

Machine Learning → Evaluation → Deployment

เอกสารฉบับนี้ออกแบบมาเพื่อให้ทุกกลุ่มนักศึกษาทำงานตามกระบวนการเดียวกันตลอดทั้งโครงการ ตั้งแต่การทำความเข้าใจข้อมูล การสร้างโมเดล การประเมินผล การวิเคราะห์ข้อผิดพลาด

ไปจนถึงการนำโมเดลไปใช้งานจริงผ่าน Web Application

โดยแต่ละกลุ่มจะแตกต่างกันเพียงชนิดของข้อมูลและรายละเอียดเล็กน้อยเท่านั้น

เพื่อให้เกิดความยุติธรรมและสามารถเปรียบเทียบผลลัพธ์ข้ามกลุ่มได้

ภาพรวมกระบวนการ (Workflow เดียวกันทุกกลุ่ม)

นักศึกษาทุกกลุ่มต้องดำเนินงานตาม 6 ขั้นตอนเดียวกัน:

1. Dataset Understanding
2. Preprocessing
3. Baseline Model Training
4. Evaluation
5. Error Analysis
6. Web Deployment

ขั้นตอนที่ 1: การทำความเข้าใจ Dataset (Dataset Understanding)

แต่ละกลุ่มต้องอธิบาย dataset ที่ใช้ โดยระบุ:

- ภาษา (ไทย / อังกฤษ)
- ประเภทข้อความ (รีวิว ความคิดเห็น ข่าว)
- ชนิดของ label
- จำนวนข้อมูล

- มีการแบ่ง clean/noisy หรือ easy/hard หรือไม่

ขั้นตอนที่ 2: การเตรียมข้อมูล (Preprocessing)

ทุกกลุ่มต้องทำ preprocessing ขั้นต่ำ และอธิบายเหตุผลของแต่ละขั้นตอน

ตัวอย่างที่อนุญาตให้ทำ:

- การจัดการช่องว่าง (whitespace normalization)
- การแปลงเป็นตัวพิมพ์เล็ก (lowercase) สำหรับภาษาอังกฤษ
- การ normalize ข้อความพื้นฐาน

ไม่อนุญาตให้ทำ over-cleaning เช่น การลบ emoji หรือ slang ทั้งหมด

ขั้นตอนที่ 3: การสร้างโมเดลพื้นฐาน (Baseline Model Training)

ทุกกลุ่มต้องใช้โมเดลพื้นฐานเหมือนกัน:

- TF-IDF (word-level)
- Logistic Regression ใช้โมเดลของ ML อื่นได้ หรือเปรียบเทียบหลายๆโมเดล แล้วเลือกโมเดลที่ได้ผลลัพธ์ดี ไป Deploy
- ตั้งค่า class_weight = 'balanced'

ข้อกำหนด:

- แบ่ง train / test อย่างเหมาะสม
- train โมเดลพื้นฐาน
- บันทึกโมเดลเป็นไฟล์ .joblib

ขั้นตอนที่ 4: การประเมินผล (Evaluation)

ทุกกลุ่มต้องรายงานผลอย่างน้อย:

- Accuracy
- Macro-F1
- Confusion Matrix

นอกจากนี้ ต้องแสดงตัวอย่างที่โมเดลทำนายผิดอย่างน้อย 10 ตัวอย่าง

ขั้นตอนที่ 5: การวิเคราะห์ข้อผิดพลาด (Error Analysis)

ทุกกลุ่มต้องจัดกลุ่มข้อผิดพลาดอย่างน้อย 3 ประเภท เช่น:

- typo หรือ noise
- mixed signal หรือความกำกวม
- negation หรือ sarcasm
- domain shift (ถ้ามี)

ต้องวิเคราะห์ว่า error ประเภทใดเกิดบ่อยที่สุด และเสนอแนวทางแก้ไขอย่างน้อย 1 แนวทาง

ขั้นตอนที่ 6: การนำไปใช้งานจริง (Web Deployment)

ทุกกลุ่มต้อง deploy ระบบเป็น Web Application ที่มีโครงสร้างเดียวกัน โดยเว็บต้องมี:

- ช่องกรอกข้อความ (หรือ Headline + Body สำหรับข่าว)
- ปุ่ม Predict

- แสดงผล Label และ Confidence
- แสดง Model Version
- แสดง Latency (ms)
- ปุ่มตัวอย่างข้อความ
- หน้าดูตัวอย่าง Error

Backend ต้องมี API อย่างน้อย:

- GET /health
- GET /model/info
- POST /predict













การแบ่งกลุ่มและ Dataset

กลุ่ม 1-4: Thai Sentiment Dataset

กลุ่ม 5-7: English Review Dataset

กลุ่ม 8-10: Thai News Topic Dataset

กลุ่ม 11-12: AG news_thai

-  1.synthetic_wisesight_like_thai_sentiment_5000.csv
-  2.synthetic_wisesight_like_thai_sentiment_hard_5000.csv
-  3.synthetic_wongnai_like_thai_reviews_5000.csv
-  4.synthetic_wongnai_like_thai_reviews_5000_hard.csv
-  5.synthetic_imdb_like_reviews_5000.csv
-  6.synthetic_imdb_like_reviews_3class_hard_5000.csv
-  7.synthetic_netflix_like_thai_reviews_5000.csv
-  8.synthetic_netflix_like_thai_reviews_3class_hard_5000.csv
-  9.synthetic_amazon_like_reviews_subset_5000.csv
-  10.synthetic_amazon_like_reviews_3class_hard_5000.csv
-  11.agnews_thai_test_hard.csv
-  12.agnews_thai_train_easy.csv

หมายเหตุ: ทุกกลุ่มใช้กระบวนการเดียวกัน แต่ต่างเฉพาะข้อมูลและรายละเอียดเล็กน้อย

สิ่งที่ต้องส่ง (Deliverables)

- Source code (GitHub repository หรือ ZIP)
- ไฟล์โมเดลที่ train แล้ว (.joblib)
- requirements.txt
- README อธิบายวิธีรันและ deploy
- รายงานสั้น 2-4 หน้า

เกณฑ์การให้คะแนน (Grading Rubric: 100 คะแนน)

- ความเข้าใจ dataset: 10 คะแนน
- เหตุผลของ preprocessing: 10 คะแนน
- ความถูกต้องของการ train โมเดล: 20 คะแนน
- การประเมินผล: 15 คะแนน

- คุณภาพของ Error Analysis: 20 คะแนน
- ความสมบูรณ์ของ Web Deployment: 25 คะแนน

ตัวอย่างหน้าจอ

The screenshot displays a web application for sentiment analysis. At the top, there are tabs for 'Sentiment' and 'Topic', with 'Sentiment' selected. A language dropdown menu is set to 'ไทย'. Below the tabs, there's a 'Text:' input field with a toggle for 'A / B comparison'. The input text is 'ชิรส์เรื่องนี้สนุกดี คอเจนนี้นับว่าทักมมสุดๆ!! 😄'. Below the input field is an 'Analyze' button. The results section, titled 'Model A and Model B predicted:', shows two columns. The left column for Model A (sentiment_v1_tfidf_lr - v1.0) predicts 'NEGATIVE' with a confidence of 0.82 and a latency of 5 ms. The right column for Model B (sentiment_v2_char_tfidf_lr - v2.1) predicts 'NEGATIVE' with a confidence of 0.84 and a latency of 6 ms. Both models list 'สนุก', 'คอเจบ', and 'ทักมม' as important words. At the bottom, there are sentiment distribution bars: Model A shows ~16% Positive, ~16% Neutral, and ~68% Negative; Model B shows ~23% Positive, ~23% Neutral, and ~54% Negative.

Sentiment Topic ไทย ▾

Text: A / B comparison ☒

Thai slang Typos Sarcasm Neutral-ish ▾

ชิรส์เรื่องนี้สนุกดี คอเจนนี้นับว่าทักมมสุดๆ!! 😄

Analyze

Model A and Model B predicted:

Model A predicted:

NEGATIVE Confidence 0.82

Model: sentiment_v1_tfidf_lr - v1.0
Latency: 5 ms

Important words:

สนุก คอเจบ ทักมม

• Positive • Neutral ~16% Negative

Model B predicted:

NEGATIVE Confidence 0.84

Model: sentiment_v2_char_tfidf_lr - v2.1
Latency: 6 ms

Important words:

สนุก คอเจบ ฝ ทักมม

• Positive • Neutral ~23% Negative