.35	97.63	98.22	99.42	86.67	83.33	98.26	94.55
VFI	73.08	82.25	54.44	67.25	66.67	8.33	83.14	62.17
J48	98.90	97.04	98.22	100.00	73.33	91.67	99.42	94.08
Random Forest	99.45	96.45	99.41	99.42	86.67	83.33	97.67	94.63

ตัวอย่าง การคำนวณค่าความจำเพาะ จากผลการทำนายร่องรอยไม่ดีประเภท Lazy Class โดย Naive Bayes จาก ตารางที่ 4.52 สามารถวิเคราะห์ผลลัพธ์การทดลองได้ว่า

- มีคลาสที่มีร่องรอยที่ไม่ดี และ ถูกทำนายว่าเป็นคลาสที่มีร่องรอยไม่ดีด้วย (TP) 19 คลาส
- มีคลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ถูกทำนายว่าเป็นคลาสที่มีร่องรอยไม่ดี (FP) 50 คลาส
- มีคลาสที่ไม่มีร่องรอยที่ไม่ดี และ ถูกทำนายว่าเป็นคลาสที่ไม่มีร่องรอยที่ไม่ดี (TN) 121 คลาส
- มีคลาสที่มีร่องรอยที่ไม่ดี แต่ถูกทำนายว่าเป็นคลาสที่ไม่มีร่องรอยที่ไม่ดี (FN) 1 คลาส

จากสูตรการคำนวณค่าความจำเพาะ สามารถคำนวณค่าความจำเพาะในการทำนายร่องรอยที่ไม่ดีได้ 95.00%

$$\text{Specificity} = \frac{TP}{TP + FN} = \frac{19}{19 + 1} = 0.95 = 95.00\%$$

ตารางที่ 4.55 แสดงความจำเพาะในการทำนายของการทำนายร่องรอยที่ไม่ดี ของวิธีการเรียนรู้เครื่องประเภท IB1, IBk และ VFI ที่มีความสามารถในการทำนายมากกว่าวิธีการเรียนรู้เครื่องอื่นๆ เพราะวิธีการเรียนรู้เครื่องกลุ่มนี้มีความสามารถในการทำนายโดยเฉลี่ยใกล้ 90%

ตารางที่ 4.55 ความจำเพาะในการทำนายร่องรอยที่ไม่ดี

	Feature Envy (%)	Long Method (%)	Long Parameter Lists (%)	Lazy Class (%)	Message Chains (%)	Middle Man (%)	Switch Statement (%)	Average (%)
Naive Bayes	88.89	100.00	100.00	95.00	66.67	66.67	94.74	87.42
Logistic regression	77.78	77.27	50.00	75.00	100.00	93.33	89.47	80.41
IB1	88.89	86.36	90.91	80.00	100.00	93.33	89.47	89.85
IBk	87.5	90.91	90.91	75.00	100.00	93.33	89.47	89.59
VFI	100.00	90.91	95.45	100.00	83.33	86.67	94.74	93.01
J48	0.00	77.27	86.36	75.00	50.00	46.67	89.47	60.68
Random Forest	88.89	90.91	95.45	75.00	100.00	86.67	89.47	89.48

4.4.4. ค่าพยากรณ์ของการทดสอบ (Predictive value of tests)

ผู้วิจัยได้ทำการคำนวณประสิทธิภาพการทำนายร่องรอยที่ไม่ดี โดยวิธีการคำนวณค่าพยากรณ์ของการทดสอบกับผลการทดลองทั้ง 49 ชุด ผู้วิจัยได้นำผลการทำนายร่องรอยที่ไม่ดีในตารางที่ 4.2 - 4.50 มาวิเคราะห์ผลลัพธ์การทดลองตามหัวข้อที่ 4.4.1. แล้วนำผลการวิเคราะห์มาคำนวณค่าพยากรณ์ของการทดสอบ ตามสูตรที่ได้อธิบายไว้ในหัวข้อที่ 2.1.5.3

ตัวอย่าง การคำนวณค่าพยากรณ์ของการทดสอบทางบวก จากผลการทำนายร่องรอยไม่ดีประเภท Lazy Class โดย Naive Bayes จาก ตารางที่ 4.52 สามารถวิเคราะห์ผลลัพธ์การทดลองได้ว่า

- มีคลาสที่มีร่องรอยที่ไม่ดี และ ถูกทำนายว่าเป็นคลาสที่มีร่องรอยไม่ดีด้วย (TP) 19 คลาส
- มีคลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ถูกทำนายว่าเป็นคลาสที่มีร่องรอยไม่ดี (FP) 50 คลาส
- มีคลาสที่ไม่มีร่องรอยที่ไม่ดี และ ถูกทำนายว่าเป็นคลาสที่ไม่มีร่องรอยที่ไม่ดี (TN) 121 คลาส

- มีคลาสที่มีร่องรอยที่ไม่ดี แต่ถูกทำนายว่าเป็นคลาสที่ไม่มีร่องรอยที่ไม่ดี (FN) 1 คลาส

จากสูตรการคำนวณค่าพยากรณ์ของการทดสอบทางบวก สามารถคำนวณค่าพยากรณ์ของการทดสอบทางบวกได้ 95.00%

$$PPV = \frac{TP}{TP + FP} = \frac{19}{19 + 50} = 0.2754 = 27.54\%$$

ตารางที่ 4.56 แสดงค่าพยากรณ์ของการทดสอบทางบวกในการทำนายร่องรอยที่ไม่ดี ของวิธีการเรียนรู้เครื่องประเภท Naive Bayes, Logistic regression, IB1, IBk และ Random Forest ที่มีความสามารถในการทำนายมากกว่า VFI และ J48 เพราะวิธีการเรียนรู้เครื่องกลุ่มนี้มีความสามารถในการทำนายโดยเฉลี่ยมากกว่า 90%

ตารางที่ 4.56 ค่าพยากรณ์ของการทดสอบทางบวกในการทำนายร่องรอยที่ไม่ดี

	Feature Envy (%)	Long Method (%)	Long Parameter Lists (%)	Lazy Class (%)	Message Chains (%)	Middle Man (%)	Switch Statement (%)	Average (%)
Naive Bayes	14.29	44.9	23.91	27.54	72.73	66.67	56.25	43.76
Logistic regression	58.33	73.91	84.62	83.33	85.71	87.5	80.95	79.19
IB1	72.73	82.61	86.96	80.00	85.71	87.5	80.95	82.35
IBk	70.00	83.33	86.96	93.75	85.71	87.5	85.00	84.61
VFI	15.52	40.00	21.43	26.32	66.67	54.17	38.30	37.49
J48	0.00	77.27	86.36	100.00	60.00	87.5	94.44	72.22
Random Forest	88.89	76.92	95.45	93.75	85.71	86.67	80.95	86.91

ตัวอย่าง การคำนวณค่าพยากรณ์ของการทดสอบทางลบ จากผลการทำนายร่องรอยไม่ดีประเภท Lazy Class โดย Naive Bayes จาก ตารางที่ 4.52 สามารถวิเคราะห์ผลลัพธ์การทดลองได้ว่า

- มีคลาสที่มีร่องรอยที่ไม่ดี และ ถูกทำนายว่าเป็นคลาสที่มีร่องรอยไม่ดีด้วย (TP) 19 คลาส

- มีคลาสที่ไม่มีร่องรอยที่ไม่ดี แต่ถูกทำนายว่าเป็นคลาสที่มีร่องรอยไม่ดี (FP) 50 คลาส
- มีคลาสที่ไม่มีร่องรอยที่ไม่ดี และ ถูกทำนายว่าเป็นคลาสที่ไม่มีร่องรอยที่ไม่ดี (TN) 121 คลาส
- มีคลาสที่มีร่องรอยที่ไม่ดี แต่ถูกทำนายว่าเป็นคลาสที่ไม่มีร่องรอยที่ไม่ดี (FN) 1 คลาส

จากสูตรการคำนวณค่าพยากรณ์ของการทดสอบทางลบ สามารถคำนวณค่าพยากรณ์ของการทดสอบทางลบได้ 99.18%

$$NPV = \frac{TN}{TN + FN} = \frac{121}{121 + 1} = 0.9918 = 99.18\%$$

ตารางที่ 4.57 แสดงค่าพยากรณ์ของการทดสอบทางลบในการทำนายร่องรอยที่ไม่ดี ของวิธีการเรียนรู้เครื่องประเภท Logistic regression, IB1, IBk และ Random Forest ที่มีความสามารถในการทำนายมากกว่าวิธีการเรียนรู้เครื่องอื่นๆ โดยมีความสามารถในการทำนายโดยเฉลี่ยมากกว่า 80%

ตารางที่ 4.57 ค่าพยากรณ์ของการทดสอบทางลบในการทำนายร่องรอยที่ไม่ดี

	Feature Envy (%)	Long Method (%)	Long Parameter Lists (%)	Lazy Class (%)	Message Chains (%)	Middle Man (%)	Switch Statement (%)	Average (%)
Naive Bayes	99.26	100.00	100.00	99.18	75.00	58.33	99.37	90.16
Logistic regression	98.88	97.02	93.82	97.11	100.00	90.91	98.82	96.65
IB1	99.44	98.21	98.81	97.66	100.00	90.91	98.82	97.69
IBk	99.44	98.8	98.81	97.14	100.00	90.91	98.83	97.70
VFI	100.00	98.58	98.92	100.00	83.33	33.33	99.31	87.64
J48	95.24	97.04	98.22	97.16	64.71	57.89	98.84	87.01
Random Forest	99.45	98.79	99.41	97.14	100.00	83.33	98.82	96.71

4.5 การเปรียบเทียบประสิทธิภาพของวิธีการเรียนรู้เครื่อง

จากตารางที่ 4.53 ผู้วิจัยได้ทำการเปรียบเทียบประสิทธิภาพของวิธีการเรียนรู้เครื่องในการทำนายร่องรอยที่ไม่ดี โดยการวิเคราะห์ผลแบบ Pair Sample T-test ตามวิธีการที่ได้อธิบายไว้ในหัวข้อที่ 2.1.6 ผู้วิจัยทำการจับคู่เปรียบเทียบวิธีการเรียนรู้เครื่องทุกวิธี เพราะฉนั้น จะต้องมีการเปรียบเทียบทั้งหมด 21 รอบ

ตัวอย่าง1 การเปรียบเทียบประสิทธิภาพของวิธีการเรียนรู้เครื่องประเภท Naive Bayes และ VFI ในการทำนายร่องรอยไม่ดี 7 ชนิด ที่ระดับนัยสำคัญ 0.05 โดยมีสมมติฐานดังต่อไปนี้

H0: ประสิทธิภาพของวิธีการเรียนรู้เครื่องประเภท Naive Bayes และ VFI ไม่แตกต่างกัน

H1: ประสิทธิภาพของวิธีการเรียนรู้เครื่องประเภท Naive Bayes มากกว่า VFI

ตารางที่ 4.58 ความถูกต้องในทำนายร่องรอยที่ไม่ดีของวิธีการเรียนรู้เครื่องประเภท Naive Bayes และ VFI

	Naive Bayes	VFI	D	D^2
Feature Envy (%)	72.77	72.25	0.52	0.27
Long Method (%)	85.34	82.72	2.62	6.86
Long Parameter Lists (%)	73.3	70.68	2.62	6.86
Lazy Class (%)	73.3	70.68	2.62	6.86
Message Chains (%)	74.07	74.07	0	0
Middle Man (%)	59.26	48.15	11.11	123.43
Switch Statement (%)	92.15	84.29	7.86	61.77
Sum	530.19	502.84	27.35	206.07

จากสูตรการคำนวณ Pair Sample T-test สามารถคำนวณค่า T-test ได้ 2.5421

$$t = \frac{\sum D}{\sqrt{\frac{N\sum D^2 - (\sum D)^2}{(N-1)}}}$$

$$t = \frac{27.35}{\sqrt{\frac{(7x206.07) - (27.35x27.35)}{(7-1)}}}$$

$$t = 2.5421$$

จากการเปิดตาราง DOF (Degree of freedom) สำหรับตัวแปร 7 ตัว และ ที่ระดับนัยสำคัญ 0.05 จะได้ค่า t วิกฤตเท่ากับ 1.9431

จากการเปรียบเทียบ t ที่ได้จากการคำนวณ และ ค่า t วิกฤต แสดงให้เห็นว่าค่า t ที่ได้จากการคำนวณ มากกว่าค่า t วิกฤต ซึ่งแสดงว่า มีการปฏิเสธสมมติฐานฐานหลัก หรือ ยอมรับสมมติฐานรอง คือ ประสิทธิภาพของวิธีการเรียนรู้เครื่องประเภท Naive Bayes มากกว่า VFI

ตัวอย่าง2 การเปรียบเทียบประสิทธิภาพของวิธีการเรียนรู้เครื่องประเภท IB1 และ Random Forest ในการทำนายร่องรอยไม่ดี 7 ชนิด ที่ระดับนัยสำคัญ 0.05 โดยมีสมมติฐานดังต่อไปนี้

H0: ประสิทธิภาพของวิธีการเรียนรู้เครื่องประเภท IB1 และ Random Forest ไม่แตกต่างกัน

H1: ประสิทธิภาพของวิธีการเรียนรู้เครื่องประเภท IB1 มากกว่า Random Forest

ตารางที่ 4.59 ความถูกต้องในทำนายร่องรอยที่ไม่ดีของวิธีการเรียนรู้เครื่องประเภท IB1 และ Random Forest

	IB1	Random Forest	D	D^2
Feature Envy (%)	96.34	96.86	-0.52	0.27
Long Method (%)	95.81	95.29	0.52	0.27
Long Parameter Lists (%)	95.81	96.86	-1.05	1.10
Lazy Class (%)	95.81	96.86	-1.05	1.10
Message Chains (%)	92.59	92.59	0	0
Middle Man (%)	85.19	81.48	3.71	13.76
Switch Statement (%)	96.86	96.86	0	0
Sum	658.41	656.8	1.61	16.50

จากสูตรการคำนวณ Pair Sample T-test สามารถคำนวณค่า T-test ได้ 0.3710

$$t = \frac{\sum D}{\sqrt{\frac{N\sum D^2 - (\sum D)^2}{(N-1)}}}$$

$$t = \frac{1.60}{\sqrt{\frac{(7x16.50) - (1.61x1.61)}{(7-1)}}}$$

$$t = 0.3710$$

จากการเปิดตาราง DOF (Degree of freedom) สำหรับตัวแปร 7 ตัว และ ที่ระดับนัยสำคัญ 0.05 จะได้ค่า t วิกฤตเท่ากับ 1.9431

จากการเปรียบเทียบ t ที่ได้จากการคำนวณ และ ค่า t วิกฤต แสดงให้เห็นว่าค่า t ที่ได้จากการคำนวณ น้อยกว่าค่า t วิกฤต ซึ่งแสดงว่า ยอมรับสมมติฐานหลัก คือ ประสิทธิภาพของวิธีการเรียนรู้เครื่องประเภท IB1 และ Random Forest ไม่แตกต่าง

เมื่อเราทำการเปรียบเทียบวิธีการเรียนรู้เครื่องทุกวิธีเป็นคู่ๆ จะได้ผลลัพธ์ตามตารางที่ 4.60 แสดงค่า t ที่ได้จากการเปรียบเทียบวิธีการเรียนรู้เครื่องทุกวิธี

ตารางที่ 4.60 ค่า t ที่ได้จากการเปรียบเทียบวิธีการเรียนรู้เครื่องทุกวิธี

	IBk	IB1	Random Forest	Logistic regression	J48	Naive Bayes	VFI
IBk		1.082	0.8686	2.4045	1.5855	6.4052	7.1057
IB1			0.3710	1.8293	1.4943	6.1925	6.9438
Random Forest				2.5690	1.5521	6.1920	7.5116
Logistic regression					1.3179	5.9375	7.0548
J48						2.2253	3.0199
Naive Bayes							2.5421
VFI							

จากการเปิดตาราง DOF (Degree of freedom) สำหรับตัวแปร 7 ตัว และ ที่ระดับนัยสำคัญ 0.05 จะได้ค่า t วิกฤตเท่ากับ 1.9431

จากการเปรียบเทียบค่า t วิกฤต ด้วยค่า t ที่ได้จากการคำนวณในตารางที่ 4.60 ในทุกคู่ จะได้ผลลัพธ์ตามตารางที่ 4.61 ซึ่งเป็นตารางความสัมพันธ์ของวิธีการเรียนรู้เครื่องแนวนอน เมื่อเทียบกับวิธีการเรียนรู้เครื่องแนวตั้ง เพื่อที่จะยอมรับสมมติฐานหลัก หรือ ยอมรับสมมติฐานรอง

ตารางที่ 4.61 ตารางการยอมรับสมมติฐานหลัก หรือ ยอมรับสมมติฐานรอง ตามวิธีการเรียนรู้เครื่อง

	IBk	IB1	Random Forest	Logistic regression	J48	Naive Bayes	VFI
IBk		=	=	>	=	>	>
IB1			=	=	=	>	>
Random Forest				>	=	>	>
Logistic regression					=	>	>
J48						>	>
Naive Bayes							>
VFI							

เมื่อ = คือ ยอมรับสมมติฐานหลัก แปลว่า ประสิทธิภาพของวิธีการเรียนรู้เครื่องแนวนอน ไม่แตกต่างวิธีการเรียนรู้เครื่องแนวตั้ง

> คือ มีการปฏิเสธสมมติฐานฐานหลัก หรือ ยอมรับสมมติฐานรอง แปลว่า ประสิทธิภาพของวิธีการเรียนรู้เครื่องแนวนอน มากกว่าวิธีการเรียนรู้เครื่องแนวตั้ง

4.6 การประเมินผล

เป้าหมายของงานวิจัยนี้ คือ การทำนายร่องรอยที่ไม่ดีของซอฟต์แวร์ตั้งแต่การออกแบบในกระบวนการผลิตซอฟต์แวร์ หัวข้อนี้จะอภิปรายความสามารถของวิธีทำนายร่องรอยที่ไม่ดีที่ได้นำเสนอ และ อภิปรายวิธีการเรียนรู้เครื่องที่เหมาะสมกับร่องรอยที่ไม่ดีเพื่อนำไปใช้ใช้งาน โดยได้เสนอเป็น 3 หัวข้อใหญ่ๆ คือ การเลือกวิธีการเรียนรู้เครื่องที่เหมาะสมกับร่องรอยที่ไม่ดีแบบทั่วไป การเลือกวิธีการเรียนรู้เครื่องที่เหมาะสมตามความร้ายแรงของร่องรอยที่ไม่ดี และ การเลือกวิธีการเรียนรู้เครื่องที่เหมาะสมกับร่องรอยที่ไม่ดีทุกประเภท

4.6.1. การเลือกวิธีการเรียนรู้เครื่องที่เหมาะสมกับร่องรอยที่ไม่ดีแบบทั่วไป

จากตารางที่ 4.53 - 4.57 แสดงค่าความไว ความจำเพาะ และ ค่าพยากรณ์ของการทดสอบ ซึ่งสรุปได้ว่าไม่มีวิธีการเรียนรู้เครื่องที่มีความสามารถดีที่สุดสำหรับทุกร่องรอยที่ไม่ดี ดังนั้นเราจึงแสดงวิธีการเรียนรู้เครื่องที่เหมาะสมในการตรวจหาร่องรอยที่ไม่ดี ตามวิธีการคำนวณประสิทธิภาพการทำนายร่องรอยที่ไม่ดี ตารางที่ 4.62

ตารางที่ 4.62 วิธีการเรียนรู้เครื่องที่เหมาะสมในการทำนายร่องรอยที่ไม่ดีตามวิธีการคำนวณประสิทธิภาพการทำนายร่องรอยที่ไม่ดี

	Prediction accuracy	Specificity	sensitivity	PPV	NPV
Feature Envy	Random Forest	VFI	Random Forest	Random Forest	VFI
Long Method	IBk	Naive Bayes	IB1 and IBk	IBk	Naive Bayes
Long Parameter Lists	J48	Naive Bayes	Logistic regression and Random Forest	Random Forest	Naive Bayes
Lazy Class	J48	VFI	J48	J48	VFI
Message Chains	Logistic regression, IB1, IBk and Random Forest	Logistic regression, IB1 and IBk	Logistic regression, IB1, IBk and Random Forest	Logistic regression, IB1 and IBk	Logistic regression, IB1 and IBk
Middle Man	IB1 and IBk	Logistic regression, IB1 and IBk	J48	J48	Logistic regression, IB1 and IBk
Switch Statement	IBk	Naive Bayes and VFI	J48	J48	Naive Bayes

1) ความถูกต้องในทำนายร่องรอยที่ไม่ดี (Bad-smell prediction accuracy)

จากการคำนวณประสิทธิภาพการทำนายร่องรอยที่ไม่ดี โดยวัดผลความถูกต้องในทำนายร่องรอยที่ไม่ดี แสดงให้เห็นว่า Naive Bayes, VFI และ J48 ไม่เหมาะสมสำหรับการทำนายร่องรอยที่ไม่ดี เพราะว่าวิธีการเรียนรู้เครื่องกลุ่มนี้มีความสามารถในการทำนายต่ำ ถึงแม้ว่าวิธีการ

เรียนรู้เครื่องกลุ่มนี้มีความสามารถในการทำนายโดยเฉลี่ยสูงกว่า 90% ในการทำนายร่องรอยที่ไม่ดีประเภท Long Method แต่วิธีการเรียนรู้เครื่องกลุ่มนี้มีความสามารถในการทำนายต่ำในการทำนายร่องรอยที่ไม่ดีประเภท Middle Man และ วิธีการเรียนรู้เครื่องกลุ่มนี้มีความสามารถในการทำนายโดยเฉลี่ยต่ำ เมื่อเปรียบเทียบกับวิธีการเรียนรู้เครื่องประเภท Random Forests วิธีการเรียนรู้เครื่องประเภท Random Forests มีความสามารถในการทำนายสูงเช่นเดียวกับ Logistic regression, IB1 และ IBk เพราะว่ากลุ่มนี้มีความสามารถในการทำนายสูงเท่า ๆ กัน

2) ความไว แล ะความจำเพาะ (Sensitivity & Specificity)

สำหรับซอร์ฟแวร์คาดว่าจะมีจำนวนร่องรอยที่ไม่ดีน้อยๆ เราควรเลือกวิธีการเรียนรู้เครื่องที่มีค่าความไวสูง ๆ ในตารางที่ 4.54 แสดงค่าความไวของทุกวิธีการเรียนรู้เครื่องตามร่องรอยที่ไม่ดี เราสามารถสรุปได้ว่า

- Random Forest เหมาะสำหรับการทำนายร่องรอยไม่ดีประเภท Feature Envy, Long Parameter Lists และ Message Chains

- Naive Bayes เหมาะสำหรับการทำนายร่องรอยไม่ดีประเภท Lazy Class, Message Chains และ Switch Statement

- IB1 และ IBk เหมาะสำหรับการทำนายร่องรอยไม่ดีประเภท Long Method และ Message Chains

- Logistic regression เหมาะสำหรับการทำนายร่องรอยไม่ดีประเภท Long Parameter Lists และ Message Chains

ในทางกลับกัน สำหรับซอร์ฟแวร์ที่คาดว่าจะมีจำนวนร่องรอยที่ไม่ดีสูง เราควรเลือกวิธีการเรียนรู้เครื่องที่มีค่าความจำเพาะสูง ๆ ในตารางที่ 4.55 แสดงค่าความจำเพาะของทุกวิธีการเรียนรู้เครื่องตามร่องรอยที่ไม่ดี เราสามารถสรุปได้ว่า

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

- Naive Bayes เหมาะสำหรับการทำนายร่องรอยไม่ดีประเภท Long Method, Long Parameter และ Switch Statement

- Logistic regression, IB1 and IBk เหมาะสำหรับการทำนายร่องรอยไม่ดีประเภท Message Chains และ Middle Man

- VFI เหมาะสำหรับการทำนายร่องรอยไม่ดีประเภท Feature Envy and Lazy Class และ Switch Statement

3) ค่าพยากรณ์ของการทดสอบ (Predictive value of tests)

สำหรับซอร์ฟแวร์ที่ต้องการความถูกต้องในการทำนายร่องรอยที่ไม่ดี เราควรเลือกวิธีการเรียนรู้เครื่องที่มีค่าพยากรณ์ของการทดสอบทางบวกสูงๆ ในตารางที่ 4.56 แสดงค่าพยากรณ์ของการทดสอบทางบวกของทุกวิธีการเรียนรู้เครื่องตามร่องรอยที่ไม่ดี เราสามารถสรุปได้ว่า

- Random Forest เหมาะสำหรับการทำนายร่องรอยไม่ดีประเภท Feature Envy และ Long Parameter Lists

- J48 เหมาะสำหรับการทำนายร่องรอยไม่ดีประเภท Lazy Class, Middle Man และ Switch Statement

- IBk เหมาะสำหรับการทำนายร่องรอยไม่ดีประเภท Long Method และ Message Chains

- Logistic regression และ IB1 เหมาะสำหรับการทำนายร่องรอยไม่ดีประเภท Message Chains

ในทางกลับกัน สำหรับซอร์ฟแวร์ที่ต้องการความถูกต้องในการทำนาย เพื่อป้องกันการทำนายคลาสที่ดีเป็นคลาสที่มีร่องรอยไม่ดี เราควรเลือกวิธีการเรียนรู้เครื่องที่มีค่าพยากรณ์ของการ

ทดสอบทางลบสูง ๆ ในตารางที่ 4.57 แสดงค่าพยากรณ์ของการทดสอบทางลบของทุกวิธีการเรียนรู้เครื่องตามร่องรอยที่ไม่ดี เราสามารถสรุปได้ว่า

- Naïve bayes เหมาะสำหรับการทำนายร่องรอยไม่ดีประเภท Long Method, Long Parameter Lists และ Switch Statement

- VFI เหมาะสำหรับการทำนายร่องรอยไม่ดีประเภท Feature Envy และ Lazy Class

- Logistic regression, IB1 และ IBk เหมาะสำหรับการทำนายร่องรอยไม่ดีประเภท Message Chains และ Middle Man

4.6.2. การเลือกวิธีการเรียนรู้เครื่องที่เหมาะสมตามความร้ายแรงของร่องรอยที่ไม่ดี

จากร่องรอยที่ไม่ดีทั้ง 7 ชนิดเราสามารถแบ่งได้ 2 กลุ่ม คือ ร่องรอยที่ไม่ดีที่มีความร้ายแรง และ ร่องรอยที่ไม่ดีที่มีความร้ายแรงน้อย ซึ่งมีวิธีในการเลือกวิธีการเรียนรู้เครื่องให้เหมาะสมตามแต่ละกลุ่มดังต่อไปนี้

1) การเลือกวิธีการเรียนรู้เครื่องที่เหมาะสมกับร่องรอยที่ไม่ดีที่มีความร้ายแรง

จากร่องรอยที่ไม่ดีชนิดที่มีความร้ายแรงมาก คือ Switch Statement Feature Envy Message Chains และ Middle Man เพราะ ร่องรอยที่ไม่ดีนี้ขัดกับการออกแบบเชิงวัตถุ หรือไม่ได้ใช้ประโยชน์ของการออกแบบเชิงวัตถุ [17] ร่องรอยที่ไม่ดีชนิดนี้ควรเลือกวิธีการเรียนรู้เครื่องที่มีค่าความผิดพลาดแบบ I น้อยที่สุด เพราะว่า ร่องรอยที่ไม่ดีชนิดนี้ต้องการความถูกต้องในการทำนาย เพื่อป้องกันไม่ให้เกิดการทำนายคลาสที่มีร่องรอยที่ไม่ดีเป็น คลาสที่ไม่มีร่องรอยที่ไม่ดี หรือ ไม่ต้องการให้คลาสที่มีร่องรอยที่ไม่ดีหลุดออกไป จากตารางที่ 4.63 สรุปได้ว่า

- Switch Statement ควรใช้วิธีการเรียนรู้เครื่องประเภท Naive Bayes

- Feature Envy ควรใช้วิธีการเรียนรู้เครื่องประเภท Random Forest

- Message Chains ควรใช้วิธีการเรียนรู้เครื่องประเภท Logistic regression IB1 IBk หรือ Random Forest

- Middle Man ควรใช้วิธีการเรียนรู้เครื่องประเภท Logistic regression IB1 หรือ IBk

ตารางที่ 4.63 ความผิดพลาดแบบ I (Type I Error) หรือ False negative (FN) ของวิธีการเรียนรู้เครื่องที่ตามประเภทของร่องรอยที่ไม่ดี

	Feature Envy (%)	Long Method (%)	Long Parameter Lists (%)	Lazy Class (%)	Message Chains (%)	Middle Man (%)	Switch Statement (%)
Naive Bayes	1	0	0	1	4	5	1
Logistic regression	2	5	11	5	0	1	2
IB1	1	3	2	4	0	1	2
IBk	1	2	2	5	0	1	2
VFI	0	2	1	0	2	2	1
J48	9	5	3	5	6	8	2
Random Forest	1	2	1	5	0	2	2

2) การเลือกวิธีการเรียนรู้เครื่องที่เหมาะสมกับร่องรอยที่ไม่ดีที่มีความร้ายแรงน้อย

จากร่องรอยที่ไม่ดีชนิดที่มีความร้ายแรงน้อย คือ Long Method Long Parameter Lists และ Lazy Class ที่แสดงถึงโค้ดที่มีขนาดใหญ่ [17] ร่องรอยที่ไม่ดีชนิดนี้ควรเลือกวิธีการเรียนรู้เครื่องที่มีค่าความผิดพลาดแบบ II น้อยที่สุด เพราะว่า ร่องรอยที่ไม่ดีชนิดนี้ต้องการความถูกต้องในการทำนาย เพื่อป้องกันไม่ให้เกิดการทำนายคลาสที่ไม่มีร่องรอยที่ไม่ดีเป็น คลาสที่มีร่องรอยที่ไม่ดี จากตารางที่ 4.64 สรุปได้ว่า

- Long Method ควรใช้วิธีการเรียนรู้เครื่องประเภท Random Forest
- Long Parameter Lists ควรใช้วิธีการเรียนรู้เครื่องประเภท Random Forest
- Lazy Class ควรใช้วิธีการเรียนรู้เครื่องประเภท J48

ตารางที่ 4.64 ความผิดพลาดแบบที่ II (Type II Error) หรือ False positive (FP) ของวิธีการเรียนรู้เครื่องที่ตามประเภทของร่องรอยที่ไม่ดี

	Feature Envy (%)	Long Method (%)	Long Parameter Lists (%)	Lazy Class (%)	Message Chains (%)	Middle Man (%)	Switch Statement (%)
Naive Bayes	48	27	70	50	3	5	14
Logistic regression	5	6	2	3	2	2	2
IB1	3	4	3	4	2	2	4
IBk	3	4	3	1	2	2	3
VFI	49	30	77	56	5	11	29
J48	2	5	3	0	4	1	1
Random Forest	1	6	1	1	2	2	4

4.6.3. การเลือกวิธีการเรียนรู้เครื่องที่เหมาะสมกับร่องรอยที่ไม่ดีทุกประเภท

การเปรียบเทียบประสิทธิภาพของวิธีการเรียนรู้เครื่องในการทำนายร่องรอยที่ไม่ดีทุกประเภท โดยการวิเคราะห์ผลแบบ Pair Sample T-test ที่ระดับนัยสำคัญ 0.05 แล้วนำมาเปรียบเทียบกับค่าวิกฤตแล้ว จะได้ตารางการยอมรับสมมติฐานหลัก หรือ ยอมรับสมมติฐานรองตามวิธีการเรียนรู้เครื่อง ซึ่งแสดงในตารางที่ 4.61 สามารถสรุปได้ว่า IBk IB1 Random Forest และ J48 มีประสิทธิภาพในการทำนายร่องรอยที่ไม่ดีทุกประเภทไม่แตกต่าง และ มีประสิทธิภาพทำนายร่องรอยที่ไม่ดีมากกว่า Logistic regression Naive Bayes และ VFI ที่ระดับนัยสำคัญ 0.05

บทที่ 5
สรุปผลการวิจัย และ ข้อเสนอแนะ

5.1. บทสรุป

วิทยานิพนธ์ฉบับนี้ได้นำเสนอวิธีการในการทำนายร่องรอยที่ไม่ดีของโมเดลการออกแบบซอฟต์แวร์โดยใช้วิธีการเรียนรู้ของเครื่องสำหรับร่องรอยที่ไม่ดี 7 ประเภท คือ Lazy Class, Feature Envy, Middle Man, Message Chains, Long Method, Long Parameter Lists และ Switch Statement แต่ละชุดข้อมูลที่ประกอบด้วยร่องรอยที่ไม่ดีแต่ละชนิด และ ค่ามาตรวัดการออกแบบ งานวิจัยนี้ใช้วิธีการเรียนรู้เครื่อง 7 ชนิดที่อยู่ในโปรแกรม Weka ในการทำนายร่องรอยที่ไม่ดีในแต่ละชุดข้อมูล งานวิจัยนี้ทำการคำนวณประสิทธิภาพการทำนายร่องรอยที่ไม่ดีหลากหลายวิธี หลังจากนั้น ก็นำผลการประเมินมาเปรียบเทียบความสามารถในการทำนายของวิธีการเรียนรู้เครื่องแต่ละประเภท ตามความเหมาะสมในการเลือกวิธีการเรียนรู้เครื่องเพื่อนำไปใช้ใช้งาน

การเลือกวิธีการเรียนรู้เครื่องที่เหมาะสมกับร่องรอยที่ไม่ดีแบบทั่วไป โดยการคำนวณประสิทธิภาพการทำนายร่องรอยที่ไม่ดีด้วย การวัดความถูกต้องในทำนายร่องรอยที่ไม่ดี (Bad-smell prediction accuracy) ความไว และ ความจำเพาะ (Sensitivity & Specificity) และ ค่าพยากรณ์ของการทดสอบ (Predictive value of tests) เราสามารถสรุปได้ว่าไม่มีวิธีการเรียนรู้เครื่องที่มีความสามารถทำนายร่องรอยที่ไม่ดีทุกร่องรอยได้เป็นอย่างดี ซึ่งต้องตัดสินความสามารถของแต่ละวิธีการเรียนรู้เครื่องตามแต่ละร่องรอยที่ไม่ดี

การเลือกวิธีการเรียนรู้เครื่องที่เหมาะสมตามความร้ายแรงของร่องรอยที่ไม่ดี สามารถสรุปได้ว่า ร่องรอยที่ไม่ดีกลุ่มที่มีความรุนแรงมาก คือ Switch Statement Feature Envy Message Chains และ Middle Man ควรใช้วิธีการเรียนรู้เครื่องแตกต่างกับ ร่องรอยที่ไม่ดีกลุ่มที่มีความรุนแรงน้อย คือ Long Method Long Parameter Lists และ Lazy Class ซึ่งต้องตัดสินเลือกใช้วิธีการเรียนรู้เครื่องตามแต่ละร่องรอยที่ไม่ดีเช่นเดียวกัน

การเลือกวิธีการเรียนรู้เครื่องที่เหมาะสมกับร่องรอยที่ไม่ดีทุกประเภท สามารถสรุปได้ว่าว่า IBk IB1 Random Forest และ J48 มีประสิทธิภาพในการทำนายร่องรอยที่ไม่ดีไม่แตกต่าง และ มีประสิทธิภาพทำนายร่องรอยที่ไม่ดีมากกว่า Logistic regression Naive Bayes และ VFI ที่ระดับนัยสำคัญ 0.05

วิธีการทำนายร่องรอยที่ไม่ดีที่ได้นำเสนอในวิทยานิพนธ์นี้ เป็นแนวทางสำหรับช่วยในการทำนายร่องรอยที่ไม่ดีแต่ละประเภท ส่วนการตัดสินใจ ว่าจะใช้วิธีการเรียนรู้เครื่องประเภทใด

ขึ้นอยู่กับการตัดสินใจของผู้ใช้ว่าวิธีการไหนจะเหมาะกับซอฟต์แวร์ประเภทไหน และ ต้องการความถูกต้องมากน้อยเพียงใด

5.2. ปัญหา และ ข้อจำกัด

จากการดำเนินงานวิจัยนี้พบปัญหาและข้อจำกัดของระบบ ดังต่อไปนี้

1. สามารถการทำนายร่องรอยที่ไม่ดีเพียง 7 ชนิดเท่านั้น
2. ไม่มีเครื่องมือที่สามารถทำนายร่อยรอยที่ไม่มีโดยอัตโนมัติ ทำให้จะต้องมีการเตรียมข้อมูล และ การทำนายร่องรอยไม่ดีด้วยตัวเองเท่านั้น
3. สามารถระบุได้ว่ามีร่องรอยที่ไม่ดีในคลาสไหนบ้าง แต่ไม่สามารถระบุตำแหน่งที่เกิดร่องรอยที่ไม่ดีได้

5.3. ข้อเสนอแนะ

จากงานวิจัยนี้ ยังมีหัวข้อที่ยังไม่ได้ทำการทดลองในการตรวจจับร่องรอยที่ไม่ดีที่น่าจะสามารถเพิ่มความสามารถของการทำนายร่องรอยที่ไม่ดีดังต่อไปนี้

1. เราสามารถเปลี่ยน หรือ เพิ่มจำนวนมาตรวัดการออกแบบได้ และ เรายังสามารถปรับเปลี่ยนวิธีการเรียนรู้เครื่องอื่น ๆ ให้เพื่อให้ได้ประสิทธิภาพในการทำนายมากที่สุด
2. เราสามารถทำการทดลองกับบางร่อยรอยที่ไม่ดีที่ยังไม่ได้ทำ และ เรายังสามารถหาความสัมพันธ์ของร่องรอยไม่ดีแต่ละชนิดว่าการเกิดร่องรอยที่ไม่ดีนั้นมีความสัมพันธ์กันอย่างไร
3. ในงานวิจัยนี้เราใช้วิธีการเรียนรู้เครื่อง 7 ชนิด โดยการระบุคุณสมบัติของวิธีการเรียนรู้เครื่องตามค่าเริ่มต้น เพราะฉนั้น เราสามารถปรับเปลี่ยนค่าคุณสมบัติของวิธีการเรียนรู้เครื่อง เพื่อให้ได้ผลการทดลองที่ดีที่สุด

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

รายการอ้างอิง

[1] Henry, S., and Selig, C. Predicting Source-Code Complexity at the Design Stage. IEEE Software, 7, 2 (March 1990): 36-44.

[2] Fowler, M. Refactoring: Improving the Design of Existing Code. United States: Addison-Wesley Longman Publishing.

[3] Munro, M. Product Metrics for Automatic Identification of "Bad Smell" Design Problems in Java Source-Code. Proceedings of the 11th IEEE International Software Metrics Symposium (METRICS 2005).

[4] Marinescu, R. Detecting Design Flaws via Metrics in Object-Oriented Systems. Proceedings of the 39th International Conference and Exhibition on Technology of Object-Oriented Languages and Systems (TOOLS39) (August 2001):173.

[5] Pienlert, T. and Muenchaisri, P. Bad-smell Detection using Object-Oriented Software Metrics. Proceedings of the International Society of Computers and Their Applications (ISCA) International Conference on Computer Science, Software Engineering, Information Technology, e-Business, and Applications (CSITeA'04) (December 2004).

[6] Yeesoon K. Design and implementation of a tool for detecting bad smells in Java program, Master's Thesis, Department of Computer Engineering, Faculty of Engineering, Chulalongkorn University, 2008.

[7] Mekruksavanich, S. and Muenchaisri, P. Using Declarative Meta Programming for Design Flaws Detection in Object-Oriented Software. Proceeding of The 2009 International Conference on Computer Design and Applications (ICCDA2009) (May 2009).

[8] Jiang, Y., Cukic, B., Menzies, T., and Bartlow, N. Comparing Design and Code Metrics for Software Quality Prediction. Proceedings of the PROMISE 2008 Workshop (ICSE) 2008.

[9] Breiman, L.. Random Forests. In Machine Learning, pp.5-32. 2001.

[10] Brito e Abreu, F., Ochoa, L., and Goulao, M. The MOOD metrics set. INESC/ISEG Internal Report, 1998.

[11] Witten, I.H., and Frank, E. Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations [Online], 1999. Available from : http://www.cs.waikato.ac.nz/ml/weka/ [2010, October 10]

[12] Kohavi, R. A study of cross-validation and bootstrap for accuracy estimation and model selection. Proceedings of the International Joint Conference on AI (1995):1137–1145.

[13] Antoniol, G., Canfora, G., Casazza, G., Lucia, A.D., and Merlo, E. Recovering tracebility links between code and documentation. IEEE Transactions on Software Engineering (2002): 970–983.

[14] Antoniol, G., Casazza, G., Penta, M., and Fiutem, R. Object-oriented design patterns recovery. Journal of Systems and Software (2001): 181–196.

[15] Deitel, H.M., and Deitel, P.J. Java: How to program. New Jersey: Prentice Hall, 2004.

[16] William, C. Refactoring Workbook. United States: Addison-Wesley Longman Publishing Co., Inc., 2003.

[17] Mantyla, M., Vanhanen, J., and Lassenius, C. A taxonomy and an initial empirical study of bad smells in code. Proceedings of ICSM'2003 (2003):381-384.

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

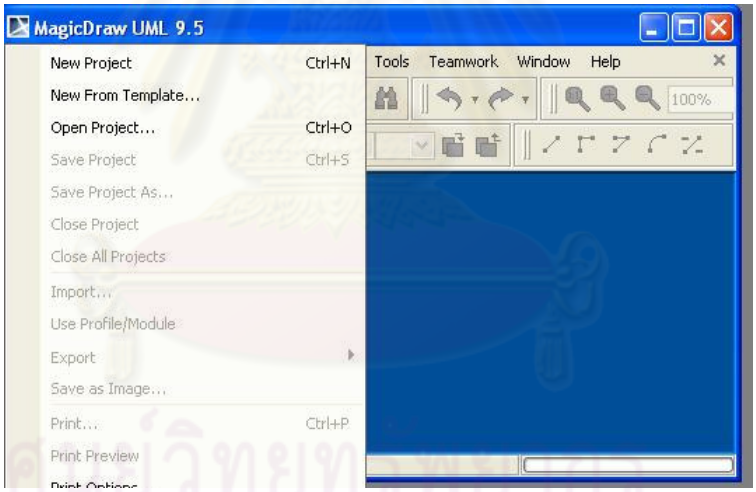
ภาคผนวก

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ก
การแปลงรหัสต้นฉบับเป็นแผนภาพคลาส การคำนวณค่ามาตรวัด

การแปลงรหัสต้นฉบับเป็นแผนภาพคลาส ในงานวิจัยนี้ใช้เครื่องมือ MagicDraw UML เวอร์ชัน 9.5 ซึ่งเป็นโปรแกรมสำหรับวาดแผนภาพยูเอ็มแอล ที่มีความสามารถในการแปลงกลับแผนภาพจากรหัสต้นฉบับ และ มีความสามารถในการคำนวณค่ามาตรวัด ในส่วนนี้จะแสดงการทำงานในส่วนของการแปลงกลับแผนภาพคลาส และ คำนวณค่ามาตรวัดจากแผนภาพคลาส มีรายละเอียดดังนี้

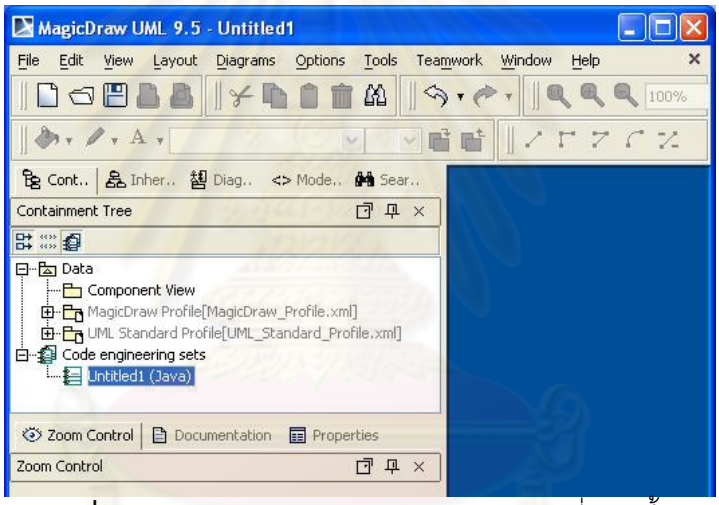
หน้าจอหลักของโปรแกรมแสดงดังรูปที่ ก-1 โดยเริ่มแรกให้เราสร้างโปรเจคใหม่โดยทำการเลือกเมนู File>New project จากนั้นโปรแกรมจะแสดงหน้าต่างย่อยทางซ้ายมือของโปรแกรม ซึ่งแสดงรายละเอียดโครงสร้างและองค์ประกอบของโปรเจคที่สร้าง โดยองค์ประกอบหลักของโปรเจคคือ Code engineering sets ซึ่งเป็นองค์ประกอบสำหรับสร้าง ระบุและทำงานต่างๆ ที่เกี่ยวข้องกับรหัสต้นฉบับและการแปลงกลับแผนภาพ



รูปที่ ก-1หน้าจอหลักของโปรแกรม MagicDraw UML

ในการแปลงกลับแผนภาพให้เริ่มจากสร้าง Code engineering setsโดยคลิกขวาที่ Code engineering sets > New และเลือกภาษาจาวา ดังแสดงในรูปที่ ก-2 จากนั้นโปรแกรมจะแสดงหน้าต่างองค์ประกอบต่างๆ ดังแสดงในรูปที่ ก-3

รูปที่ ก-2 หน้าต่างการสร้าง Code engineering sets

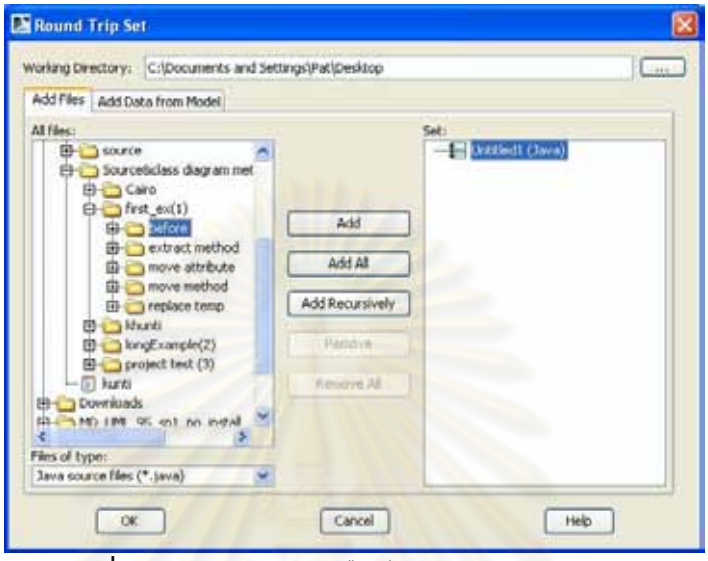


รูปที่ ก-3 หน้าต่าง Code engineering sets ที่สร้างขึ้น

หลังจากการสร้าง Code engineering sets เสร็จแล้ว ให้นำเข้าไฟล์รหัสต้นฉบับของซอฟต์แวร์ที่ต้องการแปลงกลับ โดยคลิกขวาที่ Code engineering sets แล้วเลือกคำสั่ง Edit จากนั้นโปรแกรมจะแสดงหน้าต่าง Round trip set สำหรับเลือกรายการของไฟล์รหัสต้นฉบับที่ต้องการ ดังรูปที่ ก-4 ในการเลือกไฟล์สามารถเลือกได้ 3 แบบคือ

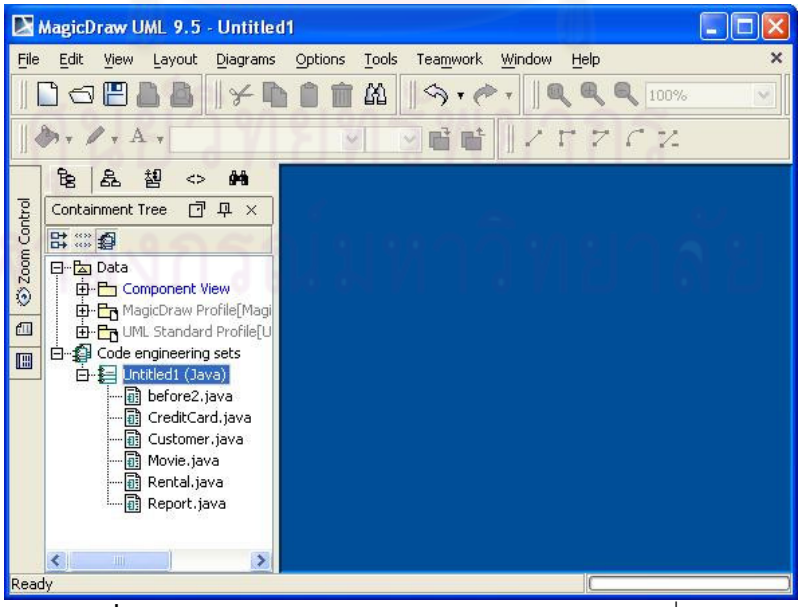
- เลือกไฟล์เดียวโดยใช้คำสั่ง Add
- เลือกทั้งหมดโดยใช้คำสั่ง Add all
- เลือกแบบวนกลับโดยใช้คำสั่ง Add recursively

โดยงานวิจัยนี้ใช้วิธีการเลือกแบบวนกลับคือการเลือกไฟล์รหัสต้นฉบับตามประเภทที่กำหนดในไดเรกทอรีปัจจุบัน และ ไดเรกทอรีย่อยทั้งหมดโดยอัตโนมัติ



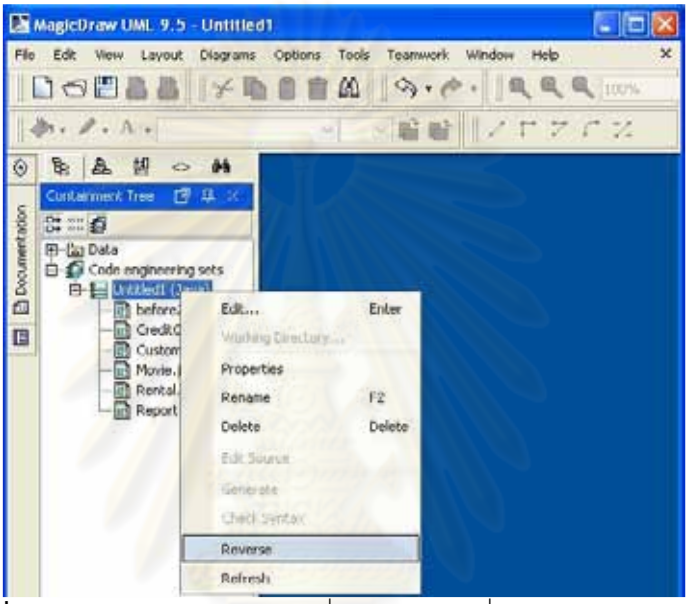
รูปที่ ก-4 หน้าต่างการเลือกไฟล์รหัสต้นฉบับนำเข้า

หลังจากทำ การเลือกไฟล์ที่ต้องการเข้าสู่โปรเจคแล้ว องค์ประกอบของ Code engineering sets แสดงรายการของไฟล์รหัสต้นฉบับที่ได้นำเข้าในส่วน Code engineering sets ดังรูปที่ ก-5



รูปที่ ก-5 หน้าต่างรายละเอียดของไฟล์รหัสต้นฉบับที่เลือก

หลังจากนั้นให้ทำการแปลงกลับรหัสต้นฉบับเป็นแผนภาพคลาสโดยคลิกขวาที่ Code engineering sets เลือกคำสั่ง Reverse ดังแสดงในรูปที่ ก- 6 จากนั้นโปรแกรมจะแสดงหน้าต่างตัวเลือกในการแปลงกลับ ซึ่งมี 4 ตัวเลือก ให้ผู้ใช้กำหนดค่าดังแสดงในรูปที่ ก-7 โดยแต่ละส่วนของหน้าต่าง Reverse Options มีรายละเอียดดังนี้



รูปที่ ก-6 หน้าต่างการเลือกคำสั่ง Reverse เพื่อแปลงกลับแผนภาพ

รูปที่ ก-7 หน้าต่างการกำหนดค่าการแปลงกลับแผนภาพ

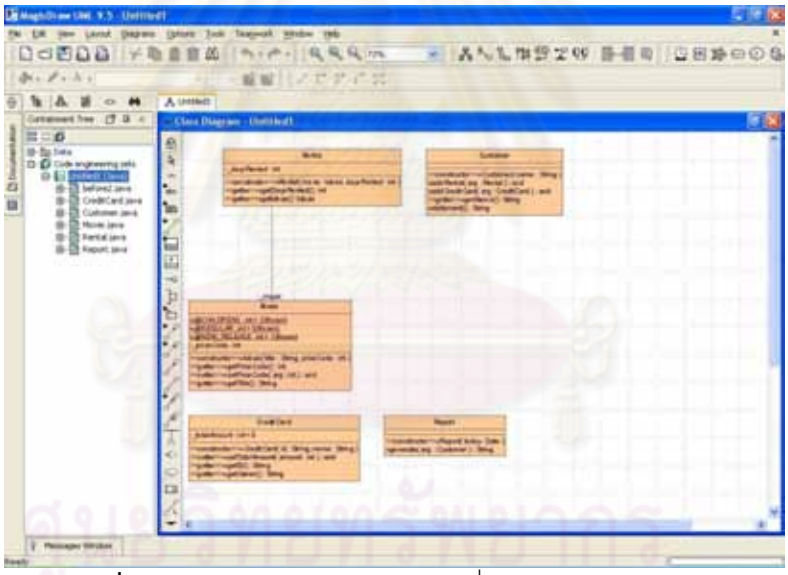
- ตัวเลือก Visualization คือตัวเลือกในการสร้างแผนภาพที่ได้จากการแปลงกลับโดยสามารถเลือกให้แสดงตัวช่วยสร้างแผนภาพ และ สร้างแผนภาพคลาสได้

- ตัวเลือก General คือตัวเลือกให้โปรแกรมทำการมองหาคลาสเพิ่มเติมในคลาสพาธหรือไม่

- ตัวเลือก Create class field as คือตัวเลือกการแสดงผลความสัมพันธ์แอสโซซิเอชันระหว่างคลาส ให้อยู่ในรูปแบบของแอททริบิวต์ของอีกคลาส หรือแสดงเป็นความสัมพันธ์แอสโซซิเอชัน

- ตัวเลือก Model refresh type คือตัวเลือกเงื่อนไขการทำให้โมเดลและรหัสต้นฉบับสอดคล้องกัน โดยมีตัวเลือกคือ ให้ทั้งโมเดลและรหัสต้นฉบับทำการแก้ไขให้สอดคล้องกันเมื่อมีการแก้ไขฝ่ายหนึ่งฝ่ายใด หรือให้โมเดลทำการแก้ไขให้สอดคล้องกับรหัสต้นฉบับเพียงฝ่ายเดียว

จากนั้นเมื่อทำการแปลงกลับข้อมูลเสร็จสิ้น โปรแกรมจะแสดงแผนภาพคลาส ดังแสดงในรูปที่ ก-8

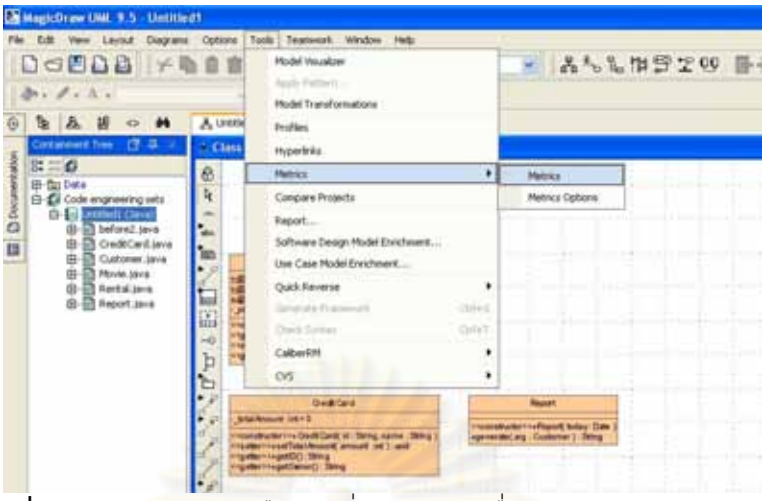


รูปที่ ก-8 หน้าต่างแผนภาพคลาสที่ได้จากการแปลงกลับ

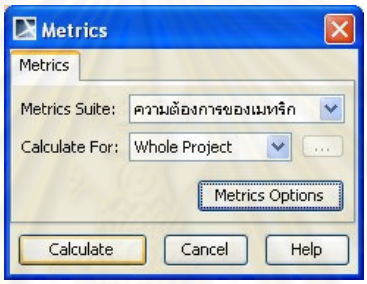
หลังจากการแปลงกลับรหัสต้นฉบับเป็นแผนภาพคลาส ขั้นตอนต่อมาคือการคำนวณค่ามาตรวัด ทำโดยคลิกซ้ายเลือกเมนู เลือกคำสั่ง Tools > Metrics > Metrics ดังแสดงในรูปที่ ก- 9 จากนั้นโปรแกรมจะแสดงหน้าต่างตัวเลือกมาตรวัดและจำนวนคลาสที่ต้องการ ให้ผู้ใช้กำหนดค่าดังแสดงในรูปที่ ก-10 โดยแต่ละส่วนของหน้าต่าง Metrics มีรายละเอียดดังนี้

- ตัวเลือก Metrics Suite คือ ตัวเลือกมาตรวัดที่ต้องการคำนวณ

- ตัวเลือก Calculate For คือ ตัวเลือกจำนวนคลาสที่ต้องการคำนวณ

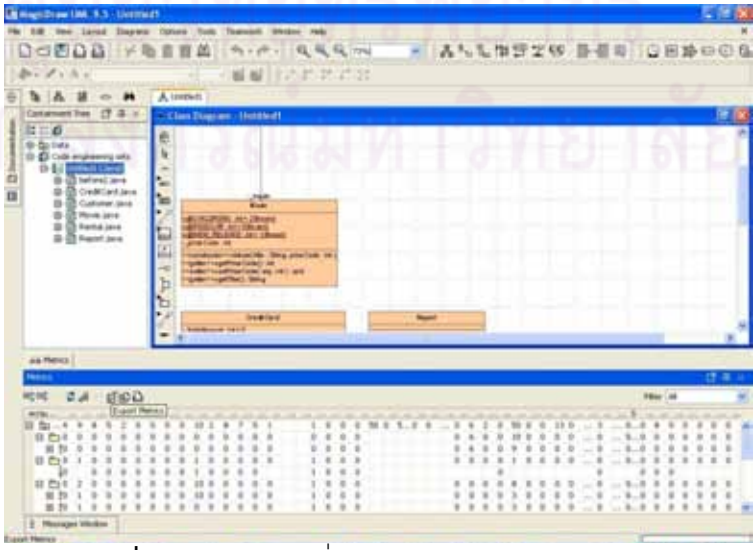


รูปที่ ก-9 หน้าต่างการเลือกคำสั่ง Metrics เพื่อการคำนวณค่ามาตรวัด

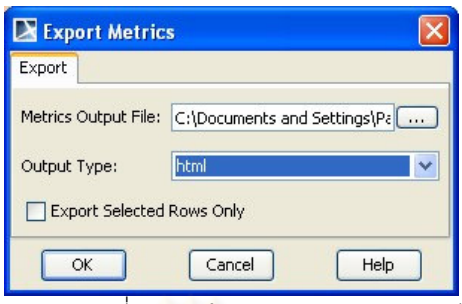


รูปที่ ก-10 หน้าต่างการกำหนดค่าการคำนวณมาตรวัด

จากนั้นเมื่อทำการคำนวณข้อมูลเสร็จสิ้น โปรแกรมจะแสดงค่ามาตรวัด ดังแสดงใน รูปที่ ก-11 หลังจากนั้นให้ทำการบันทึกข้อมูลโดยการเลือกดิสต์ (Export Metrics) เหนือตารางค่ามาตรวัดจากนั้นจะแสดงหน้าต่างตำแหน่งที่จะบันทึกข้อมูล และ ประเภทไฟล์ที่ต้องการบันทึก ดังแสดงใน รูปที่ ก-12



รูปที่ ก-11 ผลลัพธ์ที่ได้จากการคำนวณค่ามาตรวัด



รูปที่ ก-12 หน้าต่างตำแหน่งที่จะบันทึกข้อมูล และ ประเภทไฟล์ที่ต้องการบันทึก

ตารางค่ามาตรวัดจะถูกบันทึกในตำแหน่งที่ต้องการตามตัวอย่างนี้เราบันทึกในชื่อ “ความต้องการของเมทริก” และเป็นประเภท HTML ดังแสดงใน รูปที่ ก-13 ซึ่งเราสามารถเปิดได้ด้วยบราวเซอร์ ดังแสดงใน รูปที่ ก-14 ซึ่งสามารถนำไปใช้เป็นข้อมูลนำเข้าในการทำนายร่องรอยที่ไม่ดี

รูปที่ ก-13 ข้อมูลค่ามาตรวัดที่ได้จากการบันทึก

Metrics Report

Element	NA	NC	NM	NO	NP	C_CROL	C_INST	C_OBJ	C_ATTR	C_PARAM	RFC	WAC	WMC
Class CreditCard()	1	0	5	4	2	0	0	0	0	1	4	2	6
Class Customer()	0	0	5	5	1	0	0	0	0	1	5	0	7
Package Default()	0	1	0	0	0	0	0	0	0	1	0	0	0
Class Date(Default)	0	0	0	0	0	0	0	0	0	1	0	0	0
Package java()	0	2	0	0	0	0	0	0	0	10	0	0	0
Package java()	0	0	0	0	0	0	0	0	0	0	0	0	0
Package lang(java)	0	1	0	0	0	0	0	0	0	10	0	0	0
Package lang(java)	0	0	0	0	0	0	0	0	0	0	0	0	0

รูปที่ ก-14 ตารางค่ามาตรวัด

ภาคผนวก ข
การจำแนกประเภทข้อมูลเพื่อใช้ในการทำนายร่องรอยที่ไม่ดี

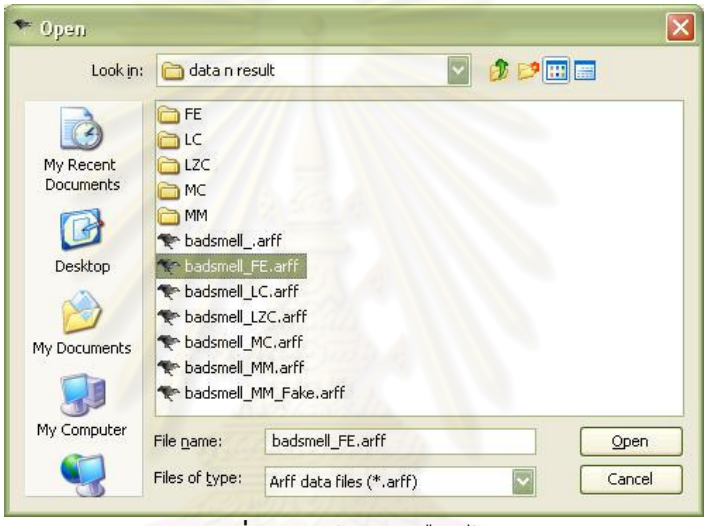
การจำแนกประเภทข้อมูล เพื่อใช้ในการทำนายร่องรอยที่ไม่ดีในงานวิจัยนี้ใช้เครื่องมือ Weka เวอร์ชัน 3.6 ซึ่งเป็นโปรแกรมที่ใช้ในการวิเคราะห์ หรือ ทำนายสิ่งต่าง ๆ ที่จะเกิดขึ้นที่ได้รับความนิยมเป็นอย่างมาก[11] ซึ่งการค้นหาความรู้ ในส่วนนี้จะแสดงการทำงานในส่วนของการทำนายร่องรอยที่ไม่ดี มีรายละเอียดดังนี้

หน้าจอการแสดงผลของโปรแกรมแสดงดังรูปที่ ข-1 ให้เราเลือก Explorer แล้วโปรแกรมจะแสดงหน้าจอหลักของโปรแกรมแสดงดังรูปที่ ข-2 ที่แท็บ Preprocess ให้คลิดปุ่ม Open ... เพื่อเลือกไฟล์ ARFF ซึ่งเป็นไฟล์ข้อมูลที่ประกอบด้วย จำนวนร่องรอยที่ไม่ดี และ ค่ามาตรวัดการออกแบบ เพื่อใช้ในการจำแนกประเภทข้อมูล

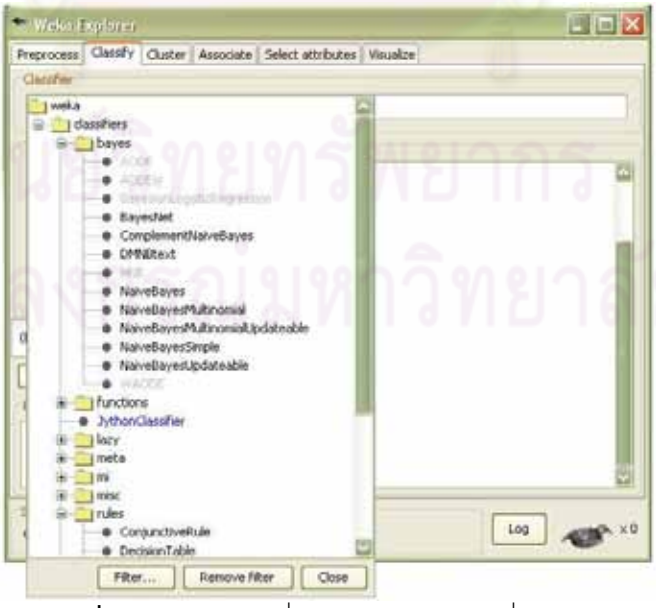
รูปที่ ข-1หน้าจอการแสดงผล

รูปที่ ข-2 หน้าจอหลักของโปรแกรม

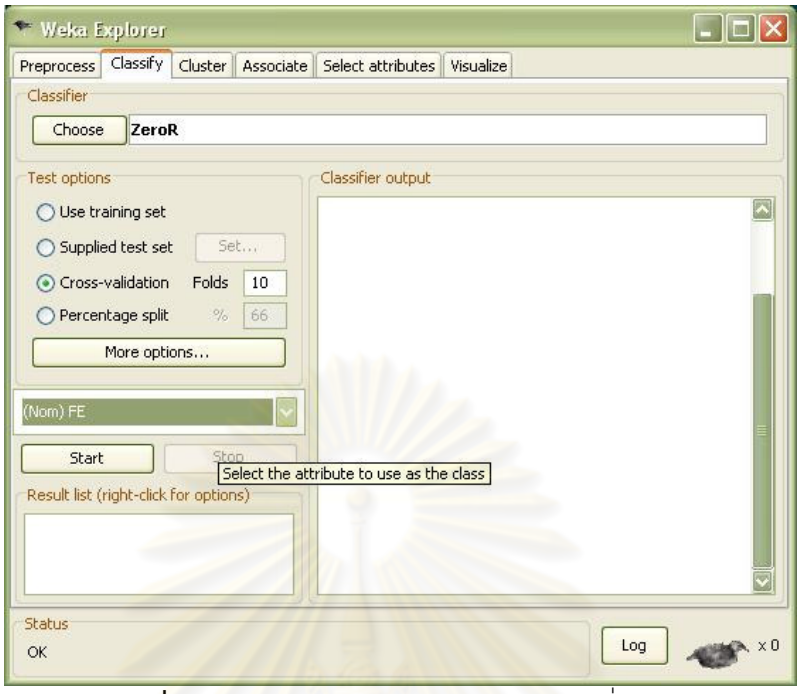
โปรแกรมจะแสดงหน้าต่างเพื่อเลือกไฟล์แสดงดังรูปที่ ข-3 ให้เลือกไฟล์ ARFF ที่เราเตรียมไว้แล้ว หลังจากนั้นให้คลิกที่แท็บ Classify แล้วกดปุ่ม Choose เพื่อเลือก Classifier หรือวิธีการเรียนรู้เครื่อง ที่เราต้องการ แสดงดังรูปที่ ข-4 แล้วให้ทำการเลือกวิธีการทดสอบข้อมูลโดยการเลือกวิธีการเลือกสุ่มข้อมูลแบบความเที่ยงตรง (Cross validation) และ ใส่ค่าเป็น 10 หลังจากนั้นให้เลือกพารามิเตอร์ที่เราต้องการทำนาย ในที่นี้ คือ ร่องรอยที่ไม่ดี ในตัวอย่างจะเลือก FE (Feature envy) ซึ่งเป็นหนึ่งในร่องรอยที่ไม่ดีที่งานวิจัยนี้ใช้ในการทดลอง แล้วกดปุ่ม Start เพื่อทำการทำนายร่องรอยที่ไม่ดี แสดงดังรูปที่ ข-5



รูปที่ ข-3 หน้าต่างเลือกไฟล์

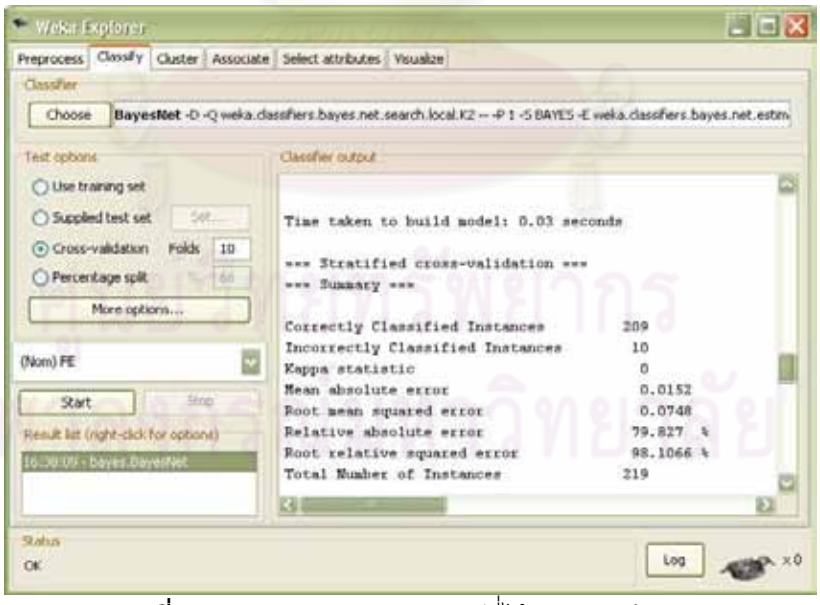


รูปที่ ข-4 หน้าต่างเพื่อเลือก Classifier ที่ต้องการ



รูปที่ ข-5 วิธีการเลือกสุ่มข้อมูลแบบความเที่ยงตรง

ระบบจะแสดงผลลัพธ์ที่ได้จากการทำนายที่ด้านขวามือ แสดงดังรูปที่ ข-6 ให้ทำการบันทึกข้อมูลโดยการเลือกข้อมูลที่ได้ใน Classifier output แล้วคัดลอกเก็บไว้ในไฟล์ที่ต้องการ



รูปที่ ข-6 ระบบจะแสดงผลลัพธ์ที่ได้จากการทำนาย

ภาคผนวก ค
ผลงานตีพิมพ์

ผลงานวิจัยนี้ ได้รับคัดเลือกให้ถูกตีพิมพ์ในงานสัมมนาวิชาการระหว่างประเทศ "The 8th International Joint Conference on Computer Science and Software Engineering" ซึ่งได้จัดขึ้นที่ Mahidol University, Nakhon Pathom, Thailand ระหว่างวันที่ 11 – 13 พฤษภาคม 2011

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ประวัติผู้เขียนวิทยานิพนธ์

นายนครินทร์ มณีรัตน์ เกิดวันที่ 4 พฤศจิกายน พ.ศ. 2527 ที่จังหวัดนครราชสีมา สำเร็จการศึกษาปริญญาวิทยาศาสตรบัณฑิต สาขาวิชาศาสตร์คอมพิวเตอร์ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยธรรมศาสตร์ ในปีการศึกษา 2548 และได้เข้าศึกษาในหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิทยาศาสตร์คอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ปีการศึกษา 2551

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย