Version 3 - Stacking Model from V2 Features - Feature Engineering Process

เอกสารอธิบายกระบวนการสร้าง feature แบบอ่านเข้าใจง่าย พร้อม code/artifact ที่เกี่ยวข้อง

Process Overview

- นำ feature base จาก V2 มาใช้ต่อเพื่อทดสอบผลของ model architecture
- สร้าง stacking ensemble จาก XGBoost, LightGBM, CatBoost และ meta learner
- ใช้ train/test split จาก V2 เพื่อให้เทียบผลจาก model ได้ยุติธรรม
- ประเมิน threshold/cost/recall trade-off

Code / Artifact Reference

Code / Artifact	Path
model_training_v3_stacking.py	docs\version 3\stacking_model_v3\scripts\model_training_v3_stacking.py
model_evaluation_v3.py	docs\version 3\stacking_model_v3\scripts\model_evaluation_v3.py

Input / Output

Stage	Artifact
Input clean data	docs\version 2\data\features\df_engineered.csv
Engineered dataset	docs\version 3\stacking_model_v3\data\df_featured.csv
Used feature list	docs\version 3\stacking_model_v3\data\v3_used_features.csv

Feature Engineering Logic

- เริ่มจาก clean dataset ของ version นั้น แล้วสร้าง numeric/encoded/interaction features
- ใช้ train/test split เดียวใน version เพื่อเปรียบเทียบ model อย่างยุติธรรม
- drop leakage/identifier fields ก่อน train
- ประเมินด้วย Accuracy, Recall, Precision, F1, AUC และ Cost Matrix

Performance Result

Accuracy	Recall	Precision	F1	AUC	Cost	Rating	Used Features
61.60%	63.76%	44.84%	52.65%	71.90%	19,500	B	38

Decision

ไม่เลือกเป็น final เพราะซับซ้อนกว่า แต่ performance ไม่ชนะ V2

Important Notes

- เอกสารนี้ไม่ทับ model artifact เดิม แต่รีเจน documentation ให้ตรงกับสถานะล่าสุดของโปรเจ็กต์
- ถ้า metrics/model เปลี่ยน ให้รัน scripts/generate_current_project_docs_v1_to_v4.py ใหม่เพื่อ sync เอกสาร
- สำหรับ production จริงควร validate ด้วยข้อมูลจริงแบบ time-based holdout ก่อน deploy