

Earthworm Farm Monitoring System using AI and IoT

ระบบติดตามสภาพแวดล้อมฟาร์มไส้เดือนโดยปัญญาประดิษฐ์และไอโอที

ทีมพัฒนาโครงการ

นาย บวรพล	จิตต์อุไร
นาย อัมมาร์	เชื้อผู้ดี

อาจารย์ที่ปรึกษาโครงการ

รองศาสตราจารย์ ดร.กอบเกียรติ สระอุบล

คณะวิทยาศาสตร์ประยุกต์

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

สาระสำคัญของโครงการ

โครงการนี้เป็นโครงการที่มุ่งเน้นสำหรับเกษตรกรและผู้ที่ต้องการเลี้ยงไส้เดือนเพื่อเพิ่มผลผลิตให้สูงขึ้น เนื่องจากปัจจัยในการเลี้ยงไส้เดือนให้สามารถดำรงชีวิตในเบตดิง(ที่อยู่และอาหาร) ได้ให้มีการผสมพันธุ์และขยายพันธุ์ได้นั้นอาศัยปัจจัยควบคุมหลายๆชนิด ไม่ว่าจะเป็นความชื้นที่ต้องอยู่ระหว่าง 40-70% อีกทั้งไส้เดือนเป็นสัตว์ไม่ชอบแสงแดดอุณหภูมิที่เหมาะสมคือระหว่าง 15-33 องศาเซลเซียส หากอุณหภูมิต่ำหรือสูงไปไส้เดือนก็จะไม่มีการขยายพันธุ์ หรือไม่มีการเพิ่มจำนวน และไส้เดือนสามารถเจริญเติบโตได้ในดินที่มีสภาพเป็นกลาง^[1]

และหนึ่งในปัจจัยสำคัญที่ส่งผลกระทบต่อการดำรงชีวิต คือศัตรูตามธรรมชาติของไส้เดือน อาทิเช่น หนู คางคก และกิ้งก่า โครงการนี้จึงพัฒนาระบบตรวจจับศัตรูของไส้เดือนโดยใช้ปัญญาประดิษฐ์ นำมาประยุกต์ใช้เพื่อลดการสูญเสียของจำนวนไส้เดือน

หลักการและเหตุผล

ไส้เดือนเป็นสิ่งมีชีวิตที่อาศัยอยู่ร่วมกับมนุษย์เป็นเวลายาวนาน เป็นสิ่งมีชีวิตที่หลายคนรู้สึกรังเกียจและไม่ชอบ เนื่องจากลักษณะทั่วไปของไส้เดือนมีลักษณะเป็นรูปร่างทรงกระบอกยาว ลักษณะคล้ายหนอน แต่หลายคนอาจจะไม่ทราบว่าไส้เดือนเปรียบเสมือนเครื่องจักรธรรมชาติทางการเกษตร เป็นสิ่งมีชีวิตที่ให้ประโยชน์มากกว่าให้โทษต่อมนุษย์ เนื่องจากการเคลื่อนที่ของไส้เดือนเป็นลักษณะการขอนไช ทำให้โครงสร้างของดินนั้นดีขึ้น เป็นผู้ย่อยสลายซากอินทรีย์ในระบบนิเวศน์

อีกทั้งไส้เดือนเป็นสิ่งมีชีวิตที่สามารถสร้างรายได้ให้กับผู้ประกอบการเดือนละหลายหมื่นถึงแสนบาท โดยการขายมูลไส้เดือนที่มีส่วนช่วยเพิ่มธาตุอาหารแก่ดิน อีกทั้งตัวไส้เดือนเองก็สามารถขายเพื่อสร้างรายได้ คณะผู้พัฒนาจึงมีแนวคิดการทำระบบฟาร์มเข้ามาช่วยดูแล เพื่อที่จะสามารถแบ่งเบาความเสี่ยงของการสูญเสียไส้เดือนจากเหตุการณ์ที่ไม่คาดคิด อีกทั้งยังช่วยลดภาระค่าใช้จ่ายให้กับผู้ประกอบการจากการจ้างงานพนักงานเพื่อมาดูแลไส้เดือน โดยใช้เทคโนโลยีไอโอที (Internet Of Thing : IoT) และปัญญาประดิษฐ์ (Artificial Intelligence : AI) ที่ทำให้สามารถดูแลไส้เดือนได้อย่างดียิ่งขึ้น ลดเวลาที่ต้องดูแลและสามารถจัดการดูแลฟาร์มได้อย่างมีประสิทธิภาพมากขึ้น

วัตถุประสงค์

เพื่อพัฒนาระบบควบคุมจัดการดูแลฟาร์มไส้เดือนด้วยระบบ IoT และรวมถึงพัฒนาระบบช่วยแจ้งเตือนเมื่อพบศัตรูตามธรรมชาติของไส้เดือนได้แก่ หนู คางคก และกิ้งก่า

ที่มาและปัญหา

ไส้เดือนเป็นสัตว์ที่ไม่มีกระดูกสันหลัง ลักษณะลำตัวเป็นข้อปล้อง เป็นสัตว์ที่มี 2 เพศในตัวเดียวกัน ไส้เดือนเป็นสัตว์ที่อ่อนไหวต่อความชื้นและอุณหภูมิอย่างมาก เนื่องจากผิวหนังของไส้เดือนบอบบางมาก ต้องผลิตสารหล่อลื่นตลอดเวลา ทำให้ความชื้นและอุณหภูมิเป็นหนึ่งในปัจจัยสำคัญในการดำรงชีวิต หากอุณหภูมิสูงเกินไป ไส้เดือนก็จะหนีออกจากที่อยู่โดยตามธรรมชาติ หรือหากความชื้นต่ำเกินไปก็จะทำให้ไส้เดือนไม่เจริญเติบโต ไม่เจริญอาหารและไม่เจริญพันธุ์ เพราะไส้เดือนเป็นสัตว์ที่ไม่มีฟันในการเคี้ยวอาหาร ลักษณะการกินอาหารคือการดูด ดังนั้นหากความชื้นต่ำจะทำให้ไส้เดือนไม่สามารถกินอาหารได้เนื่องจากอาหารมีความแข็งกว่าปกติ และอีกหนึ่งปัจจัยหลักที่ส่งผลต่อผลผลิตคือศัตรูตามธรรมชาติของไส้เดือน ปัญหาที่มักพบในการเลี้ยงไส้เดือนได้แก่

1. ไส้เดือนไม่ค่อยกินอาหาร ปัจจัยอาจเกิดจากเบตดิงที่ไม่เหมาะสมอาจเกิดจากความชื้นที่น้อยเกินไปเพราะเนื่องจากความชื้นเป็นตัวกระตุ้นให้ไส้เดือนสามารถกินอาหารได้ดีและเจริญพันธุ์ได้ดี^[2]
2. ไส้เดือนขึ้นมาอยู่บนหน้าเบตดิง เกิดจากอุณหภูมิของเบตดิงสูงเกินไปทำให้ไส้เดือนไม่สามารถอาศัยอยู่ได้หรือบางส่วนอาจตายในเบตดิงเนื่องจากไม่สามารถออกมาได้^[2]
3. ไส้เดือนไม่สามารถกินหรือย่อยสลายสารอินทรีย์ได้รวมถึงไม่สามารถอาศัยอยู่ได้
4. มีศัตรูบุกรุกที่อยู่ของไส้เดือน ทำให้จำนวนไส้เดือนลดลง



เป้าหมายและขอบเขต

โครงการนี้มีเป้าหมายเพื่อพัฒนาระบบควบคุมจัดการดูแลฟาร์มไส้เดือนด้วยระบบ IoT และระบบตรวจจับและแจ้งเตือนศัตรูของไส้เดือน เพื่อเพิ่มประสิทธิภาพในการติดตามสภาพแวดล้อมการเป็นอยู่ในของไส้เดือน โดยจะนำอุปกรณ์ Sensors มาใช้สำหรับวัดค่าความชื้น และอุณหภูมิของสภาพแวดล้อมของไส้เดือน รวมถึงผู้ใช้สามารถใช้ Mobile Application ในการติดตามค่าความชื้นและอุณหภูมิของเบตดิง และสามารถควบคุมการทำงานของปั้มน้ำได้โดยผ่าน Mobile Application ได้ตลอดเวลา รวมถึงมีการแจ้งเตือนหากตรวจพบศัตรูหรือความผิดปกติของอุณหภูมิ และความชื้น

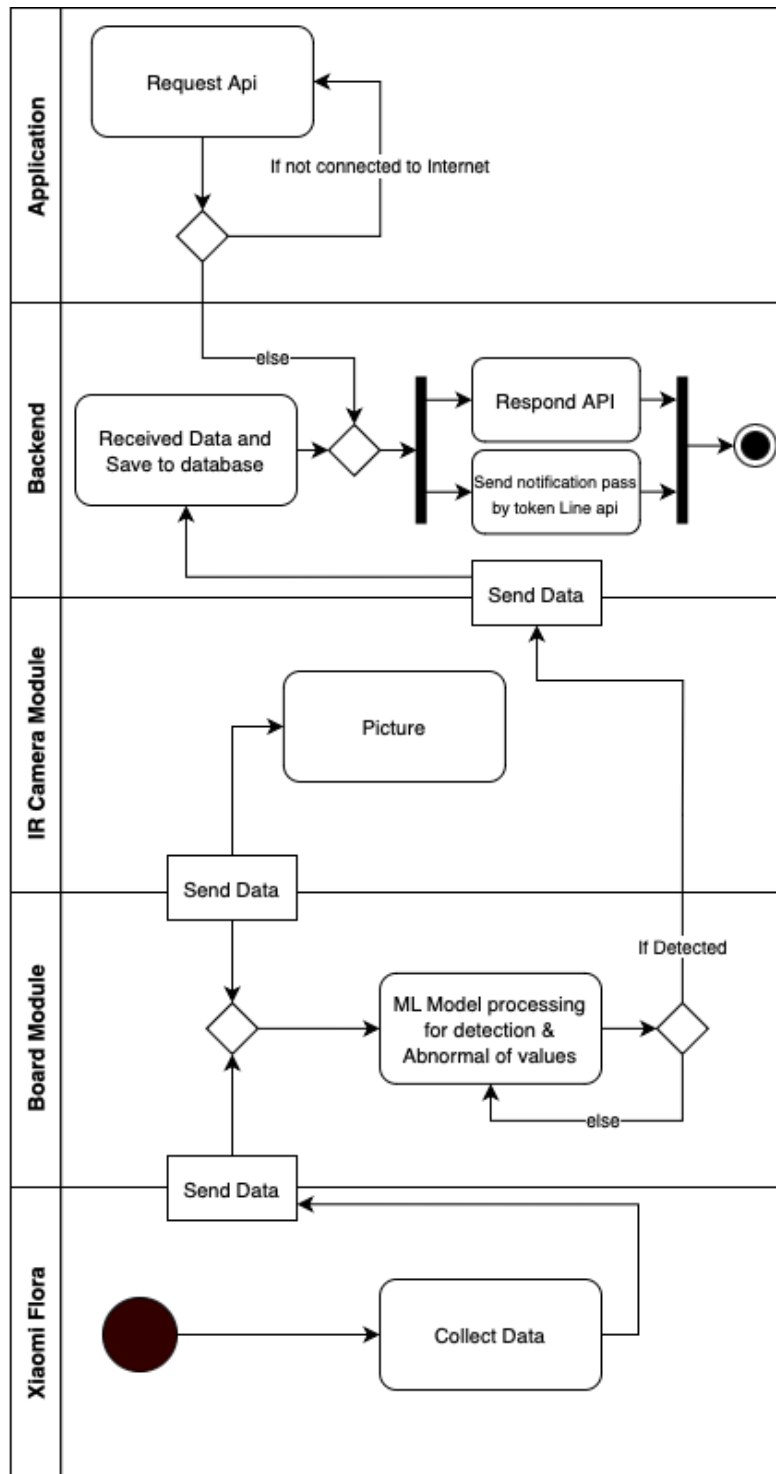
ความสามารถของระบบ

1. สามารถ Monitor อุณหภูมิและความชื้นผ่าน Mobile Application
2. สามารถควบคุมการทำงานของปั้มน้ำผ่าน Mobile Application แบบ Manual และ Auto
3. ระบบการควบคุมการจ่ายน้ำอัตโนมัติเพื่อรักษาความชื้นในดินให้เหมาะสมกับการเพาะเลี้ยงไส้เดือน
4. ระบบจะแจ้งเตือนเมื่อความชื้นหรืออุณหภูมิอยู่ในค่าที่สูงหรือต่ำผิดปกติ และตรวจจับการบุกรุกของศัตรูไส้เดือนได้แก่ หนู คางคก และกิ้งก่า โดยระบบจะทำการแจ้งเตือนไปที่โทรศัพท์เมื่อมีการตรวจพบ

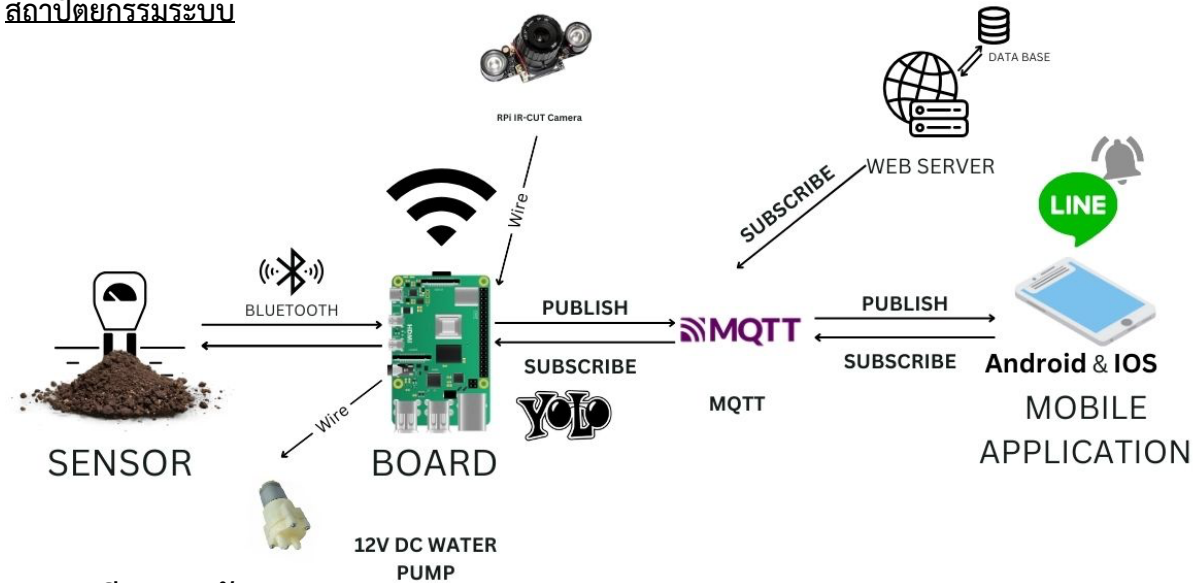
ประโยชน์ที่คาดว่าจะได้รับ

1. เพิ่มผลผลิตไส้เดือน
2. ลดค่าใช้จ่ายในการดูแลฟาร์ม
3. ผู้ดูแลสามารถติดตามการเปลี่ยนแปลงภายในโรงเรือนได้อย่างใกล้ชิด
4. สามารถลดความเสี่ยงในการสูญเสียไส้เดือนจากการบุกรุกของศัตรูได้มากขึ้น

การทำงานของระบบ



สถาปัตยกรรมระบบ



รายละเอียดการพัฒนา

No.	Product Name	Price (Bath)
1	Raspberry Pi 4 Model B+	2,526
2	Xiaomi Flora	592
3	OV5647 IR Camera	279
4	Buzzer	15
5	Relay 5 V	100
6	DC 12 V Water Pump	150
7	Adaptor DC 12 V	80

1. Raspberry Pi 4 Model B+ อ่านค่า Sensor และส่งค่าไปยัง Webserver
2. Webserver และ database เก็บข้อมูลที่จาก Sensor
3. Xiaomi Flora ในการวัด อุณหภูมิและความชื้น
4. DC 12 v Pump Water ใช้สำหรับการปั้มน้ำฉีดพรมหน้าดินและฉีดเพื่อลดอุณหภูมิ
5. Mobile application ในการแสดงค่าจาก Sensor และควบคุมการเปิดปิดปั้มน้ำแบบ Auto และ Manual
6. Buzzer สำหรับแจ้งเตือนผู้ใช้กรณีระบบตรวจพบการบุกรุกของศัตรู
7. OV5647 IR Camera สำหรับรับภาพในสภาพแวดล้อมที่มีแสง และแสงน้อย
8. Relay 5V ไว้ควบคุมปั้มน้ำ.
9. Adaptor DC 12 V จ่ายไฟให้ปั้มน้ำ
10. Line Notify สำหรับแจ้งเตือนผ่านโทรศัพท์

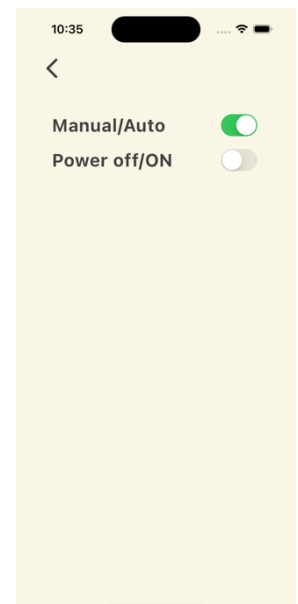
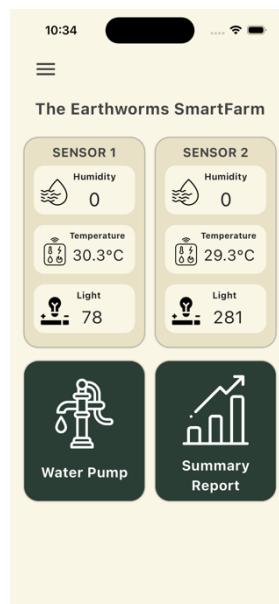
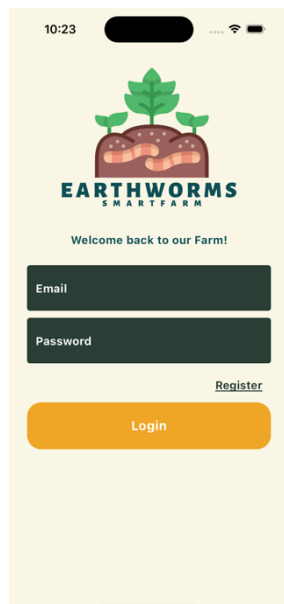
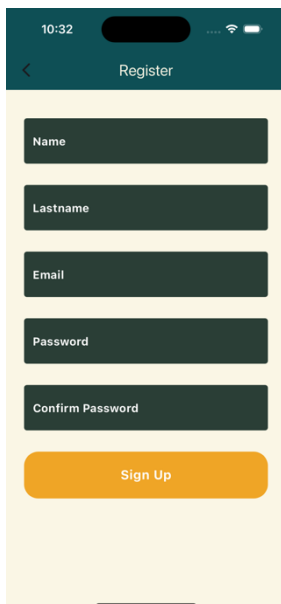
รายละเอียดสิ่งที่พัฒนา

การพัฒนาระบบด้าน IoT

1. Xiaomi flora จะส่งค่าไปยัง Raspberry Pi 4 Model B+ โดยใช้ BLE (Bluetooth Low Energy) ค่าที่จะส่งไปคือ อุณหภูมิและความชื้น
2. Raspberry Pi 4 Model B+ จะประมวลผลว่าอุณหภูมิและความชื้นเหมาะสมหรือไม่ หากค่าอุณหภูมิสูงเกินกว่าเกณฑ์ หรือความชื้นต่ำกว่าเกณฑ์ ระบบจะสั่งเปิดทำงานของปั้มน้ำกรณี user เปิดระบบอัตโนมัติเอาไว้ นอกจากนี้ Raspberry Pi 4 Model B+ จะส่งค่าไปยัง Webserver
3. นอกจากนี้หากกล้องและบอร์ดตรวจจับการบุกรุกของศัตรูได้เดือน จะส่งค่าไปยัง Webserver
4. Webserver รับค่าจาก Raspberry Pi 4 Model B+ เพื่อนำไปส่งให้ Mobile application ต่อไปนอกจากนี้ Webserver ยังประมวลผลค่าที่รับมา หากพบความผิดปกติจะส่ง Line notify เพื่อแจ้งเตือนผู้ใช้งาน
5. Database เก็บค่า Email, Password และข้อมูลอุณหภูมิ ความชื้น
6. Mobile application แสดงค่าที่ได้รับจาก Webserver และสามารถเปลี่ยนโหมดการ เปิด/ปิด ปั้มน้ำ เพื่อความสะดวกแก่ผู้ใช้ โดยจะมีโหมดการใช้งานทั้ง Auto และ Manual

การพัฒนาระบบด้าน Mobile Application

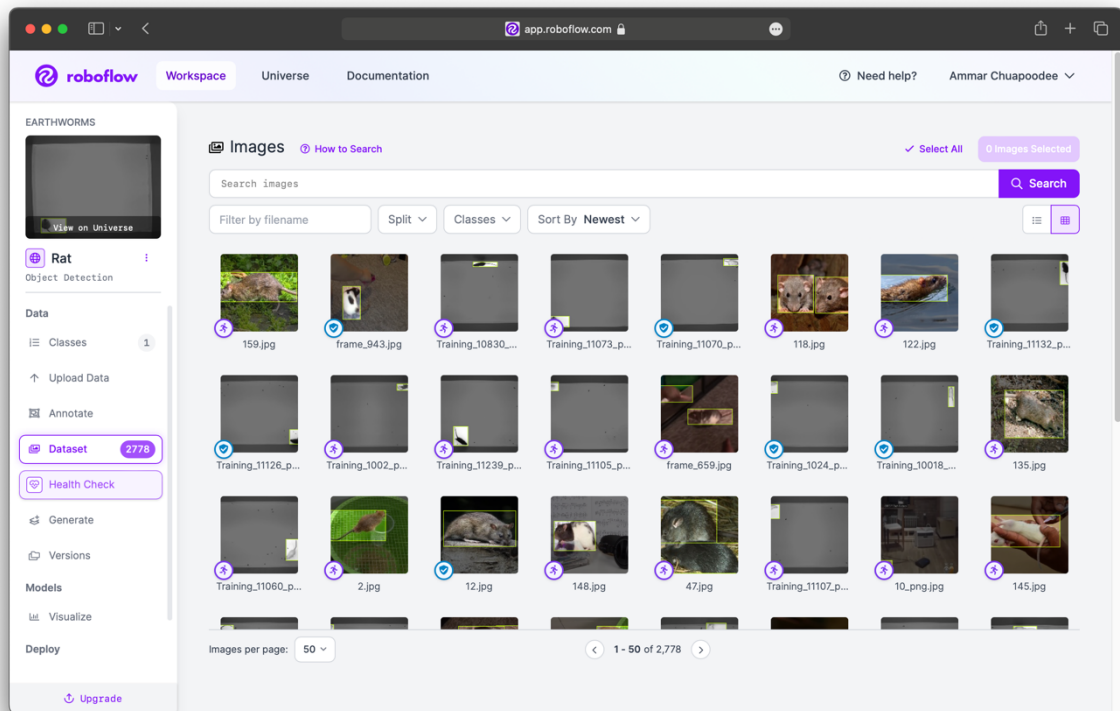
1. หน้า User Interface การใช้งาน



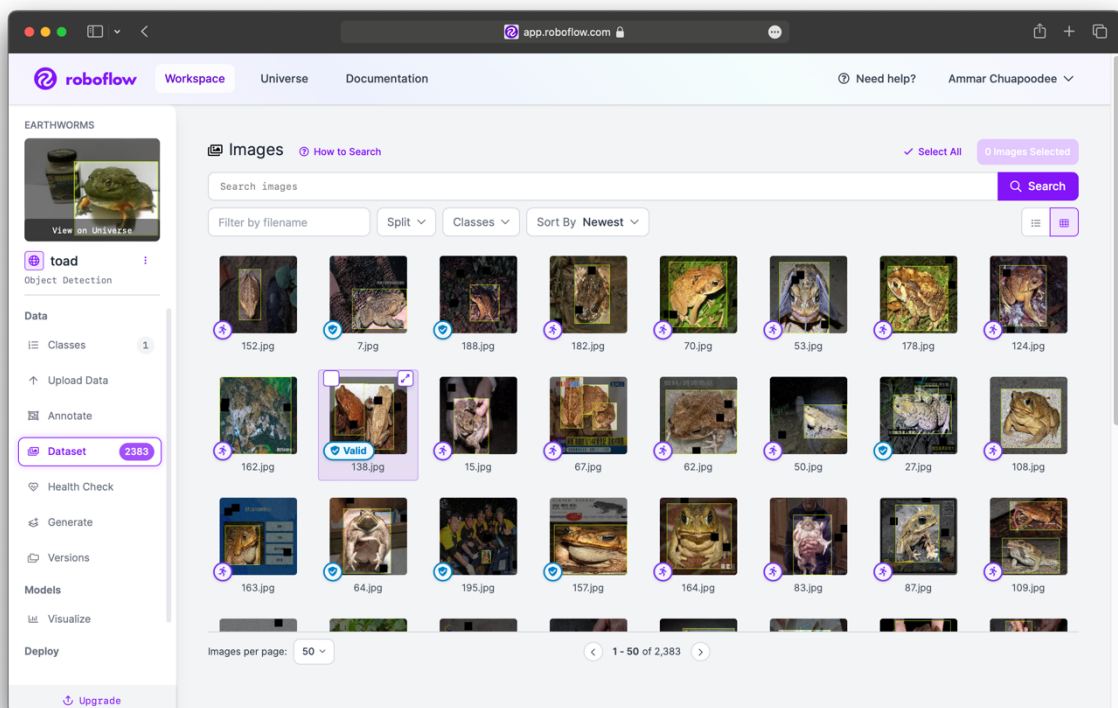
การพัฒนาระบบปัญญาประดิษฐ์

1. ชุดข้อมูล (Dataset) ของระบบ ประกอบด้วยสัตว์ที่เป็นศัตรูของ ไล่เดือน ประกอบด้วย หนู กิ้งก่า และ คางคก

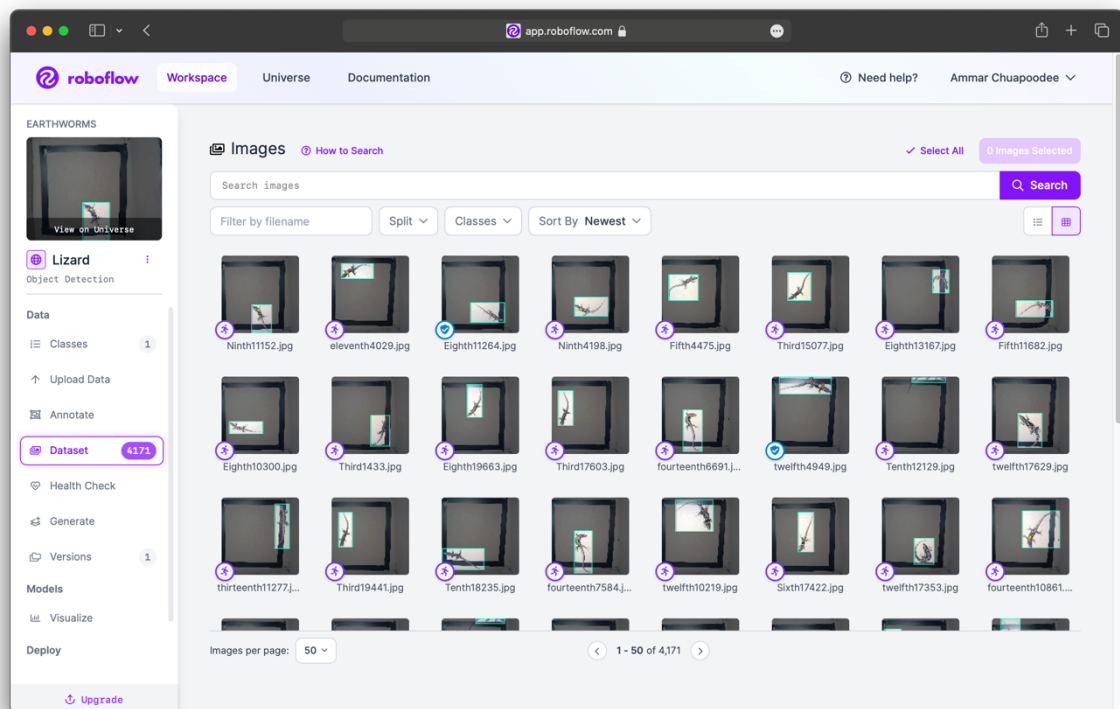
ประกอบไปด้วยสัตว์ที่เป็นศัตรูของไล่เดือน ได้แก่ Class หนู 2,801 รูป



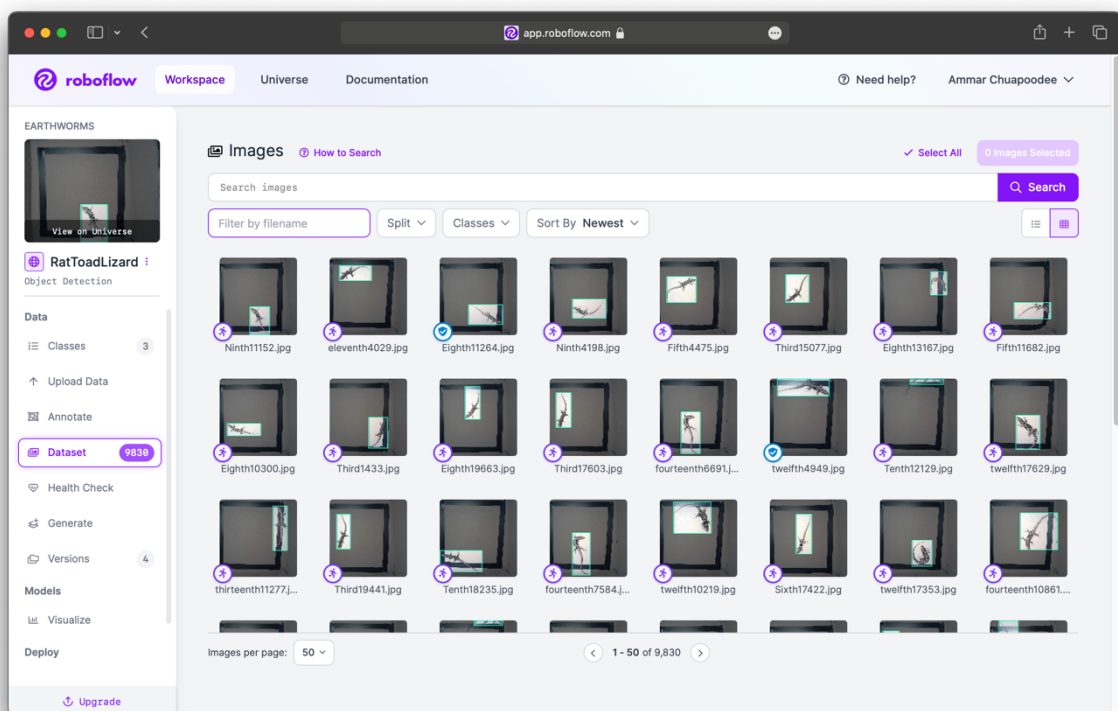
Class คางคก 2,882 รูป



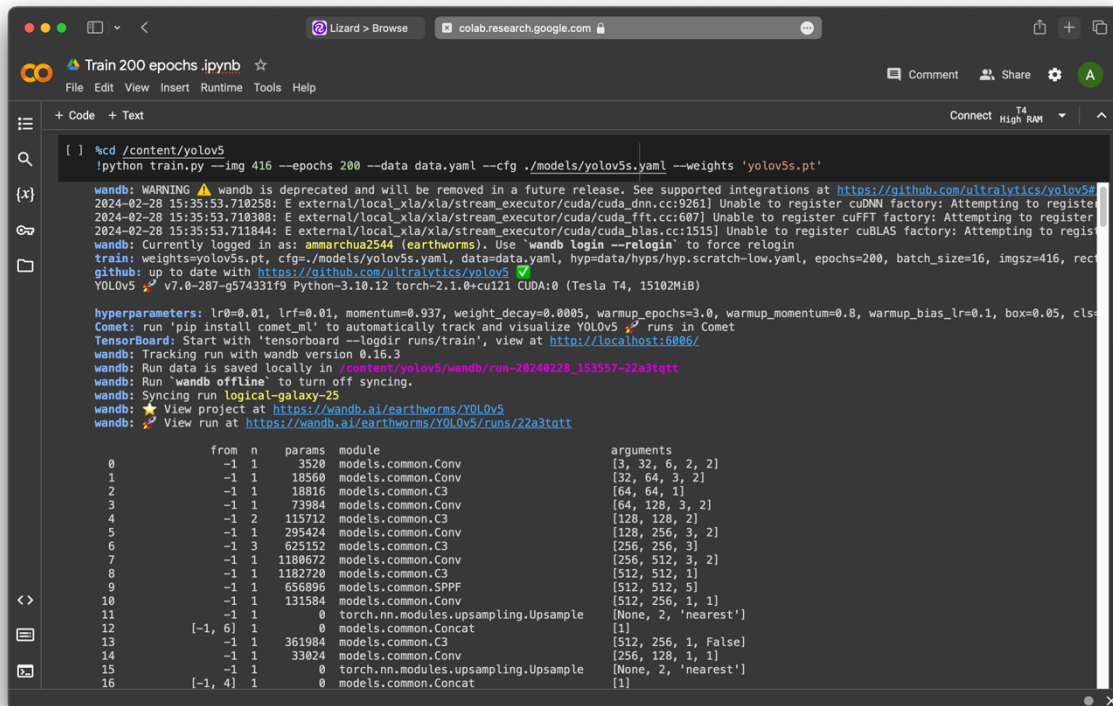
Class กิ่งก่า 4217 รูป



2. Merge dataset และ Classification เพื่อแยกประเภทของสัตว์ทั้ง 3 ชนิด



3. เริ่มการ train โดยใช้ Google Colab



```
[ ] %cd /content/yolov5
!python train.py --img 416 --epochs 200 --data data.yaml --cfg ./models/yolov5s.yaml --weights 'yolov5s.pt'

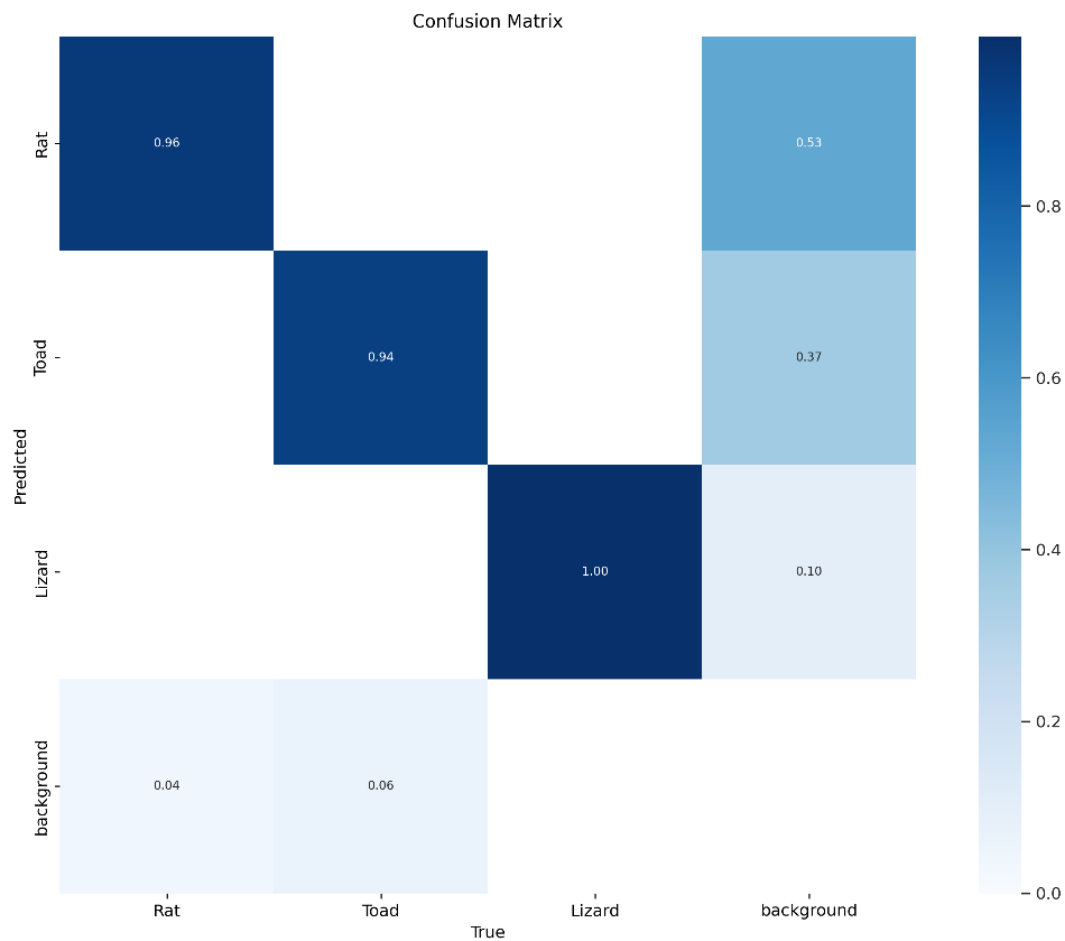
wandb: WARNING ⚠ wandb is deprecated and will be removed in a future release. See supported integrations at https://github.com/ultralytics/yolov5
2024-02-28 15:35:53.710258: E external/local_xla/xla/stream_executor/cuda/cuda_dnn.cc:9261] Unable to register cuDNN factory: Attempting to register
2024-02-28 15:35:53.710308: E external/local_xla/xla/stream_executor/cuda/cuda_fft.cc:607] Unable to register cuFFT factory: Attempting to register
2024-02-28 15:35:53.711844: E external/local_xla/xla/stream_executor/cuda/cuda_blas.cc:1515] Unable to register cuBLAS factory: Attempting to register
wandb: Currently logged in as: ammarchua2544 (earthworms). Use `wandb login --relogin` to force relogin
train: weights=yolov5s.pt, cfg=./models/yolov5s.yaml, data=data.yaml, hyp=data/hyps/hyp.scratch-low.yaml, epochs=200, batch_size=16, imgsz=416, rec
github: up to date with https://github.com/ultralytics/yolov5 ✓
YOLOv5 v7.0-287-g574331f9 Python-3.10.12 torch-2.1.0+cu121 CUDA:0 (Tesla T4, 15102MiB)

hyperparameters: lr0=0.01, lrf=0.01, momentum=0.937, weight_decay=0.0005, warmup_epochs=3.0, warmup_momentum=0.8, warmup_bias_lr=0.1, box=0.05, cls
Comet: run 'pip install comet_ml' to automatically track and visualize YOLOv5 runs in Comet
TensorBoard: Start with 'tensorboard --logdir runs/train', view at http://localhost:6006/
wandb: Tracking run with wandb version 0.16.3
wandb: Run data is saved locally in /content/yolov5/wandb/run-20240228_153557-22a3tqtt
wandb: Run 'wandb offline' to turn off syncing.
wandb: Syncing run logical-galaxy-25
wandb: View project at https://wandb.ai/earthworms/YOLOv5
wandb: View run at https://wandb.ai/earthworms/YOLOv5/runs/22a3tqtt

from n      params  module                                arguments
0         -1  1      3520  models.common.Conv                    [3, 32, 6, 2, 2]
1         -1  1     18560  models.common.Conv                    [32, 64, 3, 2]
2         -1  1     18816  models.common.C3                      [64, 64, 1]
3         -1  1     73984  models.common.Conv                    [64, 128, 3, 2]
4         -1  2    115712  models.common.C3                      [128, 128, 2]
5         -1  1    295424  models.common.Conv                    [128, 256, 3, 2]
6         -1  3    625152  models.common.C3                      [256, 256, 3]
7         -1  1   1180672  models.common.Conv                    [256, 512, 3, 2]
8         -1  1   1182720  models.common.C3                      [512, 512, 1]
9         -1  1    656896  models.common.SPPF                    [512, 512, 5]
10        -1  1   131584  models.common.Conv                    [512, 256, 1, 1]
11        -1  1         0  torch.nn.modules.upsampling.Upsample  [None, 2, 'nearest']
12        [-1, 6] 1         0  models.common.Concat                  [1]
13        -1  1    361984  models.common.C3                      [512, 256, 1, False]
14        -1  1    33024  models.common.Conv                    [256, 128, 1, 1]
15        -1  1         0  torch.nn.modules.upsampling.Upsample  [None, 2, 'nearest']
16        [-1, 4] 1         0  models.common.Concat                  [1]
```

ใช้ Google Colab ร่วมกับ YOLOv5s ในการ train Model จำนวน 200 epochs

3.1 หลังจากที่ Google Colab ดำเนินการครบทั้ง 200 Epochs แล้ว Google Colab จะ Export ไฟล์ Confusion matrix เพื่อแสดงผลการตัดสินใจและทำนายของระบบการเรียนรู้ปัญญาประดิษฐ์



โดยจะเป็นตารางอธิบายการเปรียบเทียบการตัดสินใจ โดยตัดสินใจจาก model ที่เราสร้างขึ้นใน machine learning โดยวัดจากสิ่งที่ทำนาย กับสิ่งที่เกิดขึ้นจริงว่าสัดส่วนเป็นอย่างไร

Confusion Matrix

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

ตารางเปรียบเทียบการตัดสินใจ [3]

True Positive (TP) : สิ่งที่ทำนาย ตรงกับสิ่งที่เกิดขึ้นจริงในกรณีทำนายว่า **จริง** และสิ่งที่เกิดขึ้นก็คือ **จริง**

True Negative (TN) : สิ่งที่ทำนายตรงกับสิ่งที่เกิดขึ้นในกรณีทำนายว่า **ไม่จริง** และสิ่งที่เกิดขึ้นก็คือ **ไม่จริง**

False Positive (FP) : สิ่งที่ทำนายไม่ตรงกับสิ่งที่เกิดขึ้น คือทำนายว่า **จริง** แต่สิ่งที่เกิดขึ้น คือ **ไม่จริง**

False Negative (FN) : สิ่งที่ทำนายไม่ตรงกับที่ที่เกิดขึ้นจริง คือทำนายว่า **ไม่จริง** แต่สิ่งที่เกิดขึ้นคือ **จริง**

โดยค่าเหล่านี้สามารถนำมาหาผลลัพธ์เพื่อประเมินประสิทธิภาพการทำนายของ Model เราได้ดังนี้

1. Accuracy (ค่าความถูกต้อง)

ค่าความถูกต้องที่ model หายได้ตรงกับสิ่งที่เกิดขึ้นจริง

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} = \frac{0.96 + 1.94}{0.96 + 1.00 + 1.94 + 0.10}$$

ผลจากการคำนวณเท่ากับ 0.725 หรือ 72.5%

2. Precision (ค่าความแม่นยำ)

เป็นค่าที่แสดงถึง การทำนายที่ถูกต้องว่าจริง (TP) กับ การทำนายที่เกิดขึ้นจริง แต่สิ่งที่เกิดขึ้นคือไม่จริง (FP)

$$Precision = \frac{TP}{TP + FP} = \frac{0.96}{0.96 + 1.00}$$

ผลจากการคำนวณเท่ากับ 0.498 หรือ 48.9%

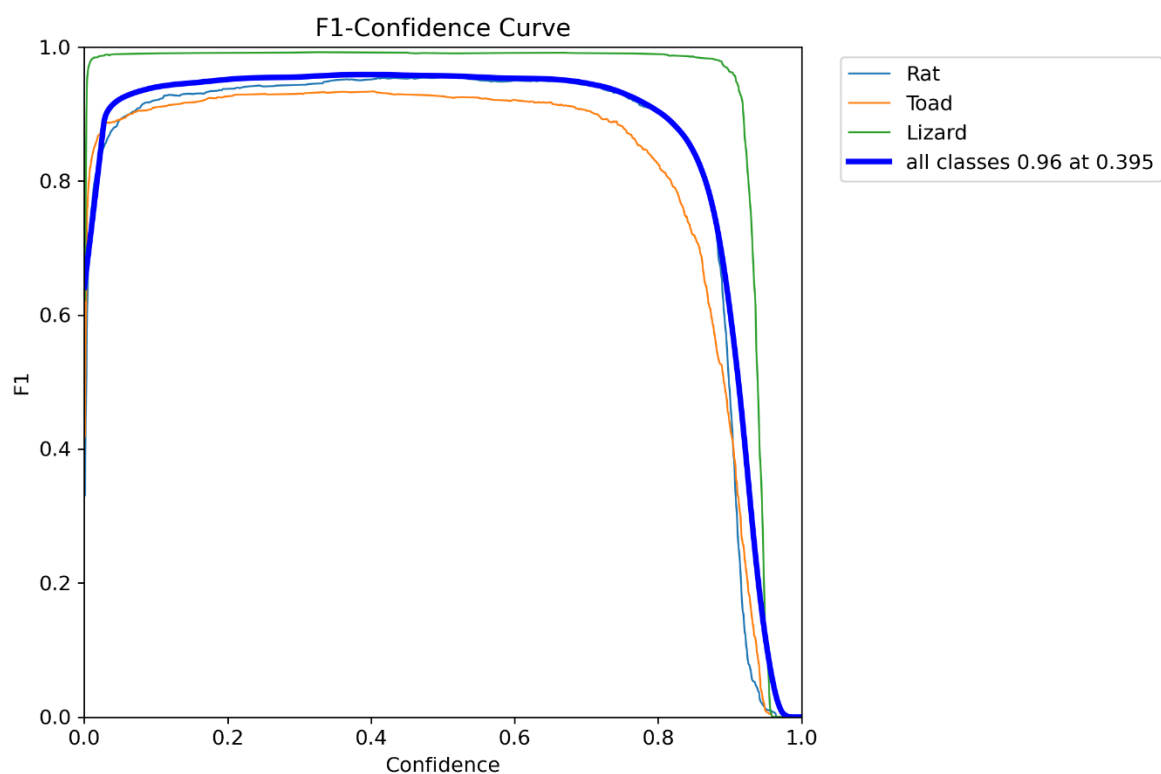
3. Recall

ความถูกต้องของการทำนายว่าจะเป็น “จริง” เทียบกับจำนวนครั้งของเหตุการณ์ ทั้งทำนาย และ เกิดขึ้นว่า “เป็นจริง”

$$Recall = \frac{TP}{TP + FN} = \frac{0.96}{0.96 + 0.10}$$

ผลจากการคำนวณเท่ากับ 0.905 หรือ 90.5%

3.2 กราฟ F1-SCORE คือกราฟที่วัดความสามารถของ Model โดยสรุปมาจาก Precision และ Recall



4. ประเมินผลการทำนายของระบบปัญญาประดิษฐ์



ประเมินผลการทำนายของปัญญาประดิษฐ์โดยการนำภาพที่ไม่ได้อยู่ใน Dataset ให้ปัญญาประดิษฐ์ทำนาย เพื่อดูผลการทำนาย

เอกสารอ้างอิง

[1] สืบค้นด้วยตัวเองโดยการไปเยี่ยมชมสถานที่จริง ณ ฟาร์มรัก FARM RAK จังหวัดนนทบุรี เพื่อเก็บข้อมูล และปัญหาจากผู้ประกอบการ

[2] Fulfill Agri ฟาร์มไส้เดือนหัวใจสีเขียว.//ปัญหาอดีตในการเลี้ยงไส้เดือน.//สืบค้นเมื่อวันที่ 19 พฤษภาคม 2567,/จาก

/https://fulfillagri.wordpress.com/2019/08/06/%E0%B8%9B%E0%B8%B1%E0%B8%8D%E0%B8%AB%E0%B8%B2%E0%B8%A2%E0%B8%AD%E0%B8%94%E0%B8%AE%E0%B8%B4%E0%B8%95%E0%B9%83%E0%B8%99%E0%B8%81%E0%B8%B2%E0%B8%A3%E0%B9%80%E0%B8%A5%E0%B8%B5%E0%B9%89%E0%B8%A2%E0%B8%87/

[3] Pagon Gatchalee.// Confusion Matrix เครื่องมือสำคัญในการประเมินผลลัพธ์ของการทำนาย ใน Machine learning.//สืบค้นเมื่อ 7 พฤษภาคม 2567,/จาก

/https://medium.com/@pagongatchalee/confusion-matrix-เครื่องมือสำคัญในการประเมินผลลัพธ์ของการทำนาย-ในmachine-learning-fba6e3f9508c



