



## รายงาน

การวิเคราะห์ภาพเซลล์มะเร็งเม็ดเลือดขาวชนิด ALL เพื่อการจำแนกประเภทด้วยโมเดลการเรียนรู้  
Analysis and Classification of Acute Lymphoblastic Leukemia (ALL) Cell Images Using  
Machine Learning Techniques

## เสนอ

รองศาสตราจารย์ ดร.ไกรศักดิ์ เกษร

## จัดทำโดย

นางสาวสิริฉัตร ปานน้อย

รหัสนิสิต 65315185

รายงานฉบับนี้เป็นส่วนหนึ่งของรายวิชาวิทยาศาสตร์ข้อมูล (254486)

ภาคเรียนที่ 2 ปีการศึกษา 2567

สาขาวิชาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยนเรศวร

## คำนำ

รายงานฉบับนี้เป็นส่วนหนึ่งของรายวิชาวิทยาศาสตร์ข้อมูล (254486) จัดทำขึ้นเพื่อเป็นส่วนหนึ่งของการศึกษาค้นคว้าและพัฒนาโครงงานด้านวิทยาการข้อมูล โดยมีจุดประสงค์เพื่อพัฒนาระบบสำหรับวิเคราะห์และจำแนกประเภทของภาพเซลล์มะเร็งเม็ดเลือดขาวชนิด Acute Lymphoblastic Leukemia (ALL) ซึ่งเป็นมะเร็งที่พบได้บ่อยในเด็กและเยาวชน โดยใช้เทคนิคทางปัญญาประดิษฐ์และการเรียนรู้ของเครื่อง (Machine Learning) ในการประมวลผลและวิเคราะห์ภาพ เพื่อให้สามารถจำแนกเซลล์ออกเป็นกลุ่มย่อย ได้แก่ benign, early, pre และ pro ได้อย่างแม่นยำ

เนื้อหาในรายงานครอบคลุมตั้งแต่การศึกษาที่มาและความสำคัญของปัญหา การรวบรวมและวิเคราะห์ข้อมูลภาพ การพัฒนา Workflow สำหรับการประมวลผลภาพ การสร้างและเปรียบเทียบประสิทธิภาพของโมเดลต่าง ๆ เช่น Deep Learning (Keras), Random Forest, MLP และ SVM รวมถึงการนำเสนอผลลัพธ์ในรูปแบบ Visualization เพื่อให้เข้าใจได้ง่ายและชัดเจน

เนื้อหาในรายงานฉบับนี้ครอบคลุมทั้งเชิงทฤษฎีและการประยุกต์ใช้จริง โดยมีเป้าหมายเพื่อแสดงให้เห็นถึงศักยภาพของการนำเทคโนโลยีปัญญาประดิษฐ์มาใช้ในการสนับสนุนกระบวนการทางการแพทย์อย่างเป็นรูปธรรม

ผู้จัดทำ

นางสาวสิริฉัตร ปานน้อย

## สารบัญ

เรื่อง	หน้า
<b>บทที่ 1 บทนำ</b>	<b>8</b>
1.1 ที่มาและความสำคัญ	8
1.2 วัตถุประสงค์ของรายงาน	8
1.3 ขอบเขต	8
1.4 ประโยชน์ที่คาดว่าจะได้รับ	9
<b>บทที่ 2 ชุดข้อมูล (Dataset)</b>	<b>10</b>
2.1 แหล่งที่มาของข้อมูล	10
2.2 ลักษณะและเนื้อหาของข้อมูล	11
2.3 ประเภทของข้อมูล	12
<b>บทที่ 3 ขั้นตอนการพัฒนา</b>	<b>13</b>
3.1 การนำเข้าข้อมูล	14
3.2 การดึงคุณลักษณะจากภาพ (Feature Extraction)	16
3.3 การสร้างโมเดล (Modeling)	20
3.3.1 Deep Learning (Keras)	20
3.3.2 Random Forest	23
3.3.3 MLP	24
3.3.4 SVM	24
3.4 การประเมินผล (Evaluation & Visualization)	25
3.4.1 Deep Learning (Keras)	25
3.4.2 Random Forest	28
3.4.3 MLP	29
3.4.4 SVM	30
3.4.5 Visualization สำหรับโมเดล Random Forest, MLP และ SVM	31
3.4.6 เปรียบเทียบแต่ละ Model	33
<b>บทที่ 4 ผลการประเมินประสิทธิภาพของโมเดล</b>	<b>36</b>
4.1 Deep Learning (Keras)	36
4.1.1 Test	36
4.1.2 Validation	37
4.2 Random Forest	38
4.2.1 Test	38
4.2.2 Validation	39

## สารบัญ (ต่อ)

เรื่อง	หน้า
4.3 MLP	41
4.3.1 Test	41
4.3.2 Validation	42
4.4 SVM	43
4.4.1 Test	43
4.4.2 Validation	44
4.5 เปรียบเทียบแต่ละ Model	46
4.5.1 Test	46
4.5.2 Validation	47
4.5.3 Accuracy test and validation	48
<b>บทที่ 5 บทสรุป</b>	<b>50</b>
5.1 สรุปผลการดำเนินงาน	50
5.2 ข้อเสนอแนะในการพัฒนาเพิ่มเติม	51

## สารบัญภาพ

เรื่อง	หน้า
ภาพที่ 2.1 ตัวอย่างภาพโรคมะเร็งเม็ดเลือดขาวชนิดเฉียบพลัน (ALL) ระยะ benign (ไม่เป็นมะเร็ง)	11
ภาพที่ 2.2 ตัวอย่างภาพโรคมะเร็งเม็ดเลือดขาวชนิดเฉียบพลัน (ALL) ระยะ early (ระยะเริ่มต้น)	11
ภาพที่ 2.3 ตัวอย่างภาพโรคมะเร็งเม็ดเลือดขาวชนิดเฉียบพลัน (ALL) ระยะ pre (ระยะก่อนแสดงอาการชัดเจน)	12
ภาพที่ 2.4 ตัวอย่างภาพโรคมะเร็งเม็ดเลือดขาวชนิดเฉียบพลัน (ALL) ระยะ pro (ระยะรุนแรง)	12
ภาพที่ 3.1 workflow การพัฒนา	13
ภาพที่ 3.2 การนำเข้าข้อมูล (1)	14
ภาพที่ 3.3 การนำเข้าข้อมูล (2)	14
ภาพที่ 3.4 การดึงคุณลักษณะจากภาพ (Feature Extraction) สำหรับโมเดล Deep Learning (Keras)	16
ภาพที่ 3.5 การดึงคุณลักษณะจากภาพ (Feature Extraction) สำหรับโมเดล Random Forest, MLP และ SVM	17
ภาพที่ 3.6 การสร้างโมเดล (Modeling) สำหรับโมเดล Deep Learning (Keras) (1)	20
ภาพที่ 3.7 การสร้างโมเดล (Modeling) สำหรับโมเดล Deep Learning (Keras) (2)	20
ภาพที่ 3.8 การสร้างโมเดล (Modeling) สำหรับโมเดล Deep Learning (Keras) (3)	20
ภาพที่ 3.9 การสร้างโมเดล (Modeling) สำหรับโมเดล Deep Learning (Keras) (4)	21
ภาพที่ 3.10 การสร้างโมเดล (Modeling) สำหรับโมเดล Random Forest	23
ภาพที่ 3.11 การสร้างโมเดล (Modeling) สำหรับโมเดล MLP	24
ภาพที่ 3.12 การสร้างโมเดล (Modeling) สำหรับโมเดล SVM	24
ภาพที่ 3.13 การประเมินผล (Evaluation & Visualization) สำหรับโมเดล Deep Learning (Keras)	25
ภาพที่ 3.14 Visualization สำหรับโมเดล Deep Learning (Keras)	26
ภาพที่ 3.15 การประเมินผล (Evaluation & Visualization) สำหรับโมเดล Random Forest	28
ภาพที่ 3.16 การประเมินผล (Evaluation & Visualization) สำหรับโมเดล MLP	29
ภาพที่ 3.17 การประเมินผล (Evaluation & Visualization) สำหรับโมเดล SVM	30
ภาพที่ 3.18 Visualization สำหรับโมเดล Random Forest, MLP และ SVM	31
ภาพที่ 3.19 การประเมินผล (Evaluation & Visualization) และเปรียบเทียบแต่ละ Model	34
ภาพที่ 4.1 Confusion Metrix จากการทดสอบโมเดล Deep Learning (Keras) ด้วย test set	36
ภาพที่ 4.2 Accuracy statistic จากการทดสอบโมเดลด้วย Deep Learning (Keras) test set	37
ภาพที่ 4.3 Confusion Metrix จากการทดสอบโมเดล Deep Learning (Keras) ด้วย validation set	37
ภาพที่ 4.4 Accuracy statistic จากการทดสอบโมเดลด้วย Deep Learning (Keras) ด้วย validation set	38

## สารบัญภาพ (ต่อ)

เรื่อง	หน้า
ภาพที่ 4.5 Confusion Metrix จากการทดสอบโมเดล Random Forest ด้วย test set	38
ภาพที่ 4.6 Accuracy statistic จากการทดสอบโมเดล Random Forest ด้วย test set	39
ภาพที่ 4.7 Confusion Metrix จากการทดสอบโมเดล Random Forest ด้วย validation set	39
ภาพที่ 4.8 Accuracy statistic จากการทดสอบโมเดล Random Forest ด้วย validation set	40
ภาพที่ 4.9 Confusion Metrix จากการทดสอบโมเดล MLP ด้วย test set	41
ภาพที่ 4.10 Accuracy statistic จากการทดสอบโมเดล MLP ด้วย test set	41
ภาพที่ 4.11 Confusion Metrix จากการทดสอบโมเดล MLP ด้วย validation set	42
ภาพที่ 4.12 Accuracy statistic จากการทดสอบโมเดล MLP ด้วย validation set	43
ภาพที่ 4.13 Confusion Metrix จากการทดสอบโมเดล SVM ด้วย test set	43
ภาพที่ 4.14 Accuracy statistic จากการทดสอบโมเดล SVM ด้วย test set	44
ภาพที่ 4.15 Confusion Metrix จากการทดสอบโมเดล SVM ด้วย validation set	45
ภาพที่ 4.16 Accuracy statistic จากการทดสอบโมเดล SVM ด้วย validation set	45
ภาพที่ 4.17 การเปรียบเทียบแต่ละ Model ด้วย Test set	46
ภาพที่ 4.18 การเปรียบเทียบแต่ละ Model ด้วย Validate set	47
ภาพที่ 4.19 การเปรียบเทียบ accuracy ของแต่ละ Model ด้วย test set และ Validate set	48

## สารบัญตาราง

เรื่อง	หน้า
ตารางที่ 2.1 ประเภทข้อมูล	12
ตารางที่ 3.1 คำอธิบายและการตั้งค่า node สำหรับการนำเข้าข้อมูลในภาพที่ 3.2 และภาพที่ 3.3	15
ตารางที่ 3.2 คำอธิบายและการตั้งค่า node สำหรับการดึงคุณลักษณะจากภาพ (Feature Extraction) ในภาพที่ 3.4 และภาพที่ 3.5	17
ตารางที่ 3.3 คำอธิบายและการตั้งค่า node สำหรับการสร้างโมเดล (Modeling) Deep Learning (Keras) ดังในภาพที่ 3.6 ถึงภาพที่ 3.89	21
ตารางที่ 3.4 คำอธิบายและการตั้งค่า node สำหรับการสร้างโมเดล (Modeling) Random Forest ดังในภาพที่ 3.910	23
ตารางที่ 3.5 คำอธิบายและการตั้งค่า node สำหรับการสร้างโมเดล (Modeling) MLP ดังในภาพที่ 3.11	24
ตารางที่ 3.6 คำอธิบายและการตั้งค่า node สำหรับการสร้างโมเดล (Modeling) SVM ดังในภาพที่ 3.12	25
ตารางที่ 3.7 คำอธิบายและการตั้งค่า node สำหรับการประเมินผล (Evaluation & Visualization) สำหรับโมเดล Deep Learning (Keras) ดังในภาพที่ 3.13 และ 3.14	26
ตารางที่ 3.8 คำอธิบายและการตั้งค่า node สำหรับการประเมินผล (Evaluation & Visualization) สำหรับโมเดล Random Forest ดังในภาพที่ 3.15	28
ตารางที่ 3.9 คำอธิบายและการตั้งค่า node สำหรับการประเมินผล (Evaluation & Visualization) สำหรับโมเดล MLP ดังในภาพที่ 3.16	29
ตารางที่ 3.10 คำอธิบายและการตั้งค่า node สำหรับการประเมินผล (Evaluation & Visualization) สำหรับโมเดล SVM ดังในภาพที่ 3.17	30
ตารางที่ 3.11 คำอธิบายและการตั้งค่า node สำหรับ Visualization สำหรับโมเดล Random Forest, MLP และ SVM ดังในภาพที่ 3.18	32
ตารางที่ 3.11 คำอธิบายและการตั้งค่า node การประเมินผล (Evaluation & Visualization) และเปรียบเทียบแต่ละ Model ดังในภาพที่ 3.19	34
ตารางที่ 5.1 ผลการเปรียบเทียบจากทั้งชุด Test set และ Validation set	50

## บทที่ 1

### บทนำ

#### 1.1 ที่มาและความสำคัญ

ในยุคปัจจุบัน การแพทย์และเทคโนโลยีได้พัฒนาอย่างรวดเร็ว โดยเฉพาะการนำเทคโนโลยีด้านปัญญาประดิษฐ์ (Artificial Intelligence - AI) เข้ามาช่วยในการวินิจฉัยและติดตามโรค เช่น โรคมะเร็งเม็ดเลือดขาวชนิดเฉียบพลันในเซลล์ลิมโฟยด์ (Acute Lymphoblastic Leukemia หรือ ALL) ซึ่งเป็นโรคที่สามารถพบได้บ่อยในเด็กและวัยรุ่น หากสามารถตรวจพบได้ในระยะแรกเริ่ม จะช่วยเพิ่มโอกาสในการรักษาให้ประสบความสำเร็จได้อย่างมาก

จากรายงานของ World Health Organization (WHO) และ Leukemia & Lymphoma Society (LLS) ระบุว่า ALL เป็นมะเร็งเม็ดเลือดที่พบได้บ่อยที่สุดในเด็ก โดยเฉพาะในช่วงอายุ 2–5 ปี โดยในสหรัฐอเมริกา พบผู้ป่วยใหม่ ประมาณ 6,500 รายต่อปี และในจำนวนนี้กว่า 75% เป็นเด็กอายุต่ำกว่า 15 ปี (LLS, 2022)

สำหรับประเทศไทย ข้อมูลจาก กระทรวงสาธารณสุข (สถาบันมะเร็งแห่งชาติ, 2565) พบว่าโรคมะเร็งเม็ดเลือดขาวอยู่ในอันดับต้น ๆ ของมะเร็งที่พบบ่อยในเด็ก และมักได้รับการวินิจฉัยล่าช้าเนื่องจากอาการไม่เฉพาะเจาะจง เช่น เหนื่อยง่าย ซีด เลือดออกง่าย หรือมีไข้เรื้อรัง ซึ่งอาจคล้ายกับโรคอื่น ๆ

อย่างไรก็ตาม การวิเคราะห์ภาพเซลล์เม็ดเลือดจากกล้องจุลทรรศน์จำเป็นต้องใช้ความชำนาญของแพทย์ผู้เชี่ยวชาญ อีกทั้งยังใช้เวลานานและมีความเสี่ยงจากความผิดพลาดของมนุษย์ โดยเฉพาะอย่างยิ่งในโรงพยาบาลหรือคลินิกที่มีบุคลากรไม่เพียงพอ

ดังนั้น การพัฒนาโมเดลที่สามารถตรวจจับและจำแนกชนิดของ ALL ได้โดยอัตโนมัติผ่านภาพถ่ายเซลล์ จึงเป็นแนวทางที่สำคัญในการช่วยลดภาระของแพทย์ และเพิ่มประสิทธิภาพในการวินิจฉัย อีกทั้งยังสามารถเป็นต้นแบบของระบบช่วยวินิจฉัยอัจฉริยะที่มีศักยภาพในการนำไปใช้งานจริงในระดับโรงพยาบาลทั่วประเทศ

#### 1.2 วัตถุประสงค์ของรายงาน

- 1.2.1 เพื่อพัฒนาระบบ Workflow บน KNIME สำหรับจำแนกภาพเซลล์มะเร็งเม็ดเลือดขาว ALL ออกเป็น 4 ประเภท ได้แก่ benign, early, pre และ pro
- 1.2.2 เพื่อเปรียบเทียบประสิทธิภาพของโมเดล Machine Learning ได้แก่ Keras (Deep Learning), Random Forest, MLP และ SVM
- 1.2.3 เพื่อแสดงผลการวิเคราะห์ด้วย Visualization ช่วยให้เข้าใจผลลัพธ์โมเดลได้ง่าย

#### 1.3 ขอบเขต

- 1.3.1 ใช้ชุดข้อมูลภาพถ่ายกล้องจุลทรรศน์ของเซลล์เม็ดเลือดที่เกี่ยวข้องกับโรค ALL จาก <https://www.kaggle.com/datasets/obulisainaren/multi->



cancer?fbclid=IwY2xjawJJrvlleHRuA2FlbQlxMAABHZGh88wiBHileCnZQzomKNDLT4y-j-ZtV5OV4G7ioWlj\_D8j3fWXpap2og\_aem\_BV7ljAz1hrzCjfvsnGW\_w ประเภทละ 1,000 รูป

- 1.3.2 พัฒนา Workflow ใน KNIME โดยเน้นการใช้ Node สำเร็จรูป
- 1.3.3 ทดสอบการทำงานด้วยโมเดล 4 รูปแบบ: Deep Learning (Keras), Random Forest, MLP และ SVM
- 1.3.4 เปรียบเทียบผลลัพธ์ด้วย Confusion Matrix, Accuracy, Precision, Recall, และ F1-score
- 1.3.5 ไม่ใช้เทคนิค Data Augmentation เพื่อให้เน้นการพัฒนาโมเดลบนชุดข้อมูลดิบ

#### 1.4 ประโยชน์ที่คาดว่าจะได้รับ

- 1.4.1 เพิ่มความแม่นยำในการจำแนกชนิดของ ALL และช่วยในการวินิจฉัยเบื้องต้น
- 1.4.2 เพิ่มโอกาสในการตรวจวินิจฉัยโรคได้รวดเร็ว
- 1.4.3 พัฒนาทักษะด้านการวิเคราะห์ข้อมูลภาพ (Image Processing) และการใช้งาน KNIME

## บทที่ 2

### ชุดข้อมูล (Dataset)

#### 2.1 แหล่งที่มาของข้อมูล

ชุดข้อมูลที่ใช้ในโครงการนี้นำมาจากชุดข้อมูลแบบเปิด (Open Dataset) บนแพลตฟอร์ม Kaggle ชื่อว่า "Multi Cancer Dataset" ซึ่งเป็นชุดข้อมูลที่รวบรวมภาพกล้องจุลทรรศน์ของเซลล์มะเร็งหลายประเภท โดยในโครงการนี้ได้เลือกใช้เฉพาะส่วนที่เกี่ยวข้องกับ Acute Lymphoblastic Leukemia (ALL) เท่านั้น

ALL ถูกแบ่งออกเป็น 4 หมวดย่อย ได้แก่:

- all\_benign: เซลล์ปกติที่ไม่เป็นมะเร็ง
- all\_early: เซลล์ในระยะแรกของการเปลี่ยนแปลงเป็นมะเร็ง
- all\_pre: เซลล์ที่มีความผิดปกติก่อนเข้าสู่ภาวะรุนแรง
- all\_pro: เซลล์ที่เข้าสู่ระยะรุนแรงของโรค

สำหรับการทดลองในโครงการนี้ ได้เลือกใช้ 1,000 ภาพแรกจากแต่ละคลาสย่อย รวมเป็นทั้งหมด 4,000 ภาพ เพื่อใช้ในการเทรนและประเมินผลโมเดล

#### 2.2 ลักษณะและเนื้อหาของข้อมูล

ข้อมูลที่ใช้ในโครงการนี้เป็นข้อมูลภาพถ่ายจากกล้องจุลทรรศน์ของเซลล์เม็ดเลือดขาวชนิดลิมโฟบลาสต์ ซึ่งเป็นเซลล์ต้นกำเนิดในไขกระดูกที่มีบทบาทสำคัญในการสร้างภูมิคุ้มกันของร่างกาย ในกรณีของโรคมะเร็งเม็ดเลือดขาวชนิดเฉียบพลัน (Acute Lymphoblastic Leukemia: ALL) เซลล์เหล่านี้จะมีการแบ่งตัวผิดปกติ ส่งผลให้เกิดการสะสมของเซลล์ผิดปกติในไขกระดูกและกระแสเลือด ซึ่งอาจนำไปสู่การรบกวนการทำงานของอวัยวะภายในต่าง ๆ

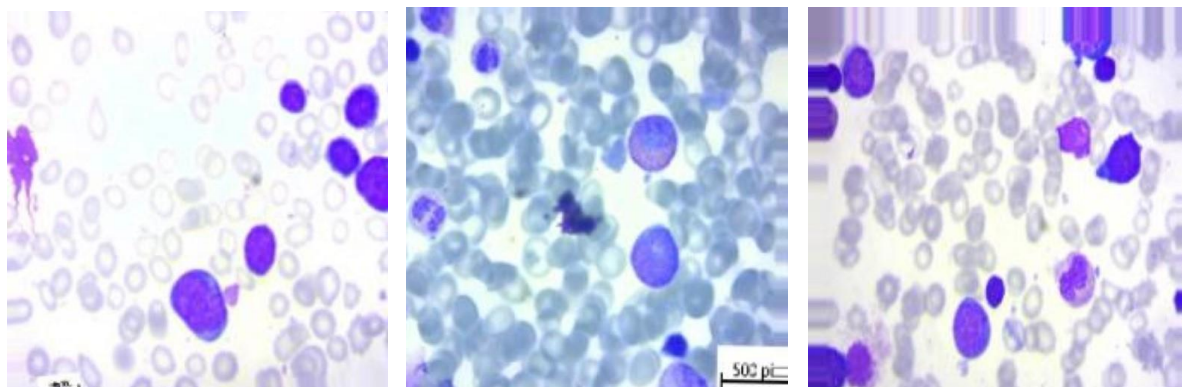
ภาพทั้งหมดในชุดข้อมูลนี้ได้รับการจัดเตรียมขึ้นในห้องปฏิบัติการไขกระดูกของโรงพยาบาล Taleqani กรุงเตหะราน ประเทศอิหร่าน โดยมาจากผู้ป่วยที่สงสัยว่าจะเป็นโรค ALL จำนวน 89 ราย รวมทั้งสิ้น 3,256 ภาพ จากฟิล์มเลือด (Peripheral Blood Smear: PBS) ภาพเหล่านี้ถูกจัดเตรียมและย้อมสีโดยเจ้าหน้าที่ห้องแล็บที่มีความชำนาญ

กล้องที่ใช้สำหรับการถ่ายภาพเป็นกล้อง Zeiss ซึ่งติดตั้งกับกล้องจุลทรรศน์ที่มีกำลังขยาย 100 เท่า โดยบันทึกภาพในรูปแบบไฟล์ .JPG ภาพที่ได้มีความละเอียดสูง สามารถเห็นรายละเอียดของนิวเคลียสและไซโตพลาสซึมได้อย่างชัดเจน ซึ่งเป็นจุดสำคัญในการแยกแยะชนิดของเซลล์เม็ดเลือด

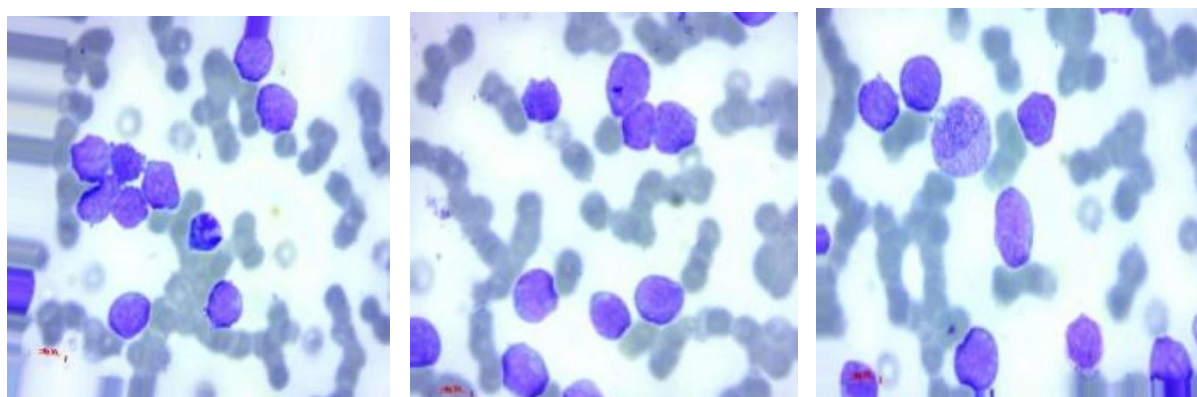
ข้อมูลถูกจำแนกออกเป็น สองกลุ่มใหญ่ คือกลุ่ม benign (ไม่เป็นมะเร็ง) ซึ่งเป็นเซลล์ปกติประเภท hematogones และ กลุ่ม malignant (มะเร็ง) ซึ่งประกอบด้วยเซลล์ในสามระยะ ได้แก่ Early Pre-B, Pre-B และ Pro-B ALL โดยแพทย์ผู้เชี่ยวชาญได้ทำการระบุชนิดและชนิดย่อยของเซลล์เหล่านี้อย่างแม่นยำโดยใช้เทคโนโลยี Flow Cytometry

นอกจากนี้ ภาพบางส่วนในชุดข้อมูลยังได้ผ่านกระบวนการแยกเซลล์ออกจากฉากหลัง (segmentation) ด้วยวิธีการ thresholding บนพื้นที่สี HSV เพื่อให้ได้ภาพที่สามารถนำไปใช้งานต่อในเชิงเทคนิค เช่น การวิเคราะห์ feature ของเซลล์ได้ง่ายขึ้น

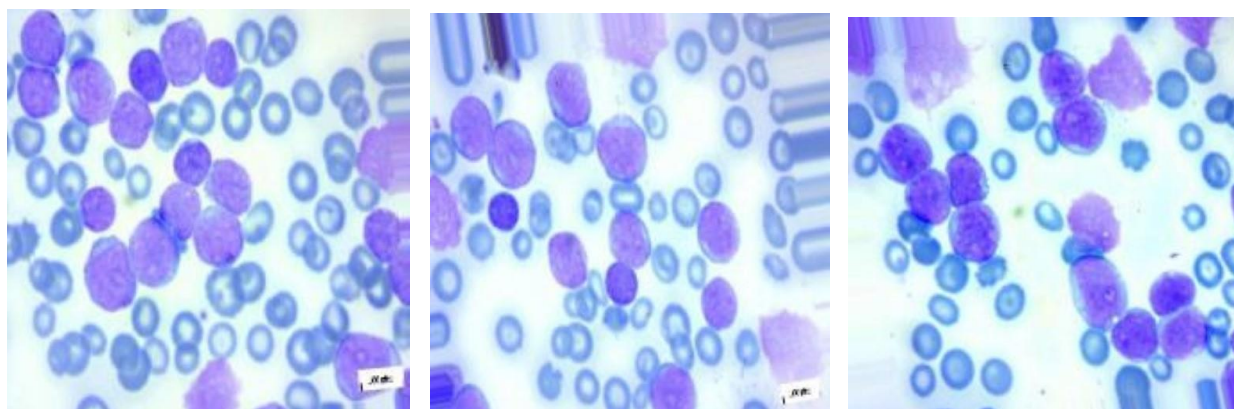
การใช้ข้อมูลภาพเหล่านี้ทำให้สามารถพัฒนาโมเดล AI ที่เรียนรู้จากลักษณะทางกายภาพของเซลล์โดยตรง ซึ่งไม่จำเป็นต้องใช้การวิเคราะห์ในระดับโมเลกุล ช่วยลดต้นทุนในการวินิจฉัย และเพิ่มศักยภาพในการนำไปใช้จริงในสภาพแวดล้อมทางคลินิก



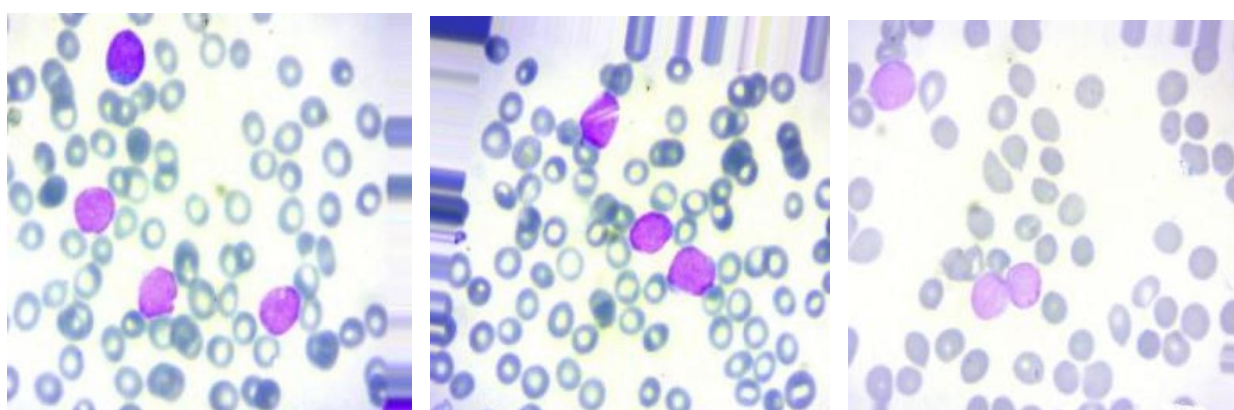
ภาพที่ 2.1 ตัวอย่างภาพโรคมะเร็งเม็ดเลือดขาวชนิดเฉียบพลัน (ALL) ระยะ benign (ไม่เป็นมะเร็ง)



ภาพที่ 2.2 ตัวอย่างภาพโรคมะเร็งเม็ดเลือดขาวชนิดเฉียบพลัน (ALL) ระยะ early (ระยะเริ่มต้น)



ภาพที่ 2.3 ตัวอย่างภาพโรคมะเร็งเม็ดเลือดขาวชนิดเฉียบพลัน (ALL) ระยะ pre (ระยะก่อนแสดงอาการชัดเจน)



ภาพที่ 2.4 ตัวอย่างภาพโรคมะเร็งเม็ดเลือดขาวชนิดเฉียบพลัน (ALL) ระยะ pro (ระยะรุนแรง)

## 2.3 ประเภทของข้อมูล

### ตารางที่ 2.1 ประเภทข้อมูล

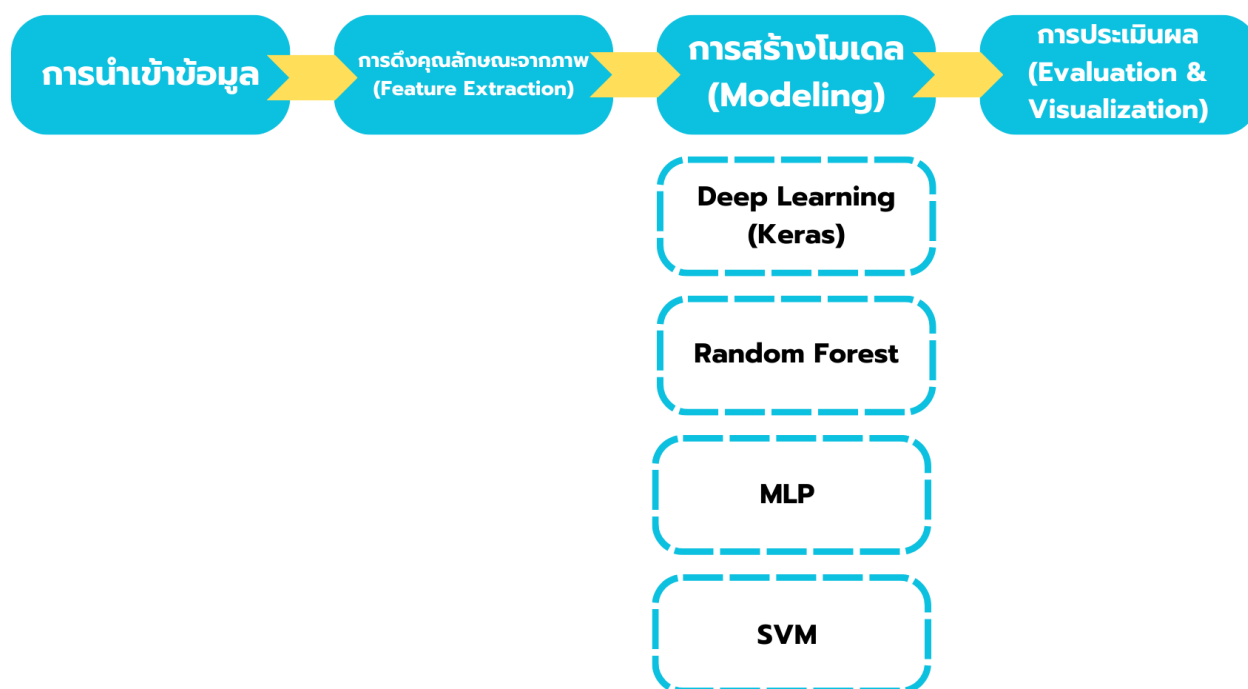
ประเภทข้อมูล	รายละเอียด
ประเภทไฟล์	รูปภาพ (JPEG)
จำนวนคลาส	4 คลาส (benign, early, pre, pro)
จำนวนภาพต่อคลาส	1,000 ภาพต่อคลาส รวมทั้งหมด 4,000 ภาพ
รูปแบบข้อมูล	ข้อมูลภาพ (Image Data) แบ่งโฟลเดอร์ตามคลาส
ลักษณะภาพ	ภาพจากกล้องจุลทรรศน์ ขนาดไม่เท่ากัน ถูก Resize ใน KNIME ให้เท่ากัน (150x150 พิกเซล)

### บทที่ 3

#### ขั้นตอนการพัฒนา

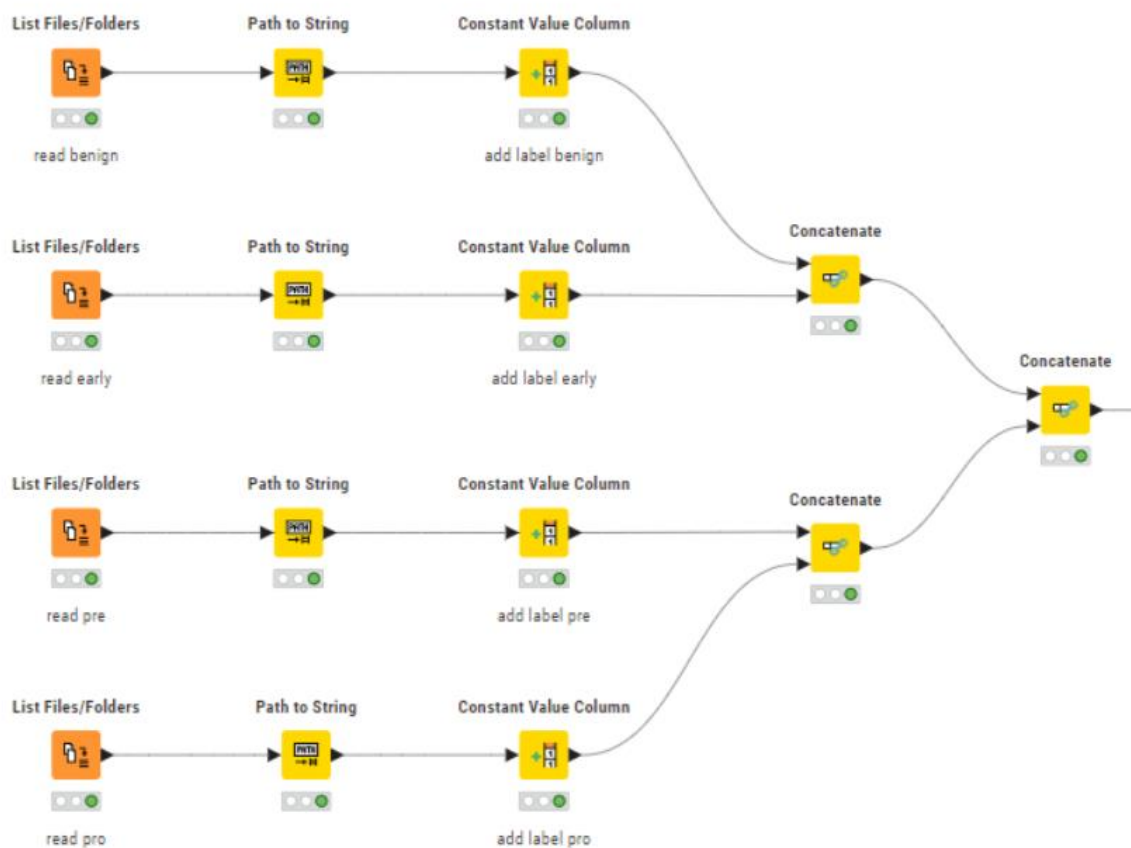
ในโครงการนี้ดำเนินการพัฒนาในรูปแบบของ Workflow โดยใช้เครื่องมือ KNIME Analytics Platform โดยใน workflow มีการแบ่งขั้นตอนออกเป็น 4 ส่วนหลัก (ดังในภาพที่ 3.1) ได้แก่

- 3.1 การนำเข้าข้อมูล
- 3.2 การดึงคุณลักษณะจากภาพ (Feature Extraction)
- 3.3 การสร้างโมเดล (Modeling)
- 3.4 การประเมินผล (Evaluation & Visualization)

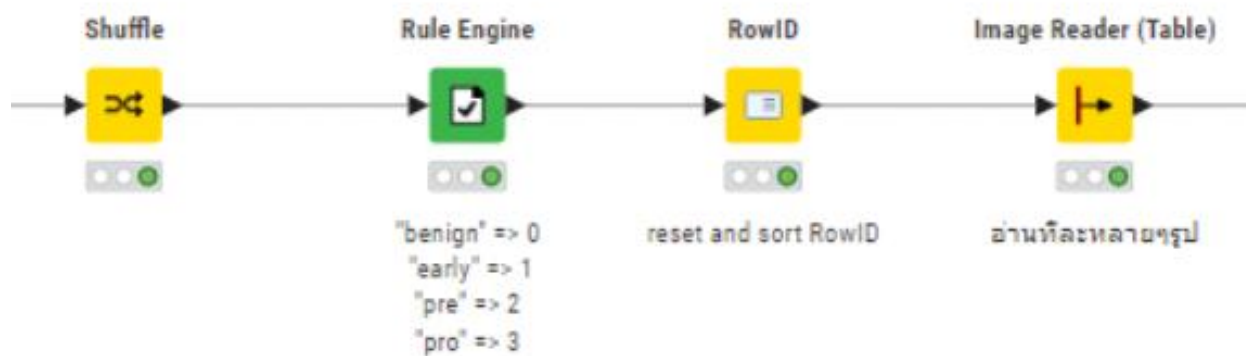


ภาพที่ 3.1 workflow การพัฒนา

### 3.1 การนำเข้าข้อมูล



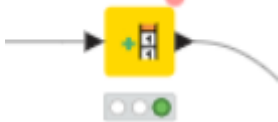





ภาพที่ 3.2 การนำเข้าข้อมูล (1)


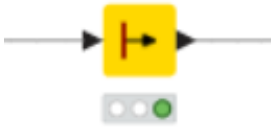


ภาพที่ 3.3 การนำเข้าข้อมูล (2)

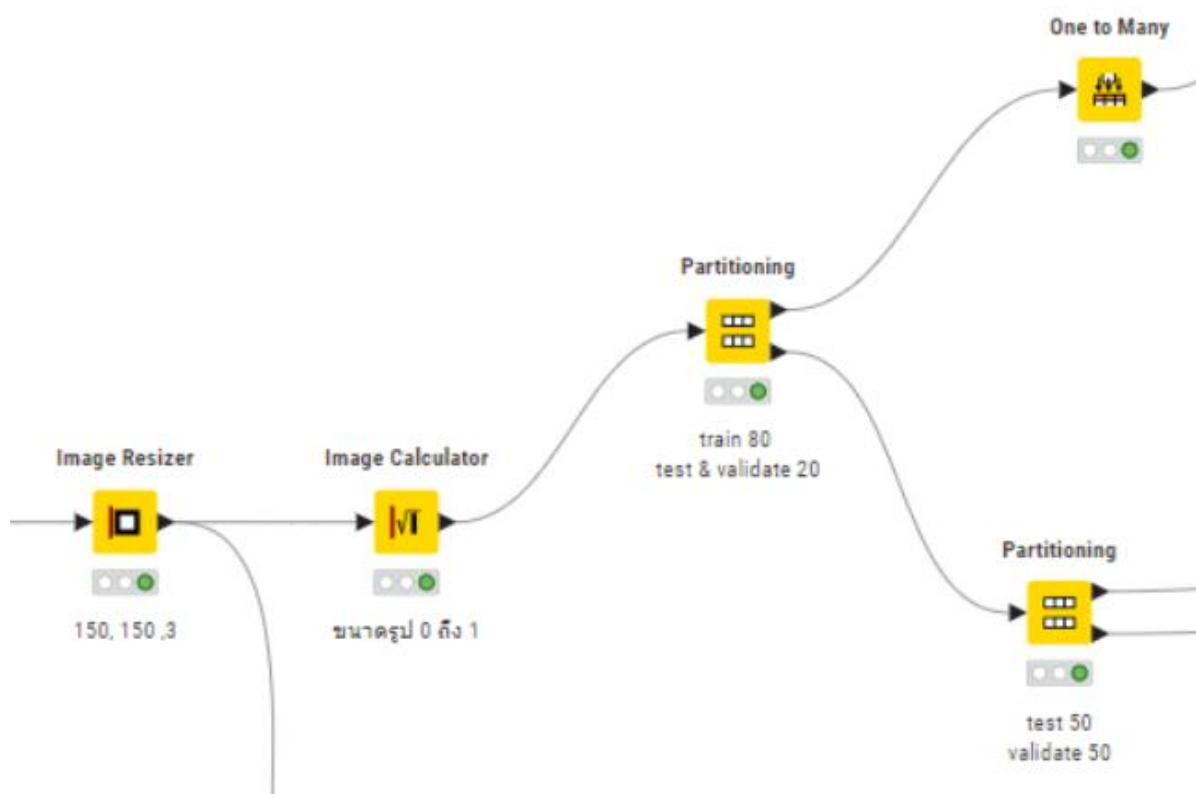
ตารางที่ 3.1 คำอธิบายและการตั้งค่า node สำหรับการนำเข้าข้อมูลในภาพที่ 3.2 และภาพที่ 3.3

ชื่อ	node	คำอธิบาย	การตั้งค่า
List Files/Folders		<ul style="list-style-type: none"> <li>- แสดงรายการไฟล์และ/หรือโฟลเดอร์ที่อยู่ในโฟลเดอร์ที่เลือกไว้</li> <li>- เข้าถึงระบบไฟล์ได้หลากหลายประเภท</li> </ul>	<ul style="list-style-type: none"> <li>- Read from : Local File System</li> <li>- Mode : Files in Folder</li> <li>- Folder : Path to all_benign, all_early, all_pre, all_pro</li> </ul>
Path to String		<ul style="list-style-type: none"> <li>- แปลงคอลัมน์ประเภท path ให้เป็นคอลัมน์ประเภท string</li> </ul>	<ul style="list-style-type: none"> <li>- Column selection : Path</li> <li>- Output : Replace selected column</li> </ul>
Constant Value Column		<ul style="list-style-type: none"> <li>- เพิ่มหรือแทนที่คอลัมน์โดยใส่ค่าคงที่เพียงค่าเดียว (Label) ในทุกแถว</li> </ul>	<ul style="list-style-type: none"> <li>- Append : ALL_category</li> <li>- Value settings : benign, early pre, pro (String)</li> </ul>
Concatenate		<ul style="list-style-type: none"> <li>- รวม (concatenate) ตารางตั้งแต่สองตารางขึ้นไปเข้าด้วยกัน</li> </ul>	<ul style="list-style-type: none"> <li>- Union</li> <li>- Create new</li> </ul>
Shuffle		<ul style="list-style-type: none"> <li>- สับเปลี่ยนลำดับของแถวในตารางให้เป็นแบบสุ่ม</li> <li>- ลด bias ที่อาจเกิดจากลำดับข้อมูลที่จัดเรียงอยู่แล้ว</li> </ul>	<ul style="list-style-type: none"> <li>- Use seed</li> </ul>
Rule Engine		<ul style="list-style-type: none"> <li>- กำหนดเงื่อนไข (if-else) ด้วยกฎที่เขียนเองแล้วเพิ่มคอลัมน์ผลลัพธ์ตามเงื่อนไข</li> </ul>	<pre>\$ALL_category\$ = "benign" =&gt; 0 \$ALL_category\$ = "early" =&gt; 1 \$ALL_category\$ = "pre" =&gt; 2 TRUE =&gt; 3</pre> <ul style="list-style-type: none"> <li>- Append Column : ALL_category_LE</li> </ul>



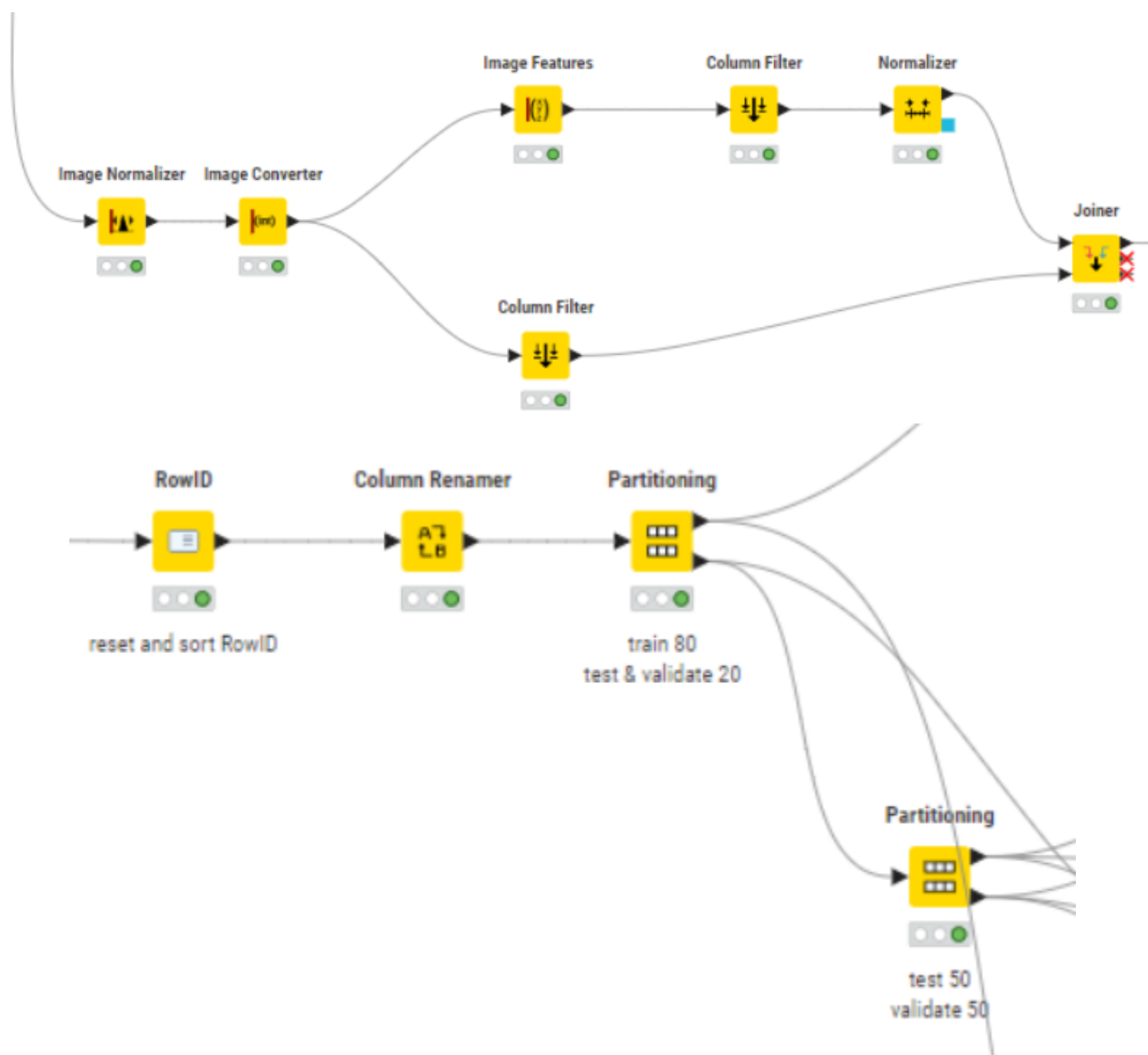
RowID		- แทนที่ RowID เดิม ด้วย RowID ใหม่	- Replace RowID - Generate new
Image Reader (Table)		- ใช้สำหรับ โหลดภาพ (image files) จากไฟล์ บนเครื่อง	Options - Input Column : Path Column Setting : Replace

### 3.2 การดึงคุณลักษณะจากภาพ (Feature Extraction)




ภาพที่ 3.4 การดึงคุณลักษณะจากภาพ (Feature Extraction) สำหรับโมเดล Deep Learning (Keras)

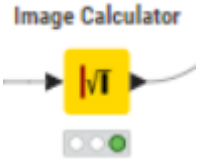
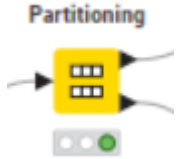
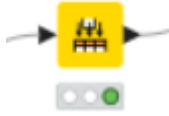

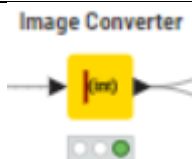




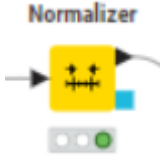
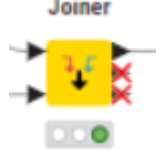
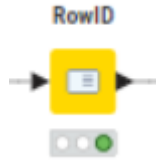



ภาพที่ 3.5 การดึงคุณลักษณะจากภาพ (Feature Extraction) สำหรับโมเดล Random Forest, MLP และ SVM

ตารางที่ 3.2 คำอธิบายและการตั้งค่า node สำหรับการดึงคุณลักษณะจากภาพ (Feature Extraction) ในภาพที่ 3.4 และภาพที่ 3.5

ชื่อ	node	คำอธิบาย	การตั้งค่า
Image Resizer		- ปรับขนาดของภาพให้เท่ากัน	<ul style="list-style-type: none"> <li>- ใช้ทั้งโมเดล Deep Learning (Keras), Random Forest, MLP และ SVM</li> <li>- Linear Interpolation</li> <li>- All Demensions Manual</li> <li>- X = 150.0, Y = 150.0, Z = 1.0, Channel = 3.0, Time = 1.0</li> </ul>

			- Absolute Image Size
Image Calculator		- ใช้สำหรับเขียนสมการกับค่าที่ได้จากภาพ เช่น การทำ Normalize ให้เหลือ 0 ถึง 1	<ul style="list-style-type: none"> <li>- \$Image\$/255.0</li> <li>- Append Column Img Calc</li> <li>- Result pixel type : UNSIGNEDBYTE</li> </ul>
Partitioning		- แบ่งข้อมูลออกเป็น 2 ชุดออกเป็น Training set และ Testing set หรือ Validating set	<ul style="list-style-type: none"> <li>- ใช้ทั้งโมเดล Deep Learning (Keras), Random Forest, MLP และ SVM</li> <li>- Relative[%]</li> <li>- train 80, test &amp; validate 20</li> <li>- test 50, validate 50</li> <li>- Stratified sampling : ALL_category/Target</li> <li>- Use random seed</li> </ul>
One to Many		- ใช้สำหรับทำ One-Hot Encoding โดยแปลงค่าประเภทให้เป็นคอลัมน์ตัวเลขหลายคอลัมน์ ที่มีค่า 1 หรือ 0	- Include ALL_category
Image Normalizer		- ปรับภาพให้เป็นแบบ normalized	Options <ul style="list-style-type: none"> <li>- Enhancement Type : NORMALIZE</li> <li>- X, Y, Channel</li> </ul> Column Selection <ul style="list-style-type: none"> <li>- Column Creation Mode : Append</li> <li>- Column suffix : _in</li> <li>- Include Image</li> </ul>
Image Converter		- แปลงประเภทของพิกเซลไปเป็นอีกชนิดหนึ่ง	Options <ul style="list-style-type: none"> <li>- Target type : FLOATTYPE</li> <li>- Conversion method : Copy</li> <li>- Factory Type : CELL_IMG_FACTORY</li> </ul> Column Selection

			<ul style="list-style-type: none"> <li>- Column Creation Mode : Append</li> <li>- Column suffix : _converted</li> <li>- Include Image_in</li> <li>- Exclude Image</li> </ul>
Image Features		<ul style="list-style-type: none"> <li>- คำนวณคุณลักษณะ (features) จากทั้งภาพโดยรวม</li> </ul>	Column Selection <ul style="list-style-type: none"> <li>- Image : Image_in_converted Features</li> <li>- First order statistics</li> </ul>
Column Filter		<ul style="list-style-type: none"> <li>- เลือกเฉพาะบางคอลัมน์ที่ต้องการเก็บไว้ และ กรอง (ลบ) คอลัมน์อื่นๆ ออกไป</li> </ul>	<div> <b>เส้นบน</b> <ul style="list-style-type: none"> <li>Exclude</li> <li>- Min</li> <li>- Max</li> <li>- Geometric Mean</li> <li>- Quantil 25</li> <li>- Median absolute deviation (MAD)</li> <li>- WeightCentroid Dim 4</li> <li>- WeightCentroid Dim 5</li> <li>- Mass Displacement</li> </ul> </div> <hr/> <div> <b>เส้นล่าง</b> <ul style="list-style-type: none"> <li>Include</li> <li>- ALL_category</li> </ul> </div>
Normalizer		<ul style="list-style-type: none"> <li>- ทำ Normalization สำหรับข้อมูลที่เป็นตัวเลข</li> </ul>	<ul style="list-style-type: none"> <li>- Min-max</li> <li>- Min : 0</li> <li>- Max :1</li> </ul>
Joiner		<ul style="list-style-type: none"> <li>- รวมสองตารางเข้าด้วยกัน คล้ายกับการทำ Join ในฐานข้อมูล</li> </ul>	<ul style="list-style-type: none"> <li>- All of the following</li> <li>- left &amp; right table : RowID</li> <li>- Compare by Value and type</li> <li>- Matching rows</li> </ul>
RowID		<ul style="list-style-type: none"> <li>- แทนที่ RowID เดิม ด้วย RowID ใหม่</li> </ul>	<ul style="list-style-type: none"> <li>- Replace RowID</li> <li>- Generate new</li> </ul>

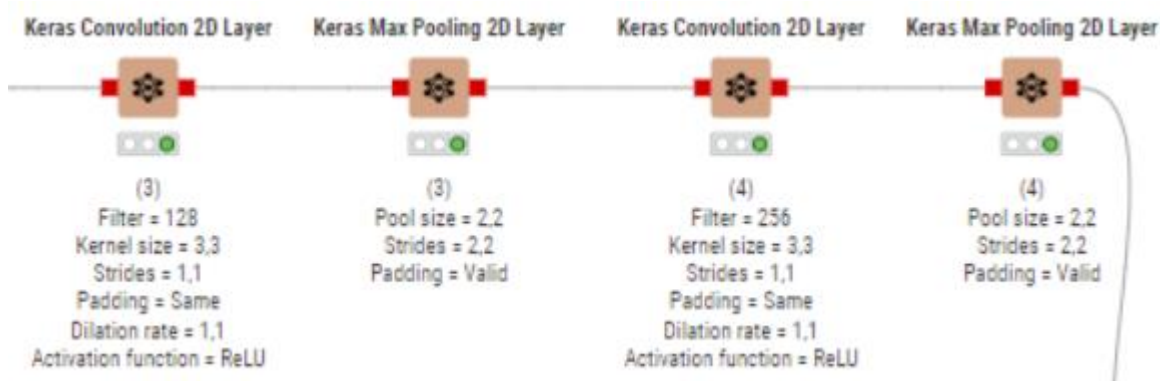
Column Renamer		- เปลี่ยนชื่อคอลัมน์ในตาราง	- ALL_category → target
----------------	---	-----------------------------	-------------------------

### 3.3 การสร้างโมเดล (Modeling)

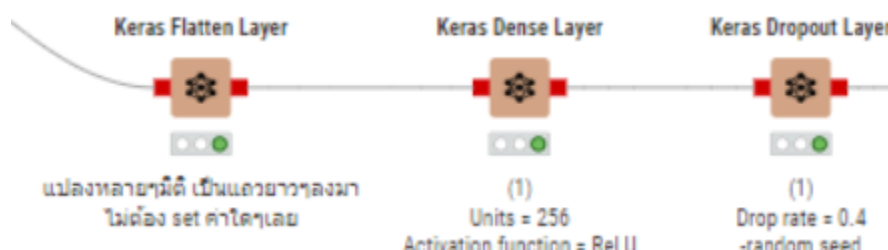
#### 3.3.1 Deep Learning (Keras)



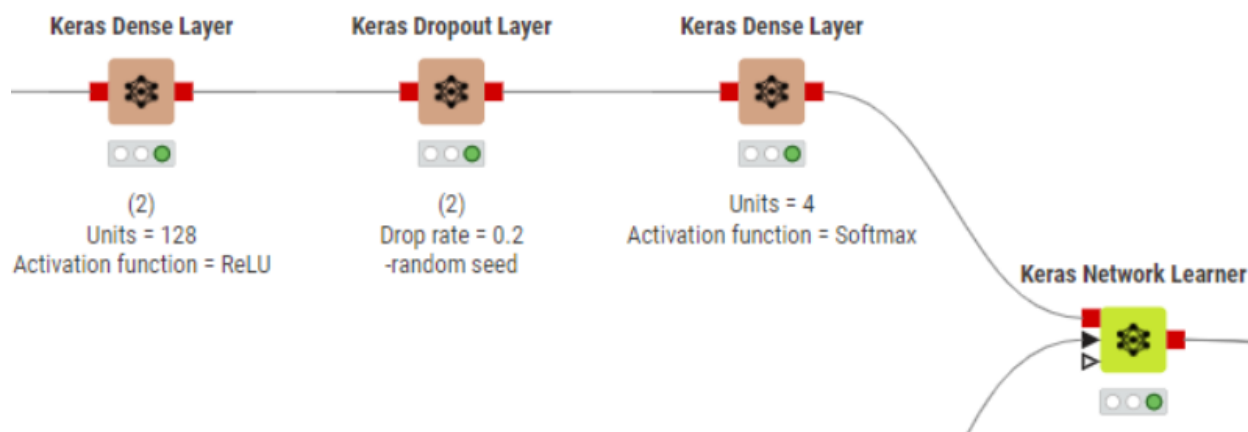
ภาพที่ 3.6 การสร้างโมเดล (Modeling) สำหรับโมเดล Deep Learning (Keras) (1)



ภาพที่ 3.7 การสร้างโมเดล (Modeling) สำหรับโมเดล Deep Learning (Keras) (2)

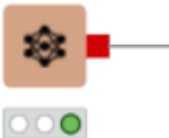
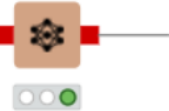


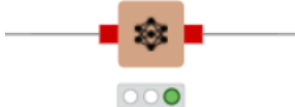



ภาพที่ 3.8 การสร้างโมเดล (Modeling) สำหรับโมเดล Deep Learning (Keras) (3)

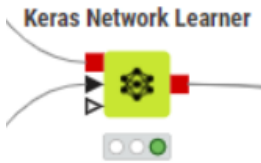


ภาพที่ 3.9 การสร้างโมเดล (Modeling) สำหรับโมเดล Deep Learning (Keras) (4)

ตารางที่ 3.3 คำอธิบายและการตั้งค่า node สำหรับการสร้างโมเดล (Modeling) Deep Learning (Keras) ตั้งในภาพที่ 3.6 ถึงภาพที่ 3.89

ชื่อ	node	คำอธิบาย	การตั้งค่า
Keras Input Layer	<b>Keras Input Layer</b> 	- เริ่มต้นสร้างโมเดล สร้าง Input Layer	- Shape = 150, 150, 3 - Data type = Float 32 - Data Format = Channel Last
Keras Convolution 2D Layer	<b>Keras Convolution 2D Layer</b> 	- สร้างคอนโวลูชันคอร์เนล (convolution kernel)	<div>(1)</div> <div>- Filter = 32 - Kernel size = 3,3 - Strides = 1,1 - Padding = Same - Dilation rate = 1,1 - Activation function = ReLU</div> <hr/> <div>(2)</div> <div>- Filter = 64 - Kernel size = 3,3 - Strides = 1,1 - Padding = Same - Dilation rate = 1,1 - Activation function = ReLU</div> <hr/> <div>(3)</div> <div>- Filter = 128</div>

			<ul style="list-style-type: none"> <li>- Kernel size = 3,3</li> <li>- Strides = 1,1</li> <li>- Padding = Same</li> <li>- Dilation rate = 1,1</li> <li>- Activation function = ReLU</li> </ul> <hr/> <p>(4)</p> <ul style="list-style-type: none"> <li>- Filter = 256</li> <li>- Kernel size = 3,3</li> <li>- Strides = 1,1</li> <li>- Padding = Same</li> <li>- Dilation rate = 1,1</li> <li>- Activation function = ReLU</li> </ul>
Keras Max Pooling 2D Layer	 <p>Keras Max Pooling 2D Layer</p>	- ใช้สำหรับทำ Max Pooling	<p>(1), (2), (3)</p> <ul style="list-style-type: none"> <li>- Pool size = 2,2</li> <li>- Strides = 2,2</li> <li>- Padding = Valid</li> </ul>
Keras Flatten Layer	 <p>Keras Flatten Layer</p>	- แปลงข้อมูลอินพุตให้เป็นเวกเตอร์ 1 มิติ	- ไม่มีการตั้งค่า
Keras Dense Layer	 <p>Keras Dense Layer</p>	- ทำการคำนวณชั้นสุดท้าย	<p>(1)</p> <ul style="list-style-type: none"> <li>- Units = 256</li> <li>- Activation function = ReLU</li> </ul> <hr/> <p>(2)</p> <ul style="list-style-type: none"> <li>- Units = 128</li> <li>- Activation function = ReLU</li> </ul> <hr/> <ul style="list-style-type: none"> <li>- Units = 4</li> <li>- Activation function = Softmax</li> </ul>
Keras Dropout Layer	 <p>Keras Dropout Layer</p>	<ul style="list-style-type: none"> <li>- สุ่มปิดหน่วยอินพุตบางส่วน (เป็น 0) ตามอัตราที่กำหนด</li> <li>- ลด overfitting)</li> </ul>	<p>(1)</p> <ul style="list-style-type: none"> <li>- Drop rate = 0.4</li> <li>- random seed</li> </ul> <hr/> <p>(2)</p> <ul style="list-style-type: none"> <li>- Drop rate = 0.2</li> <li>- random seed</li> </ul>

Keras Network Learner		<ul style="list-style-type: none"> <li>- ฝึกโมเดล Deep Learning ที่สร้างจาก โหนด Keras Network ด้วยข้อมูลที่มี label (Supervised)</li> </ul>	Input Data <ul style="list-style-type: none"> <li>- Conversion : From Image</li> <li>- Include Image Calc</li> </ul> Target Data <ul style="list-style-type: none"> <li>- Conversion : From Number (doubler)</li> <li>- Include : benign, early, pre, pro</li> <li>- Standard loss function</li> <li>- Binary cross entropy</li> </ul> Option <ul style="list-style-type: none"> <li>- Epochs = 150</li> <li>- Training batch size = 64</li> <li>- Validation batch size = 64</li> <li>- Optimizer : Adam</li> <li>- Learning rate 0.0001</li> </ul>
-----------------------------	---	--	--

### 3.3.2 Random Forest

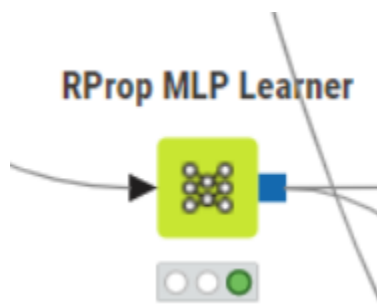


ภาพที่ 3.10 การสร้างโมเดล (Modeling) สำหรับโมเดล Random Forest

ตารางที่ 3.4 คำอธิบายและการตั้งค่า node สำหรับการสร้างโมเดล (Modeling) Random Forest ดังในภาพที่ 3.910


ชื่อ	node	คำอธิบาย	การตั้งค่า
Random Forest Learner		<ul style="list-style-type: none"> <li>- ฝึกโมเดล Random Forest</li> </ul>	<ul style="list-style-type: none"> <li>- Target Column : Target</li> <li>- Gini Index</li> <li>- Minimum node size = 1</li> <li>- Number of model = 400</li> <li>- Use random seed</li> </ul>

### 3.3.3 MLP

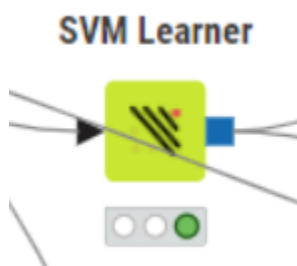


ภาพที่ 3.11 การสร้างโมเดล (Modeling) สำหรับโมเดล MLP

ตารางที่ 3.5 คำอธิบายและการตั้งค่า node สำหรับการสร้างโมเดล (Modeling) MLP ดังในภาพที่ 3.11

ชื่อ	node	คำอธิบาย	การตั้งค่า
RProp MLP Learner		- ฝึก Neural Network แบบ Multilayer Feedforward โดยใช้ อัลกอริธึม RProp	<ul style="list-style-type: none"> <li>- Maximum number of iterations = 1000</li> <li>- Number of hidden layer = 2</li> <li>- Number of hidden neuron per layer = 64</li> <li>- class column : target</li> <li>- use seed</li> </ul>

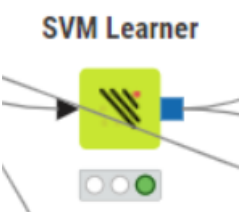
### 3.3.4 SVM



ภาพที่ 3.12 การสร้างโมเดล (Modeling) สำหรับโมเดล SVM

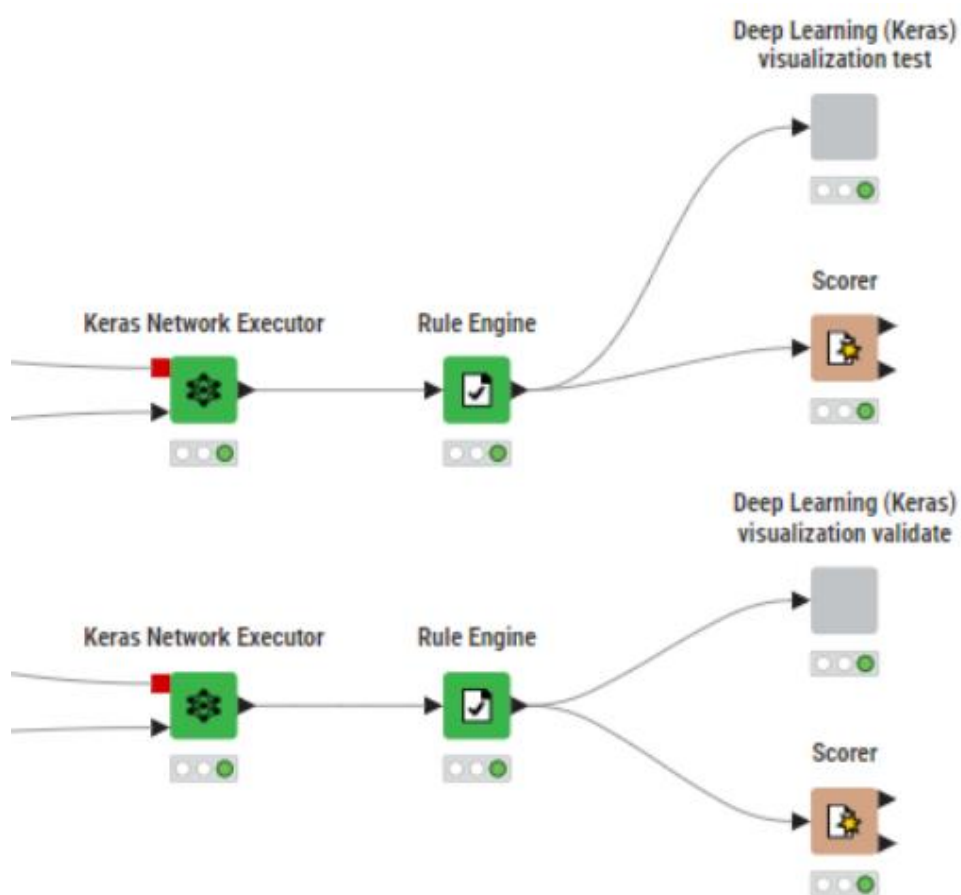


ตารางที่ 3.6 คำอธิบายและการตั้งค่า node สำหรับการสร้างโมเดล (Modeling) SVM ดังในภาพที่ 3.12

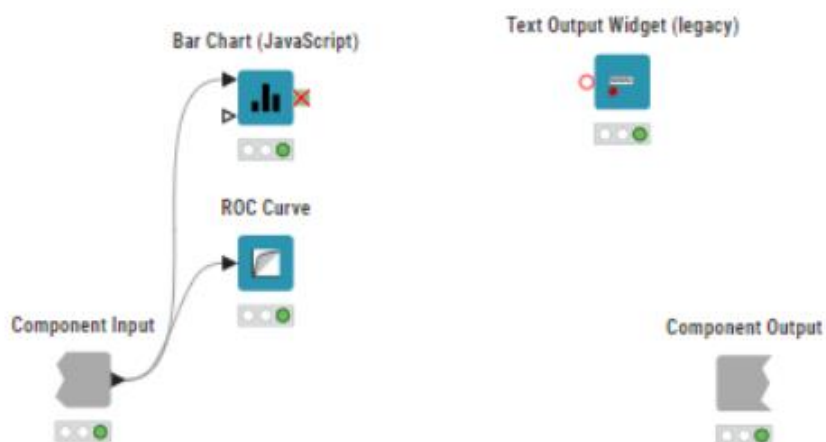
ชื่อ	node	คำอธิบาย	การตั้งค่า
SVM Learner		- ใช้สำหรับฝึกโมเดล Support Vector Machine (SVM) ซึ่งเป็นโมเดล Supervised Learning	- Class column : target - Overlapping penalty = 1.0 - RBF sigma = 0.1

### 3.4 การประเมินผล (Evaluation & Visualization)

#### 3.4.1 Deep Learning (Keras)

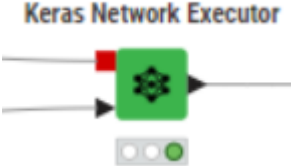
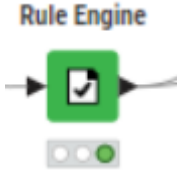


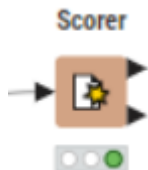
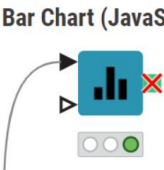
ภาพที่ 3.13 การประเมินผล (Evaluation & Visualization) สำหรับโมเดล Deep Learning (Keras)




ภาพที่ 3.14 Visualization สำหรับโมเดล Deep Learning (Keras)

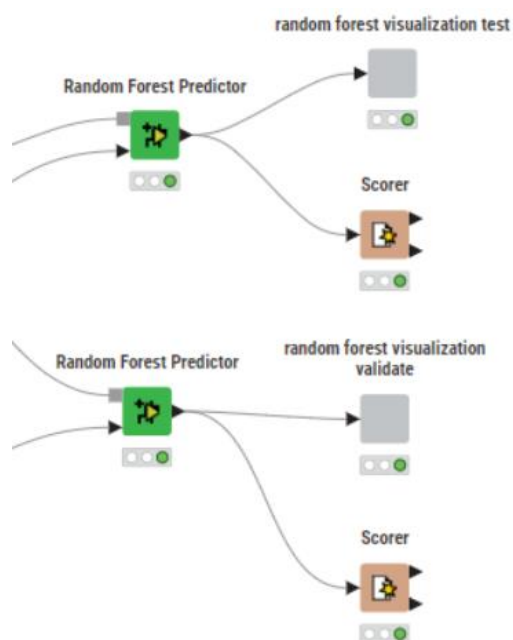
ตารางที่ 3.7 คำอธิบายและการตั้งค่า node สำหรับการประเมินผล (Evaluation & Visualization) สำหรับโมเดล Deep Learning (Keras) ดังในภาพที่ 3.13 และ 3.14

ชื่อ	node	คำอธิบาย	การตั้งค่า
Keras Network Executor		- ประมวลผลโมเดล Deep Learning ที่สร้างและฝึกไว้เรียบร้อยแล้ว โดยนำโมเดลไปใช้กับข้อมูลใหม่เพื่อทำการทำนายผล (Prediction)	- Input batch size = 32 - Conversion : From Image - Include Img Calc - add output : Softmax
Rule Engine		- กำหนดเงื่อนไข (if-else) ด้วยกฎที่เขียนเอง แล้วเพิ่มคอลัมน์ผลลัพธ์ตามเงื่อนไข	$\$dense\_3/Softmax:0\_0\$ > \$dense\_3/Softmax:0\_1\$ \text{ AND } \$dense\_3/Softmax:0\_0\$ > \$dense\_3/Softmax:0\_2\$ \text{ AND } \$dense\_3/Softmax:0\_0\$ > \$dense\_3/Softmax:0\_3\$ \Rightarrow \text{"benign"}$  $\$dense\_3/Softmax:0\_1\$ > \$dense\_3/Softmax:0\_0\$ \text{ AND } \$dense\_3/Softmax:0\_1\$ > \$dense\_3/Softmax:0\_2\$ \text{ AND }$

			<p>\$dense_3/Softmax:0_1\$ &gt; \$dense_3/Softmax:0_3\$ =&gt; "early"</p> <p>\$dense_3/Softmax:0_2\$ &gt; \$dense_3/Softmax:0_0\$ AND \$dense_3/Softmax:0_2\$ &gt; \$dense_3/Softmax:0_1\$ AND \$dense_3/Softmax:0_2\$ &gt; \$dense_3/Softmax:0_3\$ =&gt; "pre"</p> <p>\$dense_3/Softmax:0_3\$ &gt; \$dense_3/Softmax:0_0\$ AND \$dense_3/Softmax:0_3\$ &gt; \$dense_3/Softmax:0_1\$ AND \$dense_3/Softmax:0_3\$ &gt; \$dense_3/Softmax:0_2\$ =&gt; "pro"</p>
Scorer		<p>- เปรียบเทียบค่าจริง (actual) และค่าที่ทำนาย (predicted) ว่าตรงหรือไม่ตรงกัน แล้วแสดง Confusion Matrix และ สถิติด้าน ความแม่นยำต่าง ๆ ออกมา</p>	<p>- First Column : ALL_category - Second Column : prediction</p>
Bar Chart (JavaScript)		<p>- สร้างแผนภูมิแท่ง (Bar Chart)</p>	<p>- Category Column : prediction - Aggreation Method : Average - include : dense_3/Softmax:0_0, dense_3/Softmax:0_1, dense_3/Softmax:0_2, dense_3/Softmax:0_3</p>

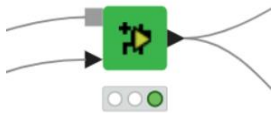

ROC Curve	<p><b>ROC Curve</b></p> 	- แสดงผลกราฟ ROC (Receiver Operating Characteristic Curve)	- Target Column : precision
-----------	---	--	-----------------------------

### 3.4.2 Random Forest



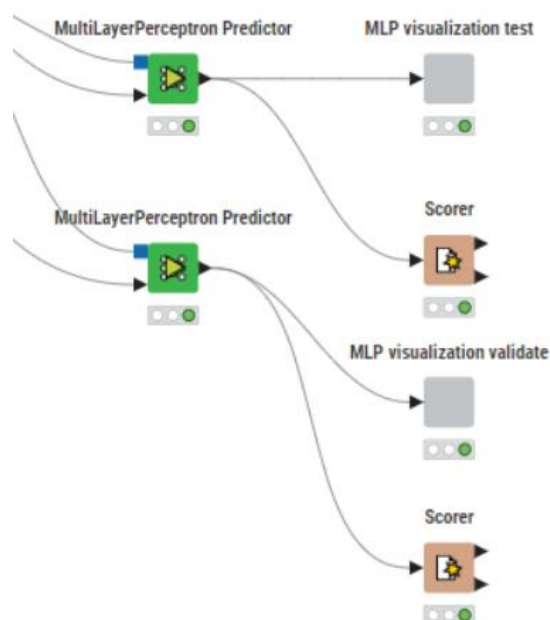
ภาพที่ 3.15 การประเมินผล (Evaluation & Visualization) สำหรับโมเดล Random Forest

ตารางที่ 3.8 คำอธิบายและการตั้งค่า node สำหรับการประเมินผล (Evaluation & Visualization) สำหรับโมเดล Random Forest ดังในภาพที่ 3.15

ชื่อ	node	คำอธิบาย	การตั้งค่า
Random Forest Predictor	<p><b>Random Forest Predictor</b></p> 	- นำโมเดลที่เรียนรู้แล้วด้วย Random Forest มาทำนายข้อมูลใหม่	- Change prediction column name : Prediction (target) - Append overall prediction confidence
Scorer	<p><b>Scorer</b></p> 	- เปรียบเทียบค่าจริง (actual) และค่าที่ทำนาย (predicted) ว่าตรงหรือไม่ตรงกัน แล้วแสดง Confusion	- First Column : target - Second Column : Prediction (target)

		Matrix และ สถิติด้าน ความแม่นยำต่าง ๆ ออกมา	
--	--	---	--

### 3.4.3 MLP



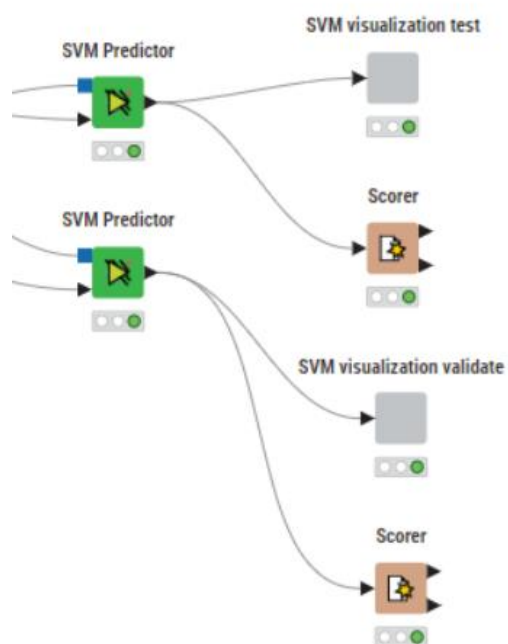
ภาพที่ 3.16 การประเมินผล (Evaluation & Visualization) สำหรับโมเดล MLP

ตารางที่ 3.9 คำอธิบายและการตั้งค่า node สำหรับการประเมินผล (Evaluation & Visualization) สำหรับโมเดล MLP ดังในภาพที่ 3.16

ชื่อ	node	คำอธิบาย	การตั้งค่า
MultiLayer Perceptron Predictor		- นำโมเดล MultiLayer Perceptron ที่ฝึกไว้แล้ว มาทำนายผลลัพธ์ของข้อมูลใหม่	- Change prediction column name : Prediction (target) - Append columns with class probabilities
Scorer		- เปรียบเทียบค่าจริง (actual) และค่าที่ทำนาย (predicted) ว่าตรงหรือไม่ตรงกัน แล้วแสดง Confusion Matrix และ สถิติด้าน	- First Column : target - Second Column : Prediction (target)

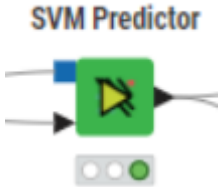
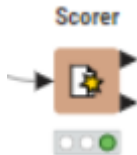
		ความแม่นยำต่าง ๆ ออกมา	
--	--	---------------------------	--

### 3.4.4 SVM



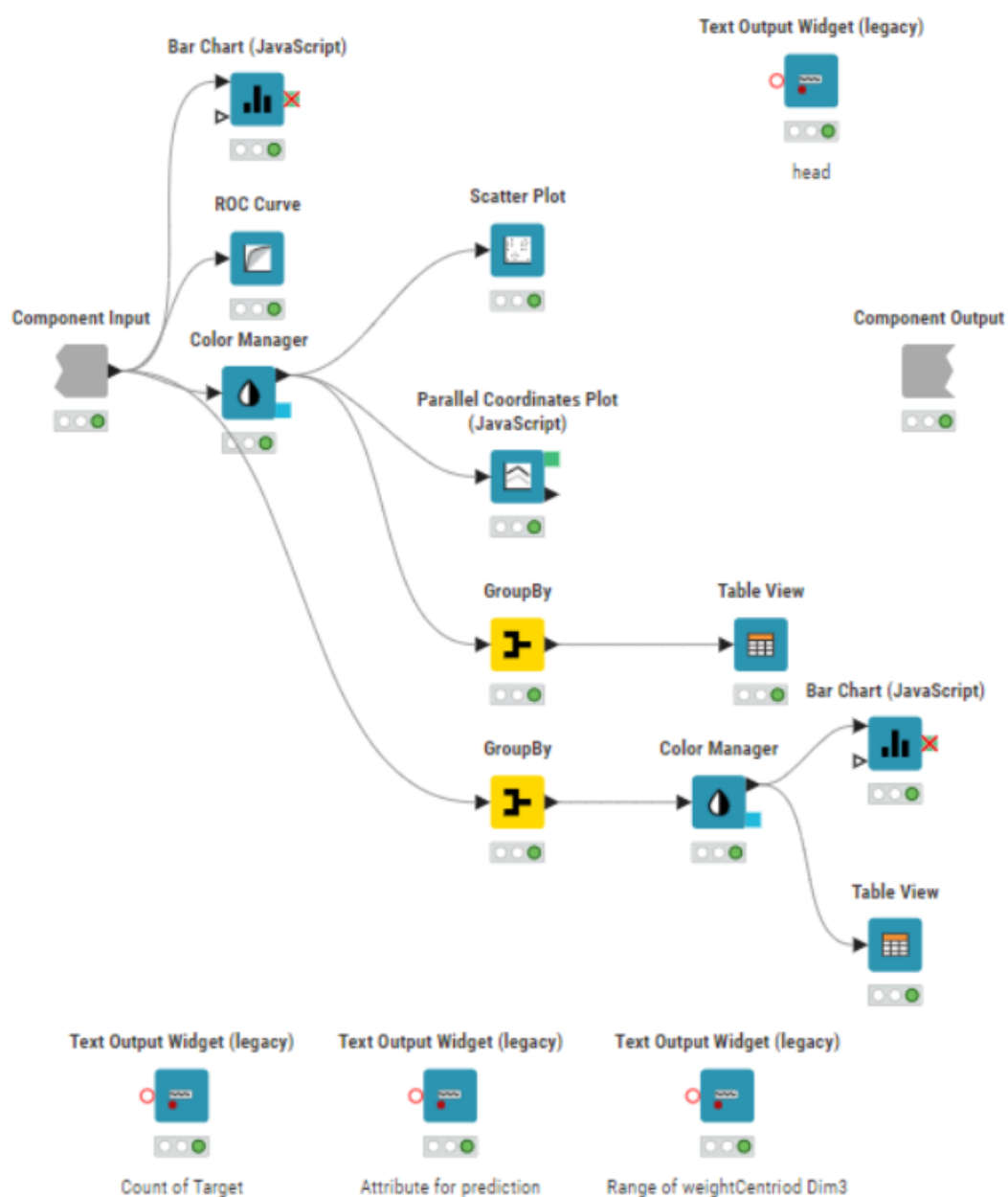
ภาพที่ 3.17 การประเมินผล (Evaluation & Visualization) สำหรับโมเดล SVM

ตารางที่ 3.10 คำอธิบายและการตั้งค่า node สำหรับการประเมินผล (Evaluation & Visualization) สำหรับโมเดล SVM ดังในภาพที่ 3.17

ชื่อ	node	คำอธิบาย	การตั้งค่า
SVM Predictor		- ทำนายผลลัพธ์จากข้อมูลใหม่ (Unseen Data) โดยใช้โมเดลที่ฝึกไว้ด้วย SVM Learner	- Change prediction column name : Prediction (target)
Scorer		- เปรียบเทียบค่าจริง (actual) และค่าที่ทำนาย (predicted) ว่าตรงหรือไม่ตรงกัน แล้วแสดง Confusion	- First Column : target - Second Column : Prediction (target)





		Matrix และ สถิติด้าน ความแม่นยำต่าง ๆ ออกมา	
--	--	---	--

### 3.4.5 Visualization สำหรับโมเดล Random Forest, MLP และ SVM


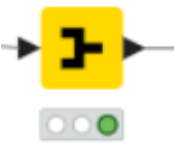



ภาพที่ 3.18 Visualization สำหรับโมเดล Random Forest, MLP และ SVM

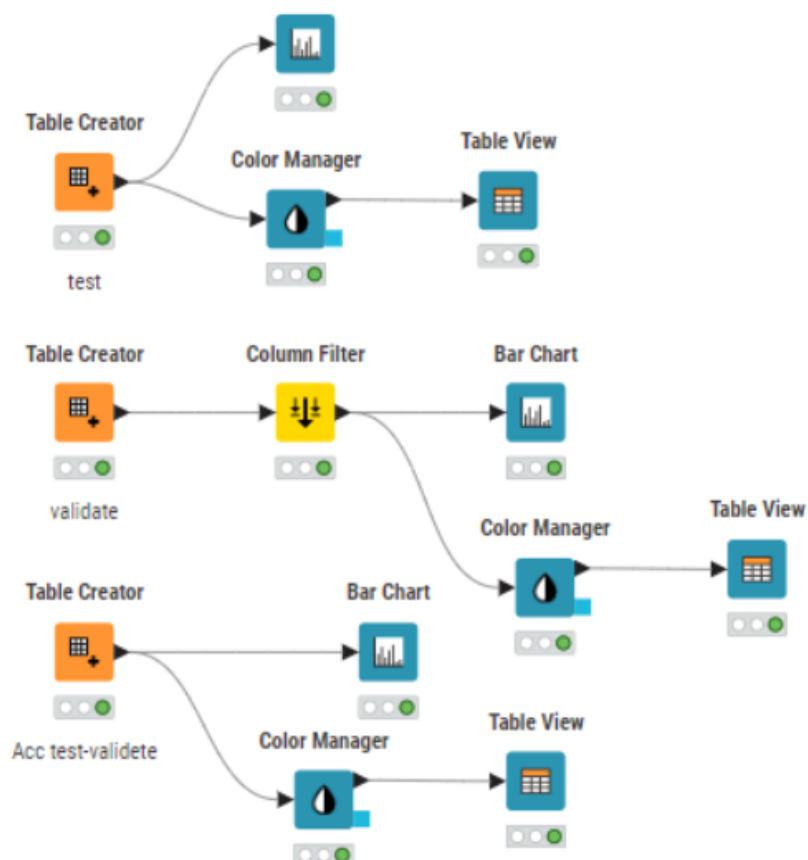
ตารางที่ 3.11 คำอธิบายและการตั้งค่า node สำหรับ Visualization สำหรับโมเดล Random Forest, MLP และ SVM ดังในภาพที่ 3.18

ชื่อ	node	คำอธิบาย	การตั้งค่า
Bar Chart (JavaScript)	<b>Bar Chart (JavaScript)</b> 	- สร้างแผนภูมิแท่ง (Bar Chart)	<div> <b>ด้านบน</b> <ul style="list-style-type: none"> <li>- Category Column : Prediction</li> <li>- Aggreation Method : Average</li> <li>- include : Mean, Sum, Squeare of Sum, Std Dev, Variance, Skewness, Kurtosis, WeightedCentroid Dim 1, WeightedCentroid Dim 2, WeightedCentroid Dim 3</li> </ul> </div> <hr/> <div> <b>ด้านล่าง</b> <ul style="list-style-type: none"> <li>- Category Column : target</li> <li>- Aggreation Method : Average</li> <li>- include : Count(Prediction(target))</li> </ul> </div>
ROC Curve	<b>ROC Curve</b> 	- แสดงผลกราฟ ROC (Receiver Operating Characteristic Curve)	- target column : Prediction (target)
Color Manager	<b>Color Manager</b> 	- กำหนดสีให้กับข้อมูล	- Color by : Prediction (target)
Scatter Plot	<b>Scatter Plot</b> 	- แสดงกราฟกระจาย (Scatter Plot) เพื่อแสดงความสัมพันธ์ระหว่าง สองคอลัมน์เชิงตัวเลข	<ul style="list-style-type: none"> <li>- Horizontal dimension : Prediction (target)</li> <li>- Vertical dimension : WeightedCentroid Dim3</li> <li>- Color dimension : Prediction (target)</li> </ul>




Parallel Coordinates Plot (JavaScript)		- สร้างกราฟ Parallel Coordinates เพื่อแสดง แนวโน้มของข้อมูลหลายมิติพร้อมกันในภาพเดียว	- include : Mean, Sum, Sqaure of Sum, Std Dev, Variance, Skewness, Kurtosis, WeightedCentroid Dim 1, WeightedCentroid Dim 2, WeightedCentroid Dim 3, Prediction (target)
GroupBy		- ใช้สำหรับจัดกลุ่มข้อมูลและสรุปผล ตามค่าของคอลัมน์ที่กำหนด	<div> <b>ด้านบน</b>            Settings            - include : Prediction (target)            Manual Aggregation            - Sum (mean)            - Sqaure of Sum (mean)            - Std Dev (mean)            - Variance, Skewness (mean)            - Kurtosis (mean)            - WeightedCentroid Dim 1 (mean)            - WeightedCentroid Dim 2 (mean)            - WeightedCentroid Dim 3 (mean)         </div> <hr/> <div> <b>ด้านล่าง</b>            Settings            - Include : target            Manual Aggregation            - Prediction (target) (Count)         </div>
Table View		- แสดงข้อมูลในรูปแบบตารางแบบ Interactive	- include : ทั้งหมด





### 3.4.6 เปรียบเทียบแต่ละ Model



ภาพที่ 3.19 การประเมินผล (Evaluation & Visualization) และเปรียบเทียบแต่ละ Model

ตารางที่ 3.11 คำอธิบายและการตั้งค่า node การประเมินผล (Evaluation & Visualization) และเปรียบเทียบแต่ละ Model ดังในภาพที่ 3.19

ชื่อ	node	คำอธิบาย	การตั้งค่า
Table Creator	<b>Table Creator</b> 	- สร้างตารางข้อมูล	- test : คัดลอกตาราง Accuracy statistic สำหรับการ test ทุกโมเดล - validate : คัดลอกตาราง Accuracy statistic สำหรับการ validate ทุกโมเดล - Acc test-validate : คัดลอก accuracy สำหรับการ test และ validate

Column Filter		<ul style="list-style-type: none"> <li>- เลือกเฉพาะบางคอลัมน์ที่ต้องการเก็บไว้ และ กรอง (ลบ) คอลัมน์อื่นๆ ออกไป</li> </ul>	<ul style="list-style-type: none"> <li>- Excludes Column ที่ไม่ได้ใช้ออก</li> </ul>
Bar Chart		<ul style="list-style-type: none"> <li>- สร้างแผนภูมิแท่ง (Bar Chart)</li> </ul>	<div> <b>Test</b> <ul style="list-style-type: none"> <li>- Category dimension : Model</li> <li>- Aggregation : Average</li> <li>- Include : Recall, Precision&lt; F-measure</li> </ul> </div> <hr/> <div> <b>Validate</b> <ul style="list-style-type: none"> <li>- Category dimension : Model</li> <li>- Aggregation : Average</li> <li>- Include : Recall, Precision&lt; F-measure</li> </ul> </div> <hr/> <div> <b>Acc test-validate</b> <ul style="list-style-type: none"> <li>- Category dimension : Model</li> <li>- Aggregation : None</li> <li>- Include : Accuracy-test, Accuracy-validate</li> </ul> </div>
Color Manager		<ul style="list-style-type: none"> <li>- กำหนดสีให้กับข้อมูล</li> </ul>	<ul style="list-style-type: none"> <li>- test : Categories</li> <li>- validate : Categories</li> <li>- Acc test-validate : Model</li> </ul>
Table View		<ul style="list-style-type: none"> <li>- แสดงข้อมูลในรูปแบบตารางแบบ Interactive</li> </ul>	<ul style="list-style-type: none"> <li>- include : ทั้งหมด</li> </ul>

## บทที่ 4

### ผลการประเมินประสิทธิภาพของโมเดล

เพื่อประเมินความสามารถของระบบในการจำแนกประเภทของเซลล์เม็ดเลือดขาวชนิดเฉียบพลัน (ALL) ได้อย่างแม่นยำ ในงานนี้ได้ทดสอบและเปรียบเทียบประสิทธิภาพของโมเดล Machine Learning และ Deep Learning ทั้ง 4 แบบ ได้แก่ Keras (Deep Learning), Random Forest, Multilayer Perceptron (MLP), และ Support Vector Machine (SVM) ผลลัพธ์ได้ดังนี้

#### 4.1 Deep Learning (Keras)

##### 4.1.1 Test

##### 4.1.1.1 Confusion Metrix

RowID	benign ↓ Number (integer)	early Number (integer)	pre Number (integer)	pro Number (integer)
benign	90	7	1	2
early	7	91	2	0
pre	2	5	93	0
pro	1	0	0	99

ภาพที่ 4.1 Confusion Metrix จากการทดสอบโมเดล Deep Learning (Keras) ด้วย test set

จากภาพที่ 4.1 พบว่า

คลาส benign: ทำนายถูกต้อง 90 ครั้ง จาก 100 ตัวอย่าง มีการทำนายผิดเป็น early จำนวน 7 ครั้ง, pre 1 ครั้ง และ pro 2 ครั้ง สะท้อนให้เห็นว่าโมเดลมีความแม่นยำสูงในกลุ่มนี้ แต่ยังมีการสับสนกับกลุ่มที่มีลักษณะคล้ายกัน โดยเฉพาะ early ซึ่งมักเกิดขึ้นในช่วงพัฒนาการเริ่มต้นของเซลล์มะเร็ง

คลาส early: ทำนายถูกต้อง 91 ครั้ง และผิดพลาดเป็น benign จำนวน 7 ครั้ง และ pre 2 ครั้ง ชี้ให้เห็นถึงความท้าทายในการแยกแยะระยะเริ่มต้นของ ALL จากเซลล์ปกติ เนื่องจากลักษณะทางภาพอาจยังไม่แสดงอาการเด่นชัด

คลาส pre: ทำนายได้ถูกต้อง 93 ครั้ง จากทั้งหมด 100 ครั้ง มีการทำนายผิดเป็น benign จำนวน 2 ครั้ง และ early จำนวน 5 ครั้ง โมเดลสามารถแยกแยะเซลล์ pre-B ได้อย่างมีประสิทธิภาพ แต่อาจมีบางกรณีที่มีลักษณะยังทับซ้อนกับ early

คลาส pro: ทำนายถูกต้องถึง 99 ครั้ง จาก 100 ตัวอย่าง มีเพียง 1 ตัวอย่างที่ผิดพลาดโดยทำนายเป็น benign และไม่มีการสับสนกับกลุ่มอื่นเลย แสดงว่าเซลล์ pro-B มีลักษณะเด่นชัดมากที่สุดสำหรับโมเดลในการเรียนรู้และจำแนก

##### 4.1.1.2 Accuracy statistic

RowID	TruePosit... Number (inte...)	FalsePosi... Number (inte...)	TrueNeg... Number (inte...)	FalseNeg... Number (inte...)	Recall Number (dou...)	Precision Number (dou...)	Sensitivity Number (dou...)	Specificity Number (dou...)	F-measure Number (dou...)	Accuracy Number (dou...)	Cohen's k... Number (dou...)
benign	90	10	290	10	0.9	0.9	0.9	0.967	0.9	0.9	0.9
early	91	12	288	9	0.91	0.883	0.91	0.96	0.897	0.9	0.9
pre	93	3	297	7	0.93	0.969	0.93	0.99	0.949	0.9	0.9
pro	99	2	298	1	0.99	0.98	0.99	0.993	0.985	0.9	0.9
Overall	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.932	0.91

ภาพที่ 4.2 Accuracy statistic จากการทดสอบโมเดลด้วย Deep Learning (Keras) test set

จากภาพที่ 4.2 แสดงผลลัพธ์ของการประเมินประสิทธิภาพของโมเดล ซึ่งมีรายละเอียดดังนี้:

- benign: Recall และ Precision เท่ากับ 0.9 แสดงถึงความสามารถของโมเดลในการทำนายกลุ่มนี้ได้ อย่างแม่นยำทั้งในแง่ของการจับกลุ่มและความถูกต้องของการทำนาย Specificity อยู่ที่ 0.967 แสดงว่าโมเดลสามารถแยกแยะกลุ่ม benign ออกจากกลุ่มอื่นได้ดี
- early: Recall เท่ากับ 0.91 และ Precision เท่ากับ 0.883 แสดงว่ามีการทำนายถูกต้องในระดับดี แต่ยังมีความสับสนกับคลาสที่ใกล้เคียงกัน F-measure = 0.897 แสดงถึงความสมดุลของ precision และ recall ที่ยังอยู่ในระดับดี
- pre: Recall = 0.93, Precision = 0.969 และ Specificity = 0.99 แสดงถึงการจำแนกคลาส pre ได้อย่างยอดเยี่ยมทั้งในด้านความไวและความจำเพาะ F-measure สูงถึง 0.949 สะท้อนว่าโมเดลมี ประสิทธิภาพสูงในการทำนายคลาสนี้
- pro: มี Recall สูงถึง 0.99 และ Precision = 0.98 ซึ่งถือว่าสูงมาก F-measure เท่ากับ 0.985 และ Specificity 0.993 บ่งชี้ว่าโมเดลสามารถแยกแยะกลุ่ม pro ได้อย่างแม่นยำและแทบไม่มีข้อผิดพลาด
- Overall: Accuracy รวมของโมเดลอยู่ที่ 0.932 Cohen's Kappa อยู่ที่ 0.91 ซึ่งบ่งชี้ว่าโมเดลมีความแม่นยำสูงและสามารถคาดการณ์ได้ดีกว่าการเดาสุ่มอย่างมีนัยสำคัญ

#### 4.1.2 Validation

##### 4.1.2.1 Confusion Matrix

RowID	benign Number (integer)	early Number (integer)	pre Number (integer)	pro Number (integer)
benign	94	3	3	0
early	10	86	4	0
pre	2	3	93	2
pro	2	0	0	98

ภาพที่ 4.3 Confusion Matrix จากการทดสอบโมเดล Deep Learning (Keras) ด้วย validation set

จากภาพที่ 4.3 Confusion Matrix แสดงผลการทดสอบโมเดล Deep Learning (Keras) บนชุดข้อมูล validation สำหรับการจำแนกเซลล์มะเร็งเม็ดเลือดขาวชนิด Acute Lymphoblastic Leukemia (ALL) โดยแบ่งออกเป็น 4 ประเภท ได้แก่ benign, early, pre, และ pro ซึ่งแต่ละแถวแสดงค่าจริง (Actual class) และแต่ละคอลัมน์แสดงค่าที่โมเดลทำนาย (Predicted class)

โมเดลสามารถจำแนกเซลล์ benign ได้ถูกต้อง 94 ภาพจากทั้งหมด 100 ภาพ, เซลล์ early ได้ถูกต้อง 86 ภาพ, เซลล์ pre ได้ถูกต้อง 93 ภาพ และเซลล์ pro ได้ถูกต้องถึง 98 ภาพ ซึ่งแสดงถึงความสามารถในการจำแนกได้อย่างแม่นยำในเกือบทุกกลุ่ม โดยเฉพาะในกลุ่ม pro ที่มีผลการทำนายถูกต้องสูงสุด แสดงให้เห็นว่าโมเดล Deep Learning ที่สร้างขึ้นมีศักยภาพสูงในการจำแนกเซลล์มะเร็งแต่ละระยะจากภาพถ่ายได้อย่างมีประสิทธิภาพ

#### 4.1.2.2 Accuracy statistic

R...	TruePositi...	FalsePositi...	TrueNegat...	FalseNega...	Recall	Precision	Sensitivity	Specificity	F-measure	Accuracy	Cohen's ka...
Number (integ...	Number (integ...	Number (integ...	Number (integ...	Number (integ...	Number (doubl...	Number (doubl...	Number (doubl...	Number (doubl...	Number (doubl...	Number (doubl...	Number (doubl...
benign	94	14	286	6	0.94	0.87	0.94	0.953	0.904	0.94	0.94
early	86	6	294	14	0.86	0.935	0.86	0.98	0.896	0.86	0.86
pre	93	7	293	7	0.93	0.93	0.93	0.977	0.93	0.93	0.93
pro	98	2	298	2	0.98	0.98	0.98	0.993	0.98	0.98	0.98
Overall	0.927	0.927	0.927	0.927	0.927	0.927	0.927	0.927	0.927	0.927	0.903

ภาพที่ 4.4 Accuracy statistic จากการทดสอบโมเดลด้วย Deep Learning (Keras) ด้วย validation set

จากภาพที่ 4.4 ผลลัพธ์แสดงให้เห็นว่าโมเดลมีความสามารถในการจำแนกกลุ่ม pro ได้ดีที่สุด (Accuracy = 0.98, F-measure = 0.98) รองลงมาคือ pre (Accuracy = 0.93, F-measure = 0.93) และ benign (Accuracy = 0.94, F-measure = 0.904) ขณะที่กลุ่ม early แม้มี Accuracy ที่ 0.86 แต่ก็ยังรักษา Precision และ Recall ได้ในระดับที่ดี

ค่า Accuracy โดยรวมของโมเดลอยู่ที่ 0.927 และ Cohen's Kappa อยู่ที่ 0.903 ซึ่งสะท้อนว่าโมเดลมีความแม่นยำและสอดคล้องกับค่าจริงในระดับสูง โดยเฉพาะในสภาวะข้อมูลไม่สมดุล (Class imbalance) ก็ยังสามารถทำงานได้อย่างมีประสิทธิภาพ

## 4.2 Random Forest

### 4.2.1 Test

#### 4.2.1.1 Confusion Metrix

RowID	benign	early	pre	pro
Number (integer)	Number (integer)	Number (integer)	Number (integer)	Number (integer)
benign	98	1	2	2
early	3	95	0	0
pre	1	2	96	0
pro	0	0	2	98

ภาพที่ 4.5 Confusion Metrix จากการทดสอบโมเดล Random Forest ด้วย test set

จากภาพที่ 4.5 เมื่อพิจารณาผลการทำนายของโมเดล พบว่า

- กลุ่ม benign มีจำนวนภาพที่ทำนายถูกต้องมากถึง 98 ภาพ จากทั้งหมด 103 ภาพ ซึ่งถือว่าแม่นยำมาก
- กลุ่ม early ซึ่งเป็นระยะเริ่มต้นของ ALL ก็สามารถทำนายได้ถูกต้องถึง 95 ภาพ จาก 98 ภาพ

- กลุ่ม pre ที่อยู่ในช่วงกึ่งกลางของการพัฒนาโรค โมเดลสามารถทำนายได้แม่นยำถึง 96 ภาพ จาก 99 ภาพ
- กลุ่ม pro หรือระยะรุนแรงของโรค ก็แสดงให้เห็นถึงประสิทธิภาพสูงเช่นกัน โดยจำแนกได้ถูกต้องถึง 98 ภาพ จาก 100 ภาพ

#### 4.2.1.2 Accuracy statistic

RowID	TruePosit... Number (inte...)	FalsePosi... Number (inte...)	TrueNeg... Number (inte...)	FalseNeg... Number (inte...)	Recall Number (dou...)	Precision Number (dou...)	Sensitivity Number (dou...)	Specificity Number (dou...)	F-measure Number (dou...)	Accuracy Number (dou...)	Cohen's k... Number (dou...)
benign	98	4	293	5	0.951	0.961	0.951	0.987	0.956	①	①
early	95	3	299	3	0.969	0.969	0.969	0.99	0.969	②	②
pre	96	4	297	3	0.97	0.96	0.97	0.987	0.965	②	②
pro	98	2	298	2	0.98	0.98	0.98	0.993	0.98	②	②
Overall	①	①	①	①	②	①	①	①	①	0.968	0.957

ภาพที่ 4.6 Accuracy statistic จากการทดสอบโมเดล Random Forest ด้วย test set

จากภาพที่ 4.6 แสดงผลลัพธ์ดังนี้

- Recall และ Precision ในทุกคลาสมีค่าใกล้เคียงกันและอยู่ในช่วง 0.96–0.98 ซึ่งบ่งชี้ว่าโมเดลมีความแม่นยำสูงทั้งในแง่ของการตรวจจับตัวอย่างที่เป็นกลุ่มเป้าหมาย (Recall) และความถูกต้องของผลการจำแนก (Precision)
- ค่าของ F-measure อยู่ในช่วง 0.956 – 0.98 ซึ่งแสดงถึงความสมดุลที่ดีระหว่าง Precision และ Recall
- ค่าของ Specificity และ Sensitivity สำหรับแต่ละคลาสก็สูงมาก แสดงว่าโมเดลสามารถแยกแยะตัวอย่างที่ไม่ใช่กลุ่มเป้าหมายได้อย่างถูกต้อง และตรวจจับตัวอย่างของคลาสเป้าหมายได้ดีเช่นกัน
- ค่าความแม่นยำรวม (Overall Accuracy) อยู่ที่ 0.968 และ Cohen's Kappa ที่ 0.957 ซึ่งถือว่าอยู่ในระดับที่ยอดเยี่ยม สะท้อนให้เห็นว่าโมเดลมีความสอดคล้องในการจำแนกประเภทอย่างมีนัยสำคัญ

### 4.2.2 Validation

#### 4.2.2.1 Confusion Metrix

RowID	benign Number (integer)	early Number (integer)	pre Number (integer)	pro Number (integer)
benign	93	3	0	1
early	7	93	2	0
pre	0	4	93	4
pro	0	1	1	98

ภาพที่ 4.7 Confusion Metrix จากการทดสอบโมเดล Random Forest ด้วย validation set

จากภาพที่ 4.7 พบว่า

- คลาส benign ถูกจำแนกได้ถูกต้องจำนวน 93 ตัวอย่าง จาก 97 ตัวอย่างทั้งหมด ซึ่งผิดพลาดเพียงเล็กน้อย โดยมีการจำแนกผิดไปเป็น early 3 ตัวอย่าง และ pro 1 ตัวอย่าง

- คลาส early มีการจำแนกถูกต้อง 93 ตัวอย่าง เช่นกัน แต่มีข้อผิดพลาดบางส่วนคือ ถูกจำแนกเป็น benign 7 ตัวอย่าง และ pre 2 ตัวอย่าง
- คลาส pre มีการจำแนกถูกต้อง 93 ตัวอย่าง จากทั้งหมด 101 ตัวอย่าง โดยจำแนกผิดไปเป็น early 4 ตัวอย่าง และ pro 4 ตัวอย่าง
- คลาส pro มีการจำแนกถูกต้องสูงถึง 98 ตัวอย่าง โดยมีการจำแนกผิดเพียงเล็กน้อยคือไปเป็น pre และ benign อย่างละ 1 ตัวอย่าง

จากภาพรวมของ Confusion Matrix นี้ แสดงให้เห็นว่าโมเดล Random Forest มีความสามารถในการจำแนกเซลล์ในแต่ละคลาสได้ดี โดยเฉพาะคลาส benign และ pro ซึ่งมีความแม่นยำสูงมาก แต่อาจมีความสับสนเล็กน้อยระหว่างคลาส early กับ pre ซึ่งมีลักษณะภาพใกล้เคียงกันในบางกรณี

#### 4.2.2.2 Accuracy statistic

RowID	TruePosit... Number (inte...)	FalsePosi... Number (inte...)	TrueNeg... Number (inte...)	FalseNeg... Number (inte...)	Recall Number (dou...)	Precision Number (dou...)	Sensitivity Number (dou...)	Specificity Number (dou...)	F-measure Number (dou...)	Accuracy Number (dou...)	Cohen's k... Number (dou...)
benign	93	7	296	4	0.959	0.93	0.959	0.977	0.944	⑦	⑦
early	93	8	290	9	0.912	0.921	0.912	0.973	0.916	⑦	⑦
pre	93	3	296	8	0.921	0.969	0.921	0.99	0.944	⑦	⑦
pro	98	5	295	2	0.98	0.951	0.98	0.983	0.966	⑦	⑦
Overall	⑦	⑦	⑦	⑦	⑦	⑦	⑦	⑦	⑦	0.943	0.923

ภาพที่ 4.8 Accuracy statistic จากการทดสอบโมเดล Random Forest ด้วย validation set

จากภาพที่ 4.8 แสดงให้เห็นว่าโมเดลมีประสิทธิภาพในการแยกคลาสน้อยดังนี้

- Benign โมเดลสามารถจำแนกเซลล์ปกติได้ดี โดยมีค่า Recall อยู่ที่ 0.959 แสดงถึงความสามารถในการตรวจจับเซลล์ปกติที่แท้จริงได้มากถึง 95.9% ขณะเดียวกัน Precision เท่ากับ 0.93 และ Specificity อยู่ที่ 0.977 แสดงว่าโมเดลสามารถแยกแยะเซลล์ benign ออกจากเซลล์อื่นได้อย่างแม่นยำ โดยมีอัตราการวินิจฉัยผิดพลาด
- Early ค่า Recall ของเซลล์ early อยู่ที่ 0.912 และ Precision เท่ากับ 0.921 บ่งชี้ว่าโมเดลมีความสามารถที่ดีในการจำแนก subtype ดังกล่าว Specificity อยู่ที่ 0.973 แสดงให้เห็นว่าเซลล์ early ถูกจำแนกผิดน้อยมาก และมีความแม่นยำโดยรวมที่น่าเชื่อถือ
- Pre ในกลุ่มเซลล์ pre โมเดลมีค่า Recall เท่ากับ 0.921 และ Precision สูงถึง 0.969 ซึ่งถือว่าโดดเด่นเป็นพิเศษในการแยก subtype pre ได้อย่างถูกต้อง ค่า Specificity อยู่ที่ 0.990 ซึ่งบ่งชี้ถึงความเฉพาะเจาะจงสูงในการจำแนก subtype นี้จากกลุ่มอื่น
- pro การจำแนกเซลล์ pro ได้ผลลัพธ์ที่ดีมากเช่นกัน โดยมีค่า Recall เท่ากับ 0.980 และ Precision เท่ากับ 0.951 ขณะเดียวกัน Specificity อยู่ที่ 0.983 ซึ่งแสดงว่าโมเดลสามารถแยกแยะเซลล์ pro ได้อย่างมีประสิทธิภาพสูง และยังมีค่า F-measure สูงสุดในบรรดาทุกคลาส
- โมเดล Random Forest มีค่า Accuracy รวมอยู่ที่ 0.943 และค่า Cohen's Kappa เท่ากับ 0.923 ซึ่งเป็นดัชนีที่สะท้อนความสอดคล้องของผลลัพธ์การจำแนกกับข้อมูลจริงในระดับสูง แสดงให้เห็นว่า



โมเดลนี้มีความแม่นยำในการทำนายและจำแนกประเภทของเซลล์เม็ดเลือดขาวชนิดเฉียบพลันได้ดีในทุกคลาส

### 4.3 MLP

#### 4.3.1 Test

##### 4.3.1.1 Confusion Metrix

RowID	benign Number (integer)	early Number (integer)	pre Number (integer)	pro Number (integer)
benign	101	1	1	0
early	3	95	0	0
pre	1	1	95	2
pro	0	0	1	99

ภาพที่ 4.9 Confusion Metrix จากการทดสอบโมเดล MLP ด้วย test set

จากภาพที่ 4.9 ผลลัพธ์แสดงให้เห็นว่าโมเดลสามารถจำแนกประเภทของเซลล์ได้อย่างมีประสิทธิภาพในระดับสูง ดังรายละเอียดต่อไปนี้:

- benign จำแนกได้ถูกต้อง 101 จาก 103 ภาพ (Recall  $\approx$  0.98) โดยมีเพียง 2 ภาพที่ถูกจำแนกผิดไปยัง early และ pre อย่างละ 1 ภาพ แสดงให้เห็นว่าโมเดลมีความสามารถสูงในการแยกแยะเซลล์ปกติ
- Early จำแนกได้ถูกต้อง 95 จาก 98 ภาพ (Recall  $\approx$  0.97) โดยมีการจำแนกผิดไปยัง benign 3 ภาพ ถือว่าอยู่ในระดับแม่นยำที่ดี
- Pre มีความแม่นยำสูง โดยจำแนกได้ถูกต้อง 95 จาก 99 ภาพ (Recall  $\approx$  0.96) และมีการจำแนกผิดไปยัง benign และ pro อย่างละ 1 และ 2 ภาพตามลำดับ ซึ่งแสดงว่าโมเดลมีความสามารถที่ดีในการแยก subtype ที่มีความคล้ายคลึงกันเชิงภาพ
- Pro จำแนกได้ถูกต้อง 99 จาก 100 ภาพ (Recall = 0.99) มีเพียง 1 ภาพที่ถูกจัดอยู่ในกลุ่ม pre ซึ่งอาจเกิดจากลักษณะภาพบางส่วนที่มีความใกล้เคียงกัน

##### 4.3.1.2 Accuracy statistic

RowID	TruePosit... Number (inte...)	FalsePosi... Number (inte...)	TrueNega... Number (inte...)	FalseNeg... Number (inte...)	Recall Number (dou...)	Precision Number (dou...)	Sensitivity Number (dou...)	Specificity Number (dou...)	F-measure Number (dou...)	Accuracy Number (dou...)	Cohen's k...
benign	101	4	293	2	0.981	0.962	0.981	0.987	0.971	⑦	⑦
early	95	2	300	3	0.969	0.979	0.969	0.993	0.974	⑦	⑦
pre	95	2	299	4	0.96	0.979	0.96	0.993	0.969	⑦	⑦
pro	99	2	298	1	0.99	0.98	0.99	0.993	0.985	⑦	⑦
Overall	⑦	⑦	⑦	⑦	⑦	⑦	⑦	⑦	⑦	0.975	0.967

ภาพที่ 4.10 Accuracy statistic จากการทดสอบโมเดล MLP ด้วย test set

จากภาพที่ 4.10 แสดงผลลัพธ์การประเมินประสิทธิภาพของโมเดล Multilayer Perceptron (MLP) เมื่อทำการทดสอบกับชุดข้อมูลทดสอบ (Test Set) ซึ่งประกอบด้วยภาพจำนวน 400 ภาพ จำแนกออกเป็น 4 ประเภทของเซลล์ ได้แก่ benign, early, pre และ pro โดยมีการคำนวณค่าวัดทางสถิติเช่น Recall, Precision,

Specificity, F-measure และ Accuracy เพื่อใช้ในการประเมินคุณภาพของโมเดลอย่างรอบด้าน ผลลัพธ์ที่ได้สามารถสรุปได้ดังนี้:

- Recall (Sensitivity) สำหรับทุกคลาสมีค่าสูงตั้งแต่ 0.96 ถึง 0.99 สะท้อนว่าโมเดลสามารถตรวจจับตัวอย่างที่เป็นจริง (True Positive) ได้อย่างมีประสิทธิภาพ โดยเฉพาะในคลาส pro ซึ่งมี Recall เท่ากับ 0.99
- Precision มีค่าสูงในทุกคลาส โดยอยู่ในช่วง 0.962 ถึง 0.98 แสดงถึงความแม่นยำของโมเดลในการคาดเดาว่าผลลัพธ์ที่เป็นบวกนั้นถูกต้องจริง
- Specificity มีค่าอยู่ระหว่าง 0.987 ถึง 0.993 บ่งชี้ถึงความสามารถของโมเดลในการจำแนกตัวอย่างที่ไม่เป็นบวกได้อย่างถูกต้อง
- F-measure หรือค่าเฉลี่ยถ่วงน้ำหนักระหว่าง Precision และ Recall มีค่าเฉลี่ยสูงในทุกกลุ่ม โดยเฉพาะในกลุ่ม pro มีค่า F-measure สูงสุดที่ 0.985
- Accuracy รวม ของโมเดลมีค่าเท่ากับ 0.975 (97.5%) แสดงว่าโมเดลสามารถจำแนกประเภทเซลล์ได้อย่างถูกต้องในอัตราที่สูง
- Cohen's Kappa = 0.967 ซึ่งเป็นดัชนีที่สะท้อนความสอดคล้องระหว่างค่าที่โมเดลทำนายได้กับค่าความจริง เมื่อเทียบกับค่าที่ได้จากการสุ่ม ชี้ให้เห็นว่าโมเดลมีความสอดคล้องสูงมาก

### 4.3.2 Validation

#### 4.3.2.1 Confusion Metrix

R... ↓	benign Number (integer)	early Number (integer)	pre Number (integer)	pro Number (integer)
benign	96	0	0	1
early	6	93	1	2
pre	1	2	94	4
pro	0	0	0	100

ภาพที่ 4.11 Confusion Metrix จากการทดสอบโมเดล MLP ด้วย validation set

จากภาพที่ 4.11 ผลการประเมิน พบว่าโมเดล MLP สามารถจำแนกตัวอย่างภาพได้อย่างแม่นยำ โดยเฉพาะคลาส pro ซึ่งมีความแม่นยำสูงมาก กล่าวคือจำแนกได้ถูกต้องทั้งหมด 100 ภาพ โดยไม่มีการจำแนกผิดไปยังคลาสอื่นเลย ซึ่งสะท้อนถึงความสามารถในการตรวจจับเซลล์ในระยะวิกฤตได้อย่างชัดเจนและมีประสิทธิภาพ

ในส่วนของคลาสอื่น ๆ ก็ให้ผลลัพธ์ที่น่าพึงพอใจ โดยในคลาส benign โมเดลสามารถจำแนกได้ถูกต้อง 96 ภาพ และเกิดข้อผิดพลาดเพียง 1 ภาพที่จำแนกผิดไปยังคลาส pro ส่วนคลาส early ซึ่งเป็นระยะเริ่มต้นของการเปลี่ยนแปลงของเซลล์ ถูกจำแนกได้ถูกต้อง 93 ภาพ และมีข้อผิดพลาดรวม 8 ภาพ ที่โมเดลจำแนกผิดไปยังคลาส benign และ pro ขณะที่คลาส pre ซึ่งเป็นระยะก่อนวิกฤต มีจำนวนการจำแนกถูกต้อง 94 ภาพ และมีการจำแนกผิดไปยัง early และ pro รวม 3 ภาพ

RowID	TruePosit... Number (inte...)	FalsePosi... Number (inte...)	TrueNega... Number (inte...)	FalseNeg... Number (inte...)	Recall Number (dou...)	Precision Number (dou...)	Sensitivity Number (dou...)	Specificity Number (dou...)	F-measure Number (dou...)	Accuracy Number (dou...)	Cohen's k... Number (dou...)
benign	96	7	296	1	0.99	0.932	0.99	0.977	0.96	①	①
early	93	2	296	9	0.912	0.979	0.912	0.993	0.944	①	①
pre	94	1	298	7	0.931	0.989	0.931	0.997	0.959	①	①
pro	100	7	293	0	1	0.935	1	0.977	0.966	①	①
Overall	①	①	①	①	①	①	①	①	①	0.958	0.943

ภาพที่ 4.12 Accuracy statistic จากการทดสอบโมเดล MLP ด้วย validation set

จากภาพที่ 4.12 ผลลัพธ์โดยรวมของโมเดล MLP มีค่า Accuracy เท่ากับ 0.958 และ Cohen's Kappa เท่ากับ 0.943 ซึ่งแสดงถึงความแม่นยำที่สูงและระดับความสอดคล้องระหว่างผลลัพธ์ของโมเดลกับค่าจริงที่ตีพิมพ์

- สำหรับคลาส benign โมเดลสามารถจำแนกได้แม่นยำสูง โดยมีค่า Recall เท่ากับ 0.99, Precision เท่ากับ 0.932 และ F-measure เท่ากับ 0.96 สะท้อนให้เห็นว่าโมเดลสามารถจับภาพเซลล์ปกติได้อย่างแม่นยำเกือบสมบูรณ์
- คลาส early แม้จะมีค่า Recall ต่ำกว่าเล็กน้อยที่ 0.912 แต่มีค่า Precision สูงถึง 0.979 ทำให้ F-measure อยู่ที่ 0.944 ซึ่งแสดงว่าแม้จะพลาดตัวอย่างบางส่วน แต่ยังคงสามารถทำนายได้อย่างแม่นยำในภาพรวม
- คลาส pre ให้ผลที่น่าพอใจเช่นกัน โดยมี Recall เท่ากับ 0.931, Precision เท่ากับ 0.989 และ F-measure เท่ากับ 0.959 สะท้อนว่าโมเดลสามารถแยกแยะเซลล์ในระยะ pre ได้แม่นยำ
- สำหรับคลาส pro ซึ่งเป็นระยะรุนแรงที่สุด โมเดลสามารถจำแนกได้อย่างแม่นยำที่สุด โดยมีค่า Recall และ Sensitivity เท่ากับ 1.0 และค่า Precision เท่ากับ 0.935 ส่งผลให้ F-measure อยู่ที่ 0.966 ซึ่งถือว่าเป็นผลลัพธ์ที่ยอดเยี่ยม

## 4.4 SVM

### 4.4.1 Test

#### 4.4.1.1 Confusion Metrix

RowID	benign Number (integer)	early Number (integer)	pre Number (integer)	pro Number (integer)
benign	100	1	0	2
early	2	96	0	0
pre	0	1	96	2
pro	0	0	1	99

ภาพที่ 4.13 Confusion Metrix จากการทดสอบโมเดล SVM ด้วย test set

จากภาพที่ 4.13 ผลการจำแนกพบว่าโมเดลสามารถทำนายคลาส benign ได้อย่างแม่นยำ โดยจำแนกถูก 100 ภาพจากทั้งหมด 103 ภาพ มีความผิดพลาดเล็กน้อยเพียง 3 ภาพ ที่ถูกจำแนกเป็น early และ pro อย่างละ 1 และ 2 ภาพตามลำดับ

- สำหรับคลาส early ทำนายถูกต้อง 96 ภาพจากทั้งหมด 98 ภาพ โดยผิดพลาด 2 ภาพซึ่งถูกจัดอยู่ในคลาส benign
- คลาส pre โมเดลสามารถจำแนกถูกต้อง 96 ภาพเช่นกัน จากทั้งหมด 99 ภาพ มีข้อผิดพลาดเล็กน้อย โดยมี 1 ภาพที่จำแนกเป็น early และอีก 2 ภาพจำแนกเป็น pro
- คลาส pro มีความแม่นยำสูงเช่นกัน โดยจำแนกถูก 99 ภาพจากทั้งหมด 100 ภาพ มีเพียง 1 ภาพที่จำแนกผิดเป็น pre

#### 4.4.1.2 Accuracy statistic

RowID	TruePositi... Number (integ...	FalsePositi... Number (integ...	TrueNegat... Number (integ...	FalseNega... Number (integ...	Recall Number (doub...	Precision Number (doub...	Sensitivity Number (doub...	Specificity Number (doub...	F-measure Number (doub...	Accuracy Number (doub...	Cohen's k... Number (doub...
benign	100	2	295	3	0.971	0.98	0.971	0.993	0.976	0.978	0.97
early	96	2	300	2	0.98	0.98	0.98	0.993	0.98	0.978	0.97
pre	96	1	300	3	0.97	0.99	0.97	0.997	0.98	0.978	0.97
pro	99	4	296	1	0.99	0.961	0.99	0.987	0.975	0.978	0.97
Overall	0.978	0.978	0.978	0.978	0.978	0.978	0.978	0.978	0.978	0.978	0.97

ภาพที่ 4.14 Accuracy statistic จากการทดสอบโมเดล SVM ด้วย test set

จากภาพที่ 1.4 พบว่า

- คลาส benign ค่า Recall เท่ากับ 0.971 และ Precision เท่ากับ 0.980 แสดงว่าโมเดลสามารถจำแนกภาพเซลล์ benign ได้อย่างถูกต้องทั้งในแง่ของความแม่นยำและความครบถ้วน โดยมีค่า F-measure รวมที่ 0.976
- คลาส early ให้ค่า Recall เท่ากับ 0.980 และ Precision เท่ากับ 0.980 เช่นเดียวกัน แสดงถึงความสามารถในการทำนายเซลล์ early ได้อย่างดีเยี่ยม โดยมีค่า F-measure เท่ากับ 0.980
- คลาส pre ให้ค่า Recall เท่ากับ 0.970 และ Precision สูงถึง 0.990 ส่งผลให้ค่า F-measure รวมสูงถึง 0.980 ซึ่งแสดงให้เห็นว่าโมเดลสามารถแยกแยะเซลล์ชนิดนี้ได้แม่นยำมาก
- คลาส pro มีค่า Recall และ Precision เท่ากับ 0.990 และ 0.961 ตามลำดับ และให้ค่า F-measure อยู่ที่ 0.975 ซึ่งถือว่ามีความแม่นยำและสมดุลระหว่าง Precision และ Recall ได้เป็นอย่างดี
- โดยภาพรวม โมเดล SVM บนชุดข้อมูลทดสอบนี้ให้ค่า Accuracy สูงถึง 0.978 และค่า Cohen's Kappa = 0.970 ซึ่งเป็นค่าที่แสดงถึงความสอดคล้องระหว่างการจำแนกของโมเดลกับค่าจริงได้อย่างยอดเยี่ยม

#### 4.4.2 Validation

##### 4.4.2.1 Confusion Metrix

RowID	benign Number (integer)	early Number (integer)	pre Number (integer)	pro Number (integer)
benign	96	1	0	0
early	5	95	0	2
pre	0	1	97	3
pro	1	0	0	99

ภาพที่ 4.15 Confusion Metrix จากการทดสอบโมเดล SVM ด้วย validation set

จากภาพที่ 4.15 จะเห็นได้ว่า

- คลาส benign มีภาพที่ทำนายถูกต้องจำนวน 96 ภาพ และมีการทำนายผิดไปยังคลาส early และ pro อย่างละ 1 ภาพเท่านั้น แสดงถึงความแม่นยำที่ค่อนข้างสูง
- คลาส early ถูกจำแนกถูกต้อง 95 ภาพ และมีการผิดพลาด 5 ภาพที่ถูกจำแนกเป็น benign และ 2 ภาพที่ทำนายผิดเป็น pro
- คลาส pre มีความแม่นยำสูงเช่นกัน โดยทำนายถูกต้อง 97 ภาพ ผิดพลาดเพียง 1 ภาพเป็น early และ 3 ภาพเป็น pro
- คลาส pro ทำนายถูกต้อง 99 ภาพ ผิดเพียง 1 ภาพที่ทำนายเป็น benign
- จากภาพจะเห็นได้ว่า โมเดล SVM มีความสามารถในการจำแนกเซลล์ทั้ง 4 ประเภทได้ค่อนข้างแม่นยำ โดยมีจำนวนการทำนายผิดค่อนข้างน้อย และสามารถแยกประเภทที่มีลักษณะคล้ายกัน เช่น early กับ benign หรือ pre กับ pro ได้ในระดับที่ดี

#### 4.4.2.2 Accuracy statistic

R...	TruePosit...	FalsePosi...	TrueNeg...	FalseNeg...	Recall	Precision	Sensitivity	Specificity	F-measure	Accuracy	Cohen's k...
Number (inte...)	Number (inte...)	Number (inte...)	Number (inte...)	Number (dou...)	Number (dou...)	Number (dou...)	Number (dou...)	Number (dou...)	Number (dou...)	Number (dou...)	Number (dou...)
benign	96	6	297	1	0.99	0.941	0.99	0.98	0.965	0.968	0.957
early	95	2	296	7	0.931	0.979	0.931	0.993	0.955	0.968	0.957
pre	97	0	299	4	0.96	1	0.96	1	0.98	0.968	0.957
pro	99	5	295	1	0.99	0.952	0.99	0.983	0.971	0.968	0.957
Overall	0.968	0.968	0.968	0.968	0.968	0.968	0.968	0.968	0.968	0.968	0.957

ภาพที่ 4.16 Accuracy statistic จากการทดสอบโมเดล SVM ด้วย validation set

จากภาพที่ 4.16 แสดงให้เห็นว่าผลลัพธ์จากโมเดล SVM ดังนี้

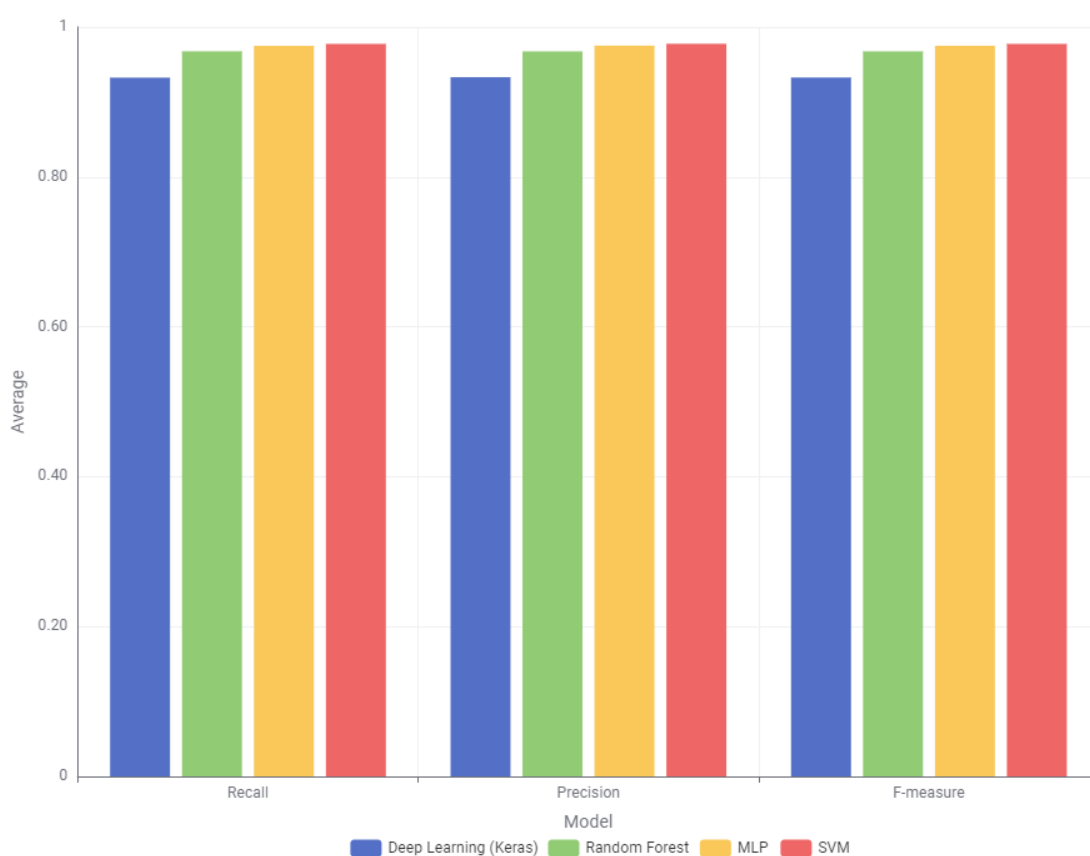
- คลาส benign: มีค่า Recall 0.99, Precision 0.941, F-measure 0.965 และ Accuracy สูงถึง 0.968 แสดงถึงความสามารถในการจำแนกเซลล์ที่ไม่เป็นมะเร็งได้อย่างแม่นยำ
- คลาส early: ให้ค่า Recall 0.931 และ Precision 0.979 ซึ่งส่งผลให้ F-measure อยู่ที่ 0.955 แสดงว่าแม้จะมีข้อผิดพลาดบ้างในการจำแนก แต่โดยรวมยังถือว่าเป็นประสิทธิภาพดี
- คลาส pre: ให้ค่า Recall 0.96 และ Precision 1.0 แสดงว่าไม่มีการทำนายผิดว่าเป็นคลาสอื่นเลย ทำให้ Specificity และ F-measure อยู่ในระดับสูงสุดคือ 1.0 และ 0.98 ตามลำดับ

- คลาส pro: มีค่า Recall และ Sensitivity เท่ากับ 0.99 และ Precision ที่ 0.952 ส่งผลให้ F-measure เท่ากับ 0.971
- ค่าความแม่นยำโดยรวม (Overall Accuracy) ของโมเดล SVM เมื่อทดสอบกับ Validation Set เท่ากับ 0.968 และค่า Cohen's Kappa ซึ่งวัดความสอดคล้องของการจำแนกอยู่ที่ 0.957 ซึ่งถือว่าอยู่ในระดับ “almost perfect agreement”

## 4.5 เปรียบเทียบแต่ละ Model

### 4.5.1 Test

Bar Chart



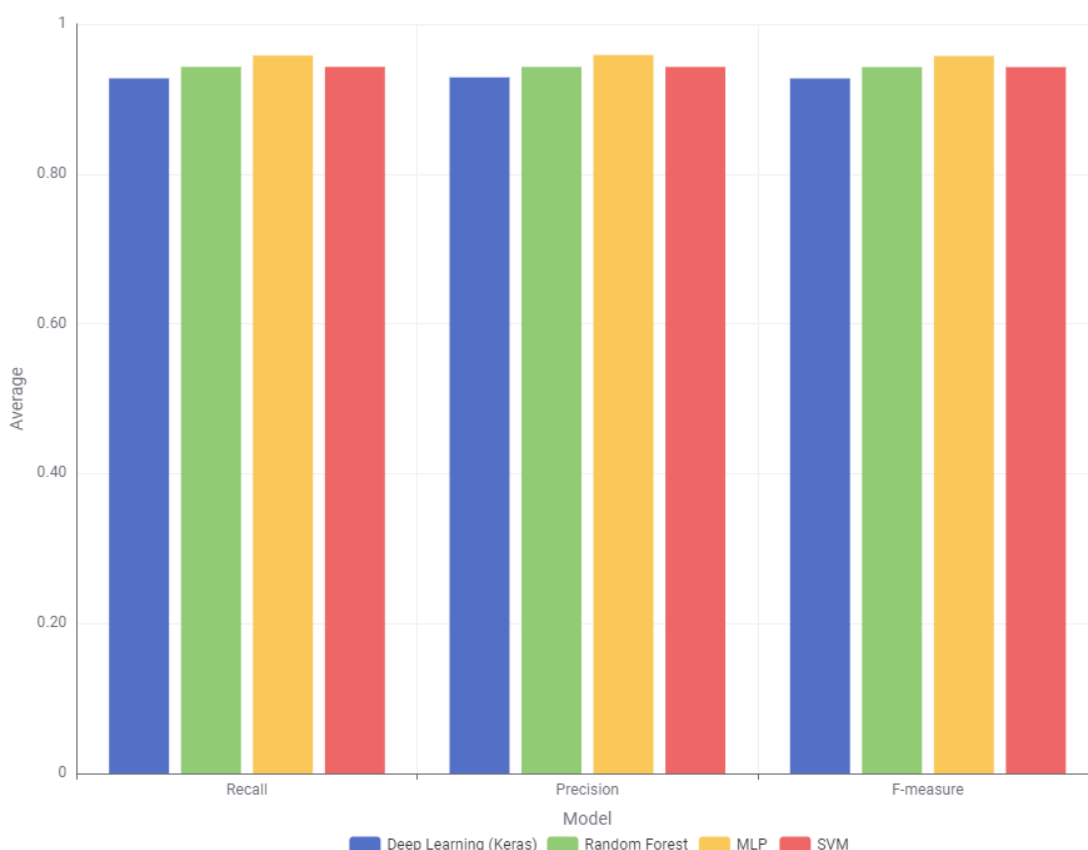
ภาพที่ 4.17 การเปรียบเทียบแต่ละ Model ด้วย Test set

ภาพที่ 4.17 แสดงการเปรียบเทียบค่าประสิทธิภาพของโมเดลทั้ง 4 ได้แก่ Deep Learning (Keras), Random Forest, Multilayer Perceptron (MLP) และ Support Vector Machine (SVM) โดยพิจารณาค่าชี้วัดเฉลี่ยของ Recall, Precision และ F-measure ที่ได้จากการประเมินผลด้วย ชุดข้อมูล Test Set จากกราฟข้างแสดงให้เห็นว่า

- โมเดล Deep Learning (Keras) ให้ค่าประสิทธิภาพที่ค่อนข้างน้อยกว่ารูปแบบอื่นเล็กน้อยในทั้ง 3 ตัวชี้วัด โดยอยู่ในช่วงประมาณ 0.92
- ในขณะที่ โมเดล Random Forest, MLP และ SVM มีค่าประสิทธิภาพเฉลี่ยของ Recall, Precision และ F-measure ที่สูงกว่ามาก โดยมีค่าเฉลี่ยเกือบถึง 0.98 ซึ่งบ่งชี้ว่าโมเดลทั้งสามสามารถจำแนกข้อมูลได้อย่างแม่นยำและครอบคลุมทุกคลาสได้ดี
- โมเดล SVM มีค่าประสิทธิภาพสูงสุดในทุกตัวชี้วัดเล็กน้อยเมื่อเทียบกับ MLP และ Random Forest แต่ความต่างนั้นน้อยมาก
- จากผลการเปรียบเทียบนี้ สามารถสรุปได้ว่า SVM, MLP และ Random Forest มีประสิทธิภาพที่ใกล้เคียงกันและโดดเด่นกว่ารูปแบบ Deep Learning (Keras) ในชุดข้อมูลนี้ ซึ่งอาจเกิดจากลักษณะของชุดข้อมูลที่ไม่ซับซ้อนมากนัก จึงเอื้อต่อการเรียนรู้ของโมเดลแบบดั้งเดิมได้ดีกว่าโมเดลเชิงลึกที่ต้องการข้อมูลขนาดใหญ่และซับซ้อนในการแสดงศักยภาพอย่างเต็มที่.

#### 4.5.2 Validation

Bar Chart



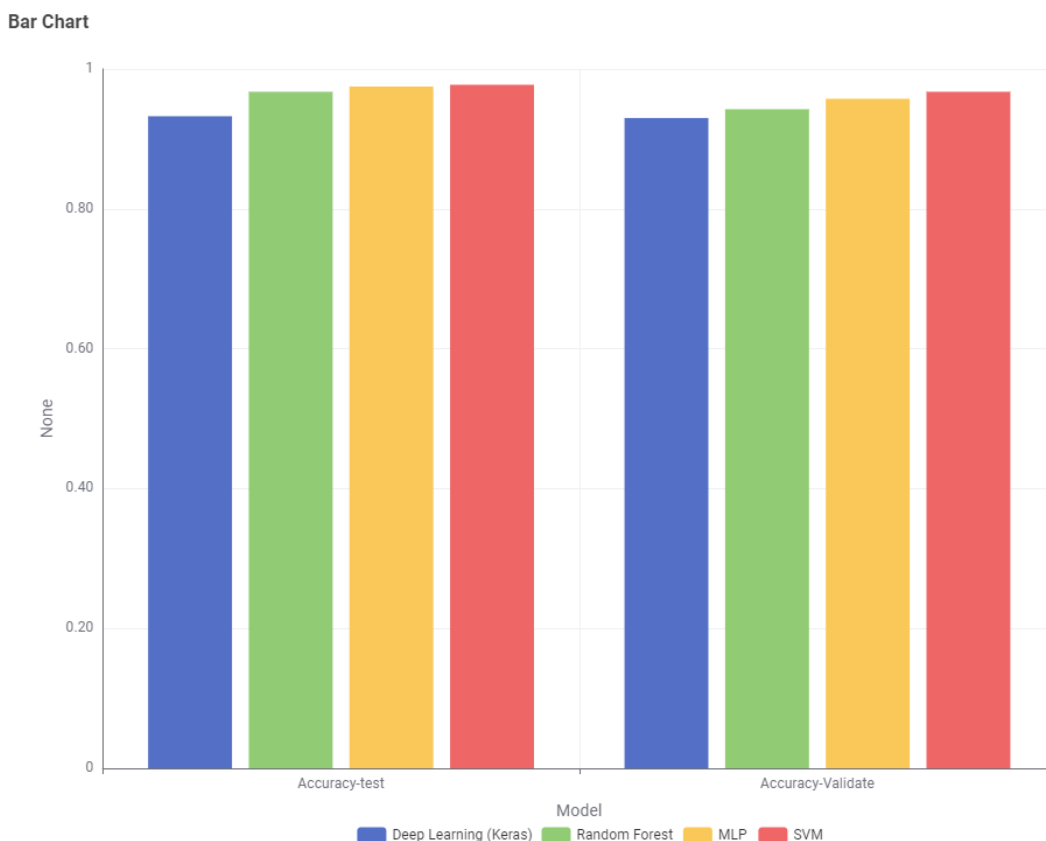
ภาพที่ 4.18 การเปรียบเทียบแต่ละ Model ด้วย Validate set

จากภาพที่ 4.18 แสดงกราฟแท่งเปรียบเทียบค่าประสิทธิภาพของโมเดลทั้ง 4 ได้แก่ Deep Learning (Keras), Random Forest, Multilayer Perceptron (MLP) และ Support Vector Machine (SVM) จากการ

ทดสอบด้วย ชุดข้อมูล Validation Set โดยพิจารณาค่าเฉลี่ยของตัวชี้วัดหลักได้แก่ Recall, Precision และ F-measure จากกราฟแสดงให้เห็นว่า

- โมเดล Deep Learning (Keras) มีค่าประสิทธิภาพต่ำกว่ารูปแบบอื่นเล็กน้อย โดย Recall, Precision และ F-measure อยู่ในช่วงประมาณ 0.93 ซึ่งถือว่าอยู่ในเกณฑ์ที่ดี แต่ยังไม่โดดเด่นเท่ากับโมเดลอื่น
- โมเดล MLP แสดงประสิทธิภาพสูงที่สุดในทุกตัวชี้วัด โดยมีค่าเฉลี่ยเกือบถึง 0.98 ทั้งใน Recall, Precision และ F-measure แสดงให้เห็นถึงความสามารถในการจำแนกภาพได้อย่างแม่นยำ
- Random Forest และ SVM ก็แสดงค่าประสิทธิภาพที่ดีมาก โดยมีผลลัพธ์ใกล้เคียงกัน และสูงกว่า Deep Learning (Keras) เล็กน้อย แต่ยังไม่พียง MLP อยู่เล็กน้อย
- จากผลการเปรียบเทียบนี้แสดงให้เห็นว่า โมเดล MLP มีความเหมาะสมที่สุดสำหรับข้อมูล Validation Set ในงานจำแนกประเภทเซลล์มะเร็งเม็ดเลือดขาวชนิด ALL ในขณะที่ Random Forest และ SVM ก็เป็นตัวเลือกที่ดี ส่วน Deep Learning (Keras) ยังสามารถพัฒนาเพิ่มเติมได้อีก เพื่อเพิ่มความแม่นยำในอนาคต.

#### 4.5.3 Accuracy test and validation



ภาพที่ 4.19 การเปรียบเทียบ accuracy ของแต่ละ Model ด้วย test set และ Validate set



จากภาพที่ 4.19 แสดงการเปรียบเทียบค่าความแม่นยำ (Accuracy) ของโมเดลทั้ง 4 ได้แก่ Deep Learning (Keras), Random Forest, Multilayer Perceptron (MLP) และ Support Vector Machine (SVM) โดยพิจารณาจากผลการทดสอบด้วยชุดข้อมูล Test Set และ Validation Set จากกราฟจะเห็นว่า

- โมเดล Deep Learning (Keras) มีค่า Accuracy ต่ำที่สุดทั้งใน Test และ Validation Set โดยมีค่าอยู่ที่ประมาณ 0.93 ซึ่งแม้จะอยู่ในระดับที่น่าพอใจ แต่ยังไม่เทียบเท่ากับโมเดลอื่น ๆ ในการจำแนกประเภทข้อมูลในงานนี้
- โมเดล Random Forest, MLP และ SVM แสดงให้เห็นถึงค่าความแม่นยำที่สูงและใกล้เคียงกันมาก ในทั้งสองชุดข้อมูล โดยเฉพาะ SVM มีค่า Accuracy สูงสุดในทั้ง Test Set และ Validation Set ซึ่งแสดงถึงความเสถียรและประสิทธิภาพในการจำแนกประเภทข้อมูล
- MLP ก็มีผลลัพธ์ที่ดีเยี่ยม โดยมีค่าความแม่นยำเกือบเท่ากับ SVM ทั้งสองชุดข้อมูล
- จากภาพนี้แสดงให้เห็นว่า โมเดล SVM และ MLP มีความสม่ำเสมอและแม่นยำสูงที่สุด เมื่อเทียบกับโมเดลอื่น ๆ ในการจำแนกประเภทเซลล์มะเร็งเม็ดเลือดขาวชนิด ALL ขณะที่ Deep Learning (Keras) ยังสามารถปรับปรุงเพิ่มเติมเพื่อเพิ่มประสิทธิภาพได้ในอนาคต.

## บทที่ 5

### บทสรุป

#### 5.1 สรุปผลการดำเนินงาน

จากการดำเนินโครงการ พบว่าสามารถออกแบบและพัฒนาเวิร์กโฟลว์บนแพลตฟอร์ม KNIME Analytics Platform ได้ครบถ้วนตามวัตถุประสงค์ที่ 1.2.1 โดยใช้ข้อมูลจากชุดภาพมะเร็งเม็ดเลือดขาว ALL ทั้งหมด 4 ประเภท ได้แก่ benign (เซลล์ปกติ), early pre-B, pre-B และ pro-B (เซลล์มะเร็งระยะต่าง ๆ) ซึ่งผ่านกระบวนการเตรียมข้อมูลและการดึงคุณลักษณะ (Feature Extraction) ด้วย Node เช่น Image Normalizer, Image Converter, และ Image Features ก่อนนำไปใช้ฝึกโมเดลจำแนกประเภทด้วยอัลกอริธึมต่าง ๆ เวิร์กโฟลว์ที่ออกแบบสามารถแยกโครงสร้างออกเป็นกลุ่ม node ได้แก่

- I/O (การนำเข้าข้อมูลภาพ)
- Feature Extraction (การดึงคุณลักษณะจากภาพ)
- Model Training (การฝึกโมเดลจำแนก)
- Evaluation and Visualization (การประเมินผลและแสดงผล)

ในโครงการได้ดำเนินการทดสอบและประเมินผลของโมเดลทั้ง 4 แบบ ได้แก่

- Deep Learning (Keras)
- Random Forest
- Multilayer Perceptron (MLP)
- Support Vector Machine (SVM)

ผลการเปรียบเทียบจากทั้งชุด Test set และ Validation set พบว่า

ตารางที่ 5.1 ผลการเปรียบเทียบจากทั้งชุด Test set และ Validation set

โมเดล	Accuracy (Test)	Accuracy (Validate)	สรุปผล
Deep Learning (Keras)	0.932	0.927	ต่ำที่สุดในกลุ่ม อาจต้องปรับปรุงโครงข่าย
Random Forest	0.968	0.943	ประสิทธิภาพดีและเสถียร
Multilayer Perceptron (MLP)	0.975	0.958	ความแม่นยำสูงมาก
Support Vector Machine (SVM)	0.978	0.968	ดีที่สุดในทุกค่าประเมิน

SVM แสดงให้เห็นถึงความแม่นยำสูงสุดในทุกชุดข้อมูล โดยเฉพาะค่าความแม่นยำ (Accuracy), ค่าความไว (Recall), ความจำเพาะ (Specificity) และค่า F-measure สูงเกือบทุกคลาส ส่งผลให้เหมาะสมต่อการใช้งานจริงในการวินิจฉัยภาพเซลล์

ในโครงการนี้ได้มีการออกแบบการแสดงผลลัพธ์ในรูปแบบภาพและแผนภูมิที่หลากหลาย ได้แก่:

- Confusion Matrix: สำหรับแต่ละโมเดลในชุด Test และ Validation เพื่อวิเคราะห์การทำนายถูก-ผิดแบบละเอียด
- ROC Curve และค่า AUC: สำหรับดูความสามารถของโมเดลแต่ละตัวในการแยกแยะข้อมูล
- Bar Chart เปรียบเทียบ Recall, Precision, F1-score: ระหว่างโมเดลต่าง ๆ
- แผนภูมิเปรียบเทียบ Accuracy ระหว่าง Test และ Validate

การใช้ Visualization ดังกล่าวช่วยให้เข้าใจผลลัพธ์ได้อย่างชัดเจนในเชิงภาพ และสามารถอธิบายการทำงานของระบบกับผู้ใช้หรือบุคลากรที่ไม่มีความเชี่ยวชาญด้านเทคนิคได้ง่ายยิ่งขึ้น

## 5.2 ข้อเสนอแนะในการพัฒนาเพิ่มเติม

- เพิ่มจำนวนข้อมูลภาพที่หลากหลายยิ่งขึ้น เพื่อป้องกัน Overfitting และเพิ่มความสามารถในการจำแนกได้แม่นยำในกรณีจริง
- ปรับแต่งพารามิเตอร์ของโมเดล Deep Learning เพิ่มเติม เช่น จำนวนชั้นซ่อน (Hidden Layers), ขนาด Batch, Learning Rate และ Epoch เพื่อเพิ่มประสิทธิภาพ
- ทดลองใช้การรวมผลจากหลายโมเดล (Ensemble Learning) เพื่อเพิ่มความเสถียรของผลลัพธ์
- เพิ่มการวิเคราะห์ด้วย ROC Curve และ AUC อย่างเป็นระบบ สำหรับทุกคลาส เพื่อแสดงประสิทธิภาพของโมเดลในทุกมิติ
- ขยายผลการใช้งาน เช่น พัฒนาอินเทอร์เฟซหรือบริการที่เชื่อมต่อกับโมเดล เพื่อนำไปใช้ในงานวินิจฉัยทางคลินิกจริง