โครงงานรายวิชา CP413008 Machine Learning Engineering for Production

วิศวกรรมการเรียนรู้ของเครื่องสำหรับการผลิต (20 คะแนน)

ภาคเรียนที่ 1 ปีการศึกษา 2568

กำหนดส่งก่อน 15 ตุลาคม 2568 นำเสนอ ___ ตุลาคม 2568

วัตถุประสงค์

เพื่อให้นักศึกษาสามารถประยุกต์ใช้ความรู้และเครื่องมือด้าน MLOps โดยใช้ MLflow ในการสร้างระบบ Machine Learning ที่สามารถติดตามผล, รันซ้ำได้ (Reproducible), และพร้อมสำหรับการนำไปใช้งานจริง (Production)

1. คำสั่งและขอบเขตของโครงงาน

ให้นักศึกษาแบ่งกลุ่ม กลุ่มละ 5 - 7 คน สร้างและนำเสนอระบบ Machine Learning Pipeline แบบครบวงจร สำหรับแก้ปัญหาการจำแนกประเภท (Classification) โดยมีรายละเอียดดังนี้

1.1. โจทย์และชุดข้อมูล (Problem & Dataset)

เลือกโจทย์: การจำแนกประเภทของ รูปภาพ หรือ ข้อความ

ชุดข้อมูล: ใช้ชุดข้อมูลสาธารณะ (Public Dataset) ที่มี อย่างน้อย 5 คลาส (10,000 records)

เป้าหมาย: สร้าง Service ที่สามารถรับข้อมูลใหม่ (รูปภาพหรือข้อความ) และทำการจำแนกประเภทได้อย่างถูกต้อง

1.2. การจัดการ ML Lifecycle ด้วย Python และ MLflow

ให้นักศึกษาสร้างชุดของโค้ด เพื่อจัดการกระบวนการต่างๆ ใน ML Lifecycle และใช้ MLflow Tracking เพื่อบันทึกผลในทุกขั้นตอน โดยต้องมีกระบวนการดังนี้:

Data Validation: เขียนสคริปต์สำหรับตรวจสอบคุณภาพและความถูกต้องของข้อมูล

ต้องสามารถตรวจจับความผิดปกติของข้อมูล (Anomalies) ได้

Log ผลลัพธ์ เช่น สถิติของข้อมูล, รายงานการตรวจสอบ, หรือ Schema ของข้อมูล

Data Preprocessing: เขียนสคริปต์สำหรับเตรียมข้อมูลที่จะใช้เทรนโมเดล ต้องใช้กระบวนการเดียวกันนี้กับข้อมูลใหม่ (เพื่อป้องกัน Training-Serving Skew) Log สิ่งที่จำเป็น เช่น ตัวแปลงข้อมูล (Scaler/Encoder object) เป็น Artifacts ใน MLflow

Model Training: เขียนสคริปต์สำหรับเทรนโมเดล (สามารถใช้ Framework ใดก็ได้ เช่น TensorFlow/Keras, PyTorch, Scikit-learn)

Log ข้อมูลสำคัญ โดยใช้ MLflow Tracking:

Parameters: Log Hyperparameters ที่ใช้ในการทดลองทั้งหมด (เช่น learning rate, batch size, architecture)

Metrics: Log Metrics ของโมเดล (เช่น Accuracy, Loss, F1-score) ในระหว่างการเทรน

Artifacts: Log โมเดลที่เทรนเสร็จแล้ว (Model File) และไฟล์อื่นๆ ที่เกี่ยวข้อง

Model Evaluation & Registration:

เขียนสคริปต์สำหรับประเมินประสิทธิภาพของโมเดลที่เทรนใหม่กับข้อมูลทดสอบ (Test Set)

กำหนดเกณฑ์การประเมินที่ชัดเจน (เช่น Accuracy ต้องมากกว่า 85%) เพื่อตัดสินใจว่าโมเดลดีพอที่จะใช้งานหรือไม่

หากโมเดลผ่านเกณฑ์ ให้ใช้ MLflow Model Registry ในการลงทะเบียนโมเดล (Register Model) และกำหนดเวอร์ชันใหม่

1.3. Experiment Tracking & Model Management (MLflow)

- ใช้ MLflow Tracking Server เพื่อเป็นศูนย์กลางในการเก็บข้อมูลการทดลอง
- แสดงให้เห็นถึงการทดลองเปรียบเทียบผลลัพธ์ของ Runs ที่มีการปรับ Hyperparameter ที่แตกต่างกันอย่างน้อย 2-3 รูปแบบผ่าน MLflow UI
- ใช้ MLflow Model Registry ในการจัดการเวอร์ชันของโมเดล และเปลี่ยนสถานะ (Stage) ของโมเดล เช่น จาก Staging ไปเป็น Production

1.4. CI/CD Pipeline (GitHub Actions)

สร้าง Workflow ด้วย GitHub Actions เพื่อทำให้กระบวนการพัฒนาเป็นอัตโนมัติ

สิ่งที่ต้องทำ: Workflow ควรถูก Trigger เมื่อมีการ Push Code ไปยัง main branch หรือเมื่อมีการสร้าง Pull Request และต้องทำงานอย่างน้อย 1 อย่างต่อไปนี้:

Code Linting: ตรวจสอบคุณภาพของโค้ด

Unit Testing: เขียนและรัน Unit Test สำหรับฟังก์ชันที่สำคัญ (เช่น ฟังก์ชันใน Preprocessing)

1.5. Deployed Model API

นำโมเดลที่ถูก Push ไป Deploy เป็น API Service ที่ทำงานได้จริง

การสาธิต: ต้องสามารถส่งข้อมูล (เช่น รูปภาพหรือข้อความ) ไปยัง API Endpoint และได้รับผลลัพธ์การทำนายกลับมาอย่างถูกต้อง

2. สิ่งที่ต้องส่ง (Deliverables)

Source Code: Repository บน GitHub ที่มีโค้ดทั้งหมดของโปรเจค, Dockerfile (ถ้ามี), และ GitHub Actions workflow

Presentation & Demo: นำเสนอ (15 นาที) และการสาธิตการทำงานของระบบ

Project Report: เอกสารรายงานสรุปโปรเจค (ความยาว 3-5 หน้า) ที่อธิบายถึง:

- System Architecture: แผนภาพและคำอธิบายสถาปัตยกรรมของระบบทั้งหมด
- Technical Decisions: เหตุผลในการเลือกใช้โมเดล, Hyperparameters, และการออกแบบ Pipeline
- Experiment Results: สรุปผลการทดลองและผลลัพธ์จาก MLflow
- Monitoring Strategy: อธิบายแผนการเฝ้าระวังโมเดลในอนาคต เช่น จะตรวจจับ Data Drift และ Concept Drift ได้อย่างไร และจะ Trigger การ Retrain โมเดลเมื่อใด
- Problems & Challenges: ปัญหาที่พบเจอและแนวทางการแก้ไข

3. เกณฑ์การให้คะแนน (20 คะแนน)

ส่วนที่ 1: Data & Pipeline Foundation (4 คะแนน)

(2 คะแนน) Data Ingestion & Validation: นำเข้าข้อมูลอย่างเหมาะสม และตรวจสอบ Data Anomalies

(2 คะแนน) Preprocessing: มีการ Preprocessing ที่ถูกต้องและสามารถป้องกัน Training-Serving Skew ได้

ส่วนที่ 2: Modeling & Evaluation (5 คะแนน)

(2 คะแนน) Model Training & Architecture: เลือกใช้สถาปัตยกรรมโมเดลที่เหมาะสมกับโจทย์ และมีโค้ดการเทรนโมเดลที่สมบูรณ์

(2 คะแนน) Pipeline Implementation: สร้าง Pipeline ที่มี Component ครบถ้วน และสามารถทำงานได้จริงตั้งแต่ต้นจนจบ

(1 คะแนน) Model Evaluation: กำหนดเกณฑ์การประเมินโมเดลใน Evaluator Component ที่ชัดเจนและสมเหตุสมผล เพื่อใช้ในการตัดสินใจ Deploy โมเดล

ส่วนที่ 3: Deployment & Automation (5 คะแนน)

(3 คะแนน) Model Deployment & Serving: สามารถ Deploy โมเดลเพื่อสร้างเป็น API Service ที่ทำงานได้จริง

(2 คะแนน) CI/CD Automation: มีการใช้ GitHub Actions สร้าง Workflow สำหรับ Linting หรือ Testing

ส่วนที่ 4: Governance & Presentation (6 คะแนน)

(2 คะแนน) Version Control & Experiment Tracking: ใช้ MLflow ในการ Log และเปรียบเทียบผลการทดลองได้อย่างถูกต้อง

(2 คะแนน) Presentation, Demo & Report: นำเสนอและสาธิตการทำงานของระบบได้ชัดเจน มี Test Case เพื่อทดสอบโมเดลอย่างน้อย 3 Test Cases และมีเอกสารรายงานที่ครบถ้วนตามหัวข้อที่กำหนด

(2 คะแนน) รับ Test Case จากอาจารย์วันนำเสนอ: รับ Test Case จากอาจารย์วันนำเสนอ เพื่อทดสอบการตรวจจับความผิดปกติของข้อมูล (Anomalies)