



Data Science for Taffy Fondué Dataset

2110403 Data Science and Data Engineering (DSDE - CEDT)

About



Traffy Fondu

- ระบบรับแจ้งปัญหาในชีวิตประจำวันของประชาชน เช่น ปัญหาน้ำท่วม, ถนนชำรุด ผ่าน LINE (Chatbot) ซึ่งทำหน้าที่เป็น ticketing system ส่งเรื่องต่อไปให้หน่วยงานรัฐดำเนินการแก้ไข
- ข้อมูลได้แก่ ประเภทปัญหา, ข้อความร้องเรียน, พิกัด, หน่วยงาน, เวลาแจ้ง, สถานะงาน, ฯลฯ รวมทั้งสิ้นกว่า 700,000 รายการ

สิ่งที่ทำใน Project

เนื่องจากระบบเดิม ไม่มีการจัดลำดับความสำคัญของปัญหาที่เหมาะสม ทีมของเราจึงพัฒนาทั้งหมด 2 ส่วน

1. **Problem Priority Recommender** เพื่อประเมินความเร่งด่วนของแต่ละเรื่อง โดยพิจารณาจาก 4 องค์ประกอบหลักคือ Severity, Urgency, Location Risk, Frequency
2. สร้าง Dashboard แสดง Hotspots แบบเรียลไทม์

Step 1: Cleaning data

Raw data from traffy fondu

Data columns (total 16 columns):			
#	Column	Non-Null Count	Dtype
0	ticket_id	778254	non-null object
1	type	786929	non-null object
2	organization	786455	non-null object
3	comment	778254	non-null object
4	photo	786911	non-null object
5	photo_after	641309	non-null object
6	coords	787026	non-null object
7	address	778254	non-null object
8	subdistrict	786460	non-null object
9	district	786465	non-null object
10	province	786831	non-null object
11	timestamp	787026	non-null object
12	state	787026	non-null object
13	star	274097	non-null float64
14	count_reopen	787026	non-null int64
15	last_activity	787026	non-null object

สิ่งที่ทำ

- เลือกแต่ DATA ที่มี PROVINCE เป็น “กรุงเทพมหานคร”
- Drop คอลัมน์ที่ไม่จำเป็นออก
 - ลบคอลัมน์ที่มีค่าเป็น NULL จำนวนมากหรือ ไม่ช่วยในการวิเคราะห์ได้แก่ STAR, TYPE, PHOTO, PHOTO_AFTER
- คัดกรองข้อมูลที่ไม่มีคุณภาพออก
 - ลบแถวที่ COMMENT ว่าง เนื่องจากไม่สามารถประเมิน SEVERITY หรือ URGENCY ได้
 - CLEAN COMMENT ด้วยการนำอิโมจิหรือข้อความที่ไม่จำเป็นออก
- แยก COLUMN COORDS ออกเป็น LATITUDE และ LONGITUDE
- จัดการข้อมูล DISTRICT/SUBDISTRICT ที่หายไป
 - เติมข้อมูลแขวง – เขตบางส่วนที่หายไป จึงทำการ EXTRACT แขวง/เขตจาก ADDRESS
 - ถ้า EXTRACT แล้วยังเป็น NULL อยู่จะ DROP ROW นั้นทิ้ง
- เพิ่มคอลัมน์ชื่อเขตภาษาอังกฤษ
 - ใช้ข้อมูลชื่อเขตภาษาไทยที่ผ่านการแก้ให้สะกดถูกต้องแล้ว
 - ทำการ MAP → ENGLISH DISTRICT NAMES ให้เป็นมาตรฐานเดียวกัน

ผลลัพธ์ที่ได้

- ได้ข้อมูลที่พร้อมใช้งานประมาณ **650,000 RECORDS**
- แบ่งชุดข้อมูลออกเป็น: TRAINING SET และ TESTING SET



Step 2: Web Scraping

OPENSTREETMAP API

- ดึงข้อมูลจาก OPENSTREETMAP API
 - OPEN SOURCE รวบรวมข้อมูลเกี่ยวกับสถานที่ทั่วโลก
 - เลือกดึงเฉพาะสถานที่สำคัญในกรุงเทพ
 - ได้แก่ โรงพยาบาล, โรงเรียน, และศาสนสถาน
 - พร้อมจำแนกตาม DISTRICT
 - ถ้า RECORD ไหนไม่มีชื่อ แต่ตรวจสอบแล้วเป็นสถานที่จริง
 - ทำการ IMPUTE ชื่อใหม่ เป็น “เขต ประเภทสถานที่ ลำดับที่” เช่น “PHAYATHAI SCHOOL 1”
 - ใช้ข้อมูลที่ได้มาประกอบการคำนวณ LOCATION RISK SCORE
 - หากเกิดเหตุใกล้สถานที่สำคัญ มีผลกระทบต่อคนจำนวนมาก คะแนนความเร่งด่วนสูง

ผลลัพธ์ที่ได้

- ได้ข้อมูลที่พร้อมใช้งานประมาณ 1,500 RECORDS
 - แต่ละ RECORD ประกอบด้วย
 - DISTRICT, NAME, LATITUDE, LONGITUDE



Step 3: classify type

สิ่งที่ทำ

เนื่องจาก COLUMN ‘TYPE’ ใน DATA เดิม มีค่าเป็น NULL เ酵ะ เราจึง DROP ไป
ในขั้นตอนการ CLEAN DATA และ DEFINED TYPE ขึ้นมาใหม่ 10 ประเภท ได้แก่

- โครงสร้างพื้นฐานและสารสนับสนุน
- การจราจรและสิ่งกีดขวาง
- ความปลอดภัยสาธารณะ
- ความสะอาดและสุขอนามัย
- ปัญหาน้ำท่วมและการระบายน้ำ
- ต้นไม้และสัตว์
- ป้ายและข้อมูลสาธารณะ
- มวลภาพและเสียงรบกวน
- เรื่องทั่วไปและการบริการ
- ข้อเสนอแนะเรื่องอื่น ๆ

โดยใช้ JOEDDAV/XLM-ROBERTA-LARGE-XNLI MODEL

- ซึ่งเป็นโมเดล ZERO-SHOT CLASSIFICATION สำหรับหลายภาษา (MULTILINGUAL) รองรับทั้งภาษาไทยและภาษาอังกฤษ
- ใช้ในการแบ่งประเภทปัญหา
 - โดยส่ง COLUMN ‘COMMENT’ ไปให้ MODEL ช่วย MAPPING

Step 4: calculate priority score

สูตรการคำนวณ

- PRIORITY SCORE = (**SEVERITY SCORE** * 35%) + (**URGENCY SCORE** * 35%) + (LOCATION RISK * 15%) + (FREQUENCY SCORE * 15%)

SEVERITY SCORE : ค่าความรุนแรงของเหตุการณ์

URGENCY SCORE : ค่าความเร่งด่วนในการแก้ไข

1. RULE BASED KEYWORD

- หากใน COMMENT มี KEYWORD เหล่านี้ ระบบ RETURN ทันทีโดยไม่ส่งต่อให้ AI
- ประเภท KEYWORD
 - CRITICAL**
 - คำที่เกี่ยวกับสถานการณ์ที่ต้องแก้ไขโดยด่วน เช่น ไฟไหม้, ระเบิด
 - RETURN SEVERITY และ URGENCY SCORE เป็น 1.0 เสมอ
 - IGNORE**
 - คำที่ไม่มีประโยชน์ต่อการวิเคราะห์ เช่น ขอบคุณ, น่ารัก, สวายงาม
 - RETURN SEVERITY และ URGENCY SCORE เป็น 0.0 เสมอ
 - SPAM**
 - COMMENT สั้นมากและไม่มีข้อความสำคัญ
 - RETURN SEVERITY และ URGENCY SCORE เป็น 0.1
 - URGENT KEYWORD**
 - คำที่เกี่ยวกับสถานการณ์ที่ควรรีบแก้ไขแต่ไม่วิกฤตเท่า CRITICAL เช่น ด่วน, อันตราย, รถชน, น้ำท่วม ยังต้องส่งให้ MODEL ประเมินคะแนน
 - แต่จะ BOOST SEVERITY และ URGENCY SCORE เพิ่มเล็กน้อยในตอนท้าย

2. AI CLASSIFICATION MODEL

- ใช้ JOEDDAV/XLM-ROBERTA-LARGE-XNLI MODEL
- MAP COMMENT ให้ตรงตามแต่ละระดับและคะแนนที่กำหนดไว้
 - SEVERITY** : ความรุนแรงของเหตุการณ์ต่อความปลอดภัย
 - แบ่งเป็น 4 ระดับ แต่ละระดับจะได้คะแนนดังนี้
 - "LIFE THREATENING" 1.00 คะแนน
 - "HAZARDOUS" 0.75 คะแนน
 - "NUISANCE" 0.40 คะแนน
 - "SAFE" 0.10 คะแนน
 - URGENCY** : ระดับความเร่งด่วนในการแก้ไข
 - แบ่งเป็น 4 ระดับ แต่ละระดับจะได้คะแนนดังนี้
 - "EMERGENCY" 1.00 คะแนน
 - "ACTION NEEDED" 0.75 คะแนน
 - "QUEUEABLE" 0.40 คะแนน
 - "IGNORE" 0.10 คะแนน

Step 4: calculate priority score

สูตรการคำนวณ

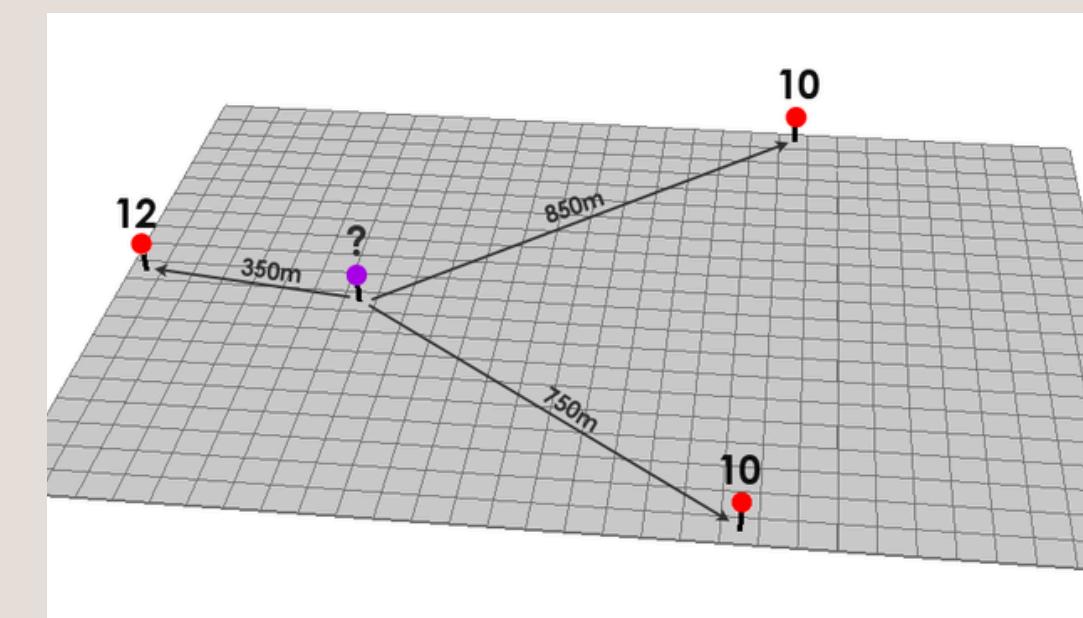
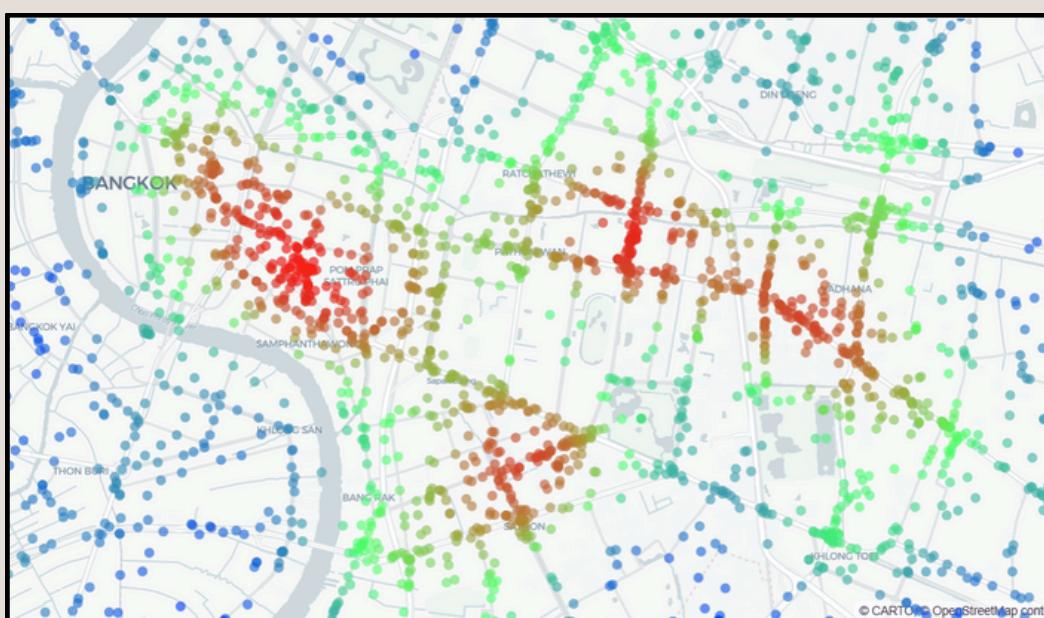
- PRIORITY SCORE = (SEVERITY SCORE * 35%) + (URGENCY SCORE * 35%) + (LOCATION RISK * 15%) + (FREQUENCY SCORE * 15%)

LOCATION RISK SCORE : ระยะห่างจากสถานที่สำคัญ ยิ่งใกล้เคียงสถานที่สำคัญหลายแห่ง ค่ายิ่งสูง

1. คำนวนหาค่า IDW (INVERSE DISTANCE WEIGHTING) ซึ่งจะผกผันกับระยะทางระหว่างจุดเกิดเหตุกับสถานที่สำคัญ (ยิ่งอยู่ใกล้ ค่ายิ่งสูง)
2. จากนั้นรวมค่า IDW ที่ได้ และทำการ NORMALIZATION ให้อยู่ในช่วงระหว่าง 0 - 1

FREQUENCY SCORE : จำนวนครั้งที่เหตุการณ์ประเภทเดิมเกิดขึ้นซ้ำ ในเขตเดียวกัน ภายในช่วงระยะเวลา 1 เดือน

1. ตรวจสอบจำนวนครั้งทั้งหมดในประเภทเดียวกัน ที่เกิดขึ้นซ้ำในแต่ละเขต ภายในระยะเวลา 1 เดือน
2. บันทึกจำนวนครั้งที่พบลงในคอลัมน์ FREQ_COUNT
3. NORMALIZE ค่าที่ได้ให้อยู่ในช่วงระหว่าง 0 - 1





Step 5: Pipeline



สิ่งที่ทำ

ใช้ AIRFLOW จัดการข้อมูลให้เป็น END-TO-END DATA PIPELINE เพื่อทำงานแบบ DAILY โดยเริ่มตั้งแต่การดึงข้อมูล → ประมวลผล → วิเคราะห์ด้วย MODEL → ส่งออกไปยัง DASHBOARD

ขั้นตอนของ PIPELINE

START DAILY

กำหนดให้ PIPELINE เริ่มทำงานอัตโนมัติทุกวัน

SIMULATE NEW DATA

SELECT 1,000 RECORDS FROM TRAFFY DATASET.

LOAD EXTERNAL DATA

SCRAPED SCHOOLS / HOSPITALS
(1,000+ RECORDS)

PREPROCESS

CLEAN DATA + IMPUTE MISSING VALUES

AI/ML

CLASSIFY PROBLEM TYPE

CALCULATE

PRIORITY SCORE

SAVE OUTPUT → DASHBOARD



step 6: visualization

สิ่งที่เราได้พัฒนาใน DASHBOARD

MAIN VIEW

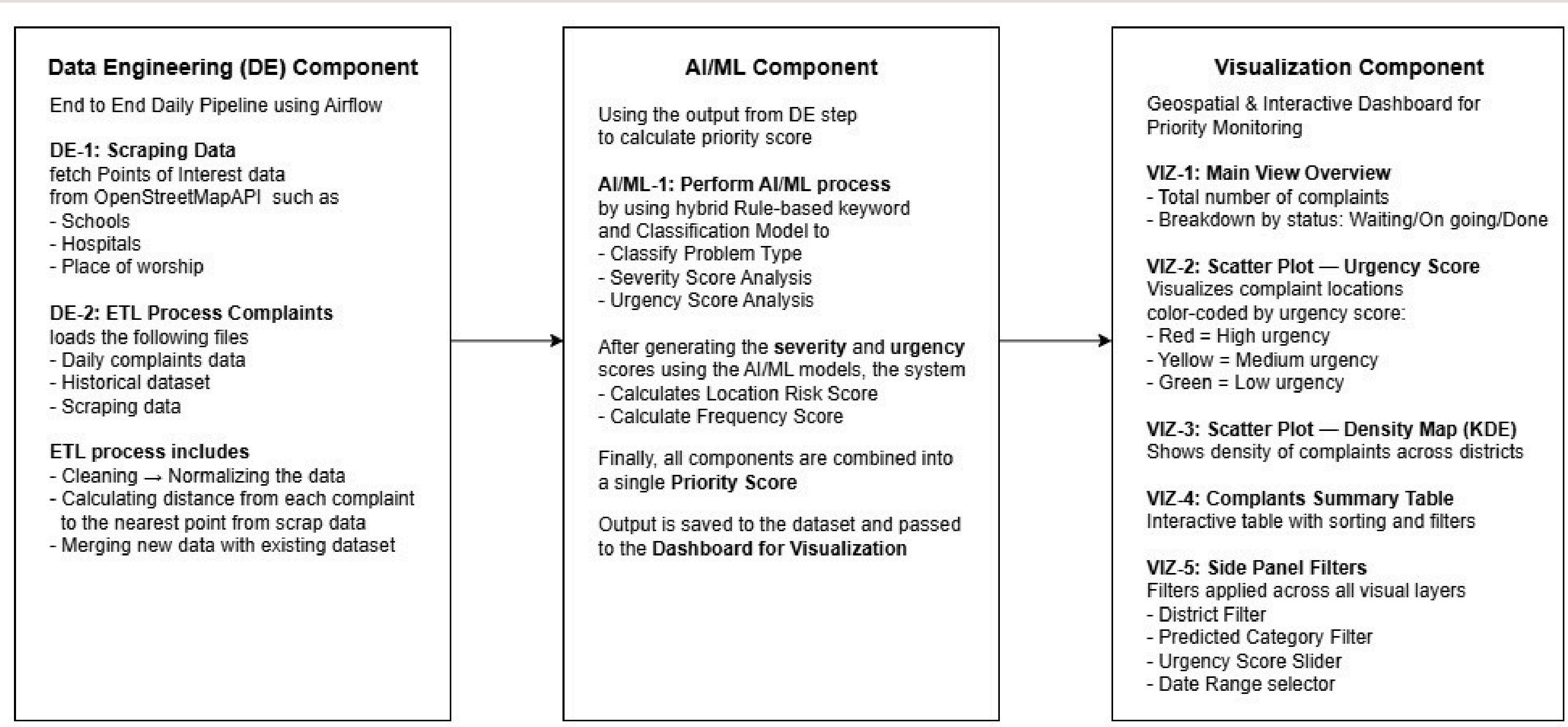
- แสดงจำนวนเหตุการณ์ทั้งหมด และจำนวนตามสถานะ
 - (รอดำเนินการ / กำลังดำเนินการ / เสร็จสิ้น)
- SCATTER PLOT (PRIORITY SCORE): จุดแต่ละจุดแทนหนึ่งคำร้อง และໄล์สีตามความเร่งด่วน
 - สีแดง = ค่า PRIORITY สูง
 - สีเขียว = ค่า PRIORITY ต่ำ
- ตารางสรุปรายการเหตุการณ์ (INCIDENT TABLE)
- SCATTER PLOT (DENSITY):
 - แสดงความหนาแน่นของเหตุการณ์ในพื้นที่ต่าง ๆ

SIDE PANEL FILTERS

- DISTRICT FILTER: เลือกเขต
- PREDICTED CATEGORY: เลือกประเภทของปัญหาที่สนใจ
- PRIORITY SCORE SLIDER: เลือกรอง INCIDENTS ตามค่า PRIORITY SCORE ที่ต้องการ
- DATE RANGE: เลือกช่วงวันที่ต้องการแสดงผล



required components





Members and roles

6733297421	ไอย์รดา จิตรารีย์รักษ์	CLEANING DATASET AND SET UP PIPELINE
6733189321	ภัค โภกิลезнันท์	CLEANING DATASET AND SET UP PIPELINE
6733269921	สาริสา สมตน	CLASSIFY TYPES AND PRIORITY SCORE
6733286521	อนันญา คณารักษ์สันติ	CLASSIFY TYPES AND PRIORITY SCORE
6733069721	ณัฐปศุลักษ์ สมบูรณ์	WEB SCRAPING AND VISUALIZATION
6733067421	ณัฐนนท์ ณ ระนอง	WEB SCRAPING AND VISUALIZATION

