การศึกษาเชิงเปรียบเทียบของแบบจำลองทรานส์ฟอร์เมอร์ภาษาไทยสำหรับพยากรณ์ความไวรัลของวิดีโอ YouTube

A COMPARATIVE STUDY OF THAI TRANSFORMER LANGUAGE MODELS FOR YOUTUBE VIDEO VIRALITY PREDICTION

นายตรัยภูรินท์ สืบสุวรรณสาร

นายธีรนนท์ ไชยลังกา

นายภาณุเดช ภู่โทสนธิ์

ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

ปริญญาวิทยาศาสตรบัณฑิต (สถิติประยุกต์และการวิเคราะห์ข้อมูล)

ภาควิชาสถิติ คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2568

A COMPARATIVE STUDY OF THAI TRANSFORMER LANGUAGE MODELS FOR YOUTUBE VIDEO VIRALITY PREDICTION

MR. TRAIPOOHRIN SUEBUSWANSARN

MR. TEERANON CHAILANGKA

MR. PHANUDECH PHUTHOSON

A SPECIAL PROBLEM SUBMITTED IN PARTIAL

FULFILLMENT OF THE REQUIREMENTS FOR

THE DEGREE OF BACHELOR OF SCIENCE

(APPLIED STATISTICS AND DATA ANALYTICS)

DEPARTMENT OF STATISTICS, SCHOOL OF SCIENCE

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ACADEMIC YEAR 2025

หัวข้อปัญหาพิเศษ	การศึกษาเชิงเปรียบเทียบของแบบจำลองทรานส์ฟอร์เมอร์ภาษาไทยสำหรับพยากรณ์ความไวรัลของวิดีโอ YouTube A Comparative Study of Thai Transformer Language Models for YouTube Video Virality Prediction
ชื่อนักศึกษา	นายตรัยภูรินท์ สืบสุวรรณสาร รหัสนักศึกษา 65050330 นายธีรนนท์ ไชยลังกา รหัสนักศึกษา 65050417 นายภาณุเดช ภู่โทสนธิ์ รหัสนักศึกษา 65050685
ปริญญา	วิทยาศาสตรบัณฑิต (สถิติประยุกต์และการวิเคราะห์ข้อมูล)
ภาควิชา	สถิติ
คณะ	วิทยาศาสตร์
มหาวิทยาลัย	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง (สจล.)
ปีการศึกษา	2568
อาจารย์ที่ปรึกษา	ผศ. ดร. พรพิมล ชัยวุฒิศักดิ์

คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง (สจล.) อนุมัติให้ ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต (สถิติประยุกต์และการวิเคราะห์ข้อมูล) ประจำปีการศึกษา 2568

คณะกรรมการสอบ	**ลายมือชื่อ**
ดร. อัครชัย อินทนิล *ประธานกรรมการ*	. . . . .
อ. สุจิตรา สุคนธมัต *กรรมการ*	. . . . .
ผศ. ดร. พรพิมล ชัยวุฒิศักดิ์ *กรรมการและอาจารย์ที่ปรึกษา*	. . . . .

ลิขสิทธิ์ของคณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

บทคัดย่อ

หัวข้อปัญหาพิเศษ	การศึกษาเชิงเปรียบเทียบของแบบจำลองทรานส์ฟอร์เมอร์ภาษาไทยสำหรับพยากรณ์ความไวรัลของวิดีโอ YouTube
ชื่อนักศึกษา	นายตรัยภูรินท์ สืบสุวรรณสาร รหัสนักศึกษา 65050330 นายธีรนนท์ ไชยลังกา รหัสนักศึกษา 65050417 นายภาณุเดช ภู่โทสนธิ์ รหัสนักศึกษา 65050685
ปริญญา	วิทยาศาสตรบัณฑิต (สถิติประยุกต์และการวิเคราะห์ข้อมูล)
ภาควิชา	สถิติ
คณะ	วิทยาศาสตร์
มหาวิทยาลัย	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง (สจล.)
ปีการศึกษา	2568
อาจารย์ที่ปรึกษา	ผศ. ดร. พรพิมล ชัยวุฒิศักดิ์

แพลตฟอร์มวิดีโอ YouTube มีผู้ใช้งานต่อเดือนมากกว่า 2.5 พันล้านคน และเป็นช่องทางการตลาด ที่สำคัญของผู้สร้างเนื้อหาภาษาไทย การพยากรณ์ *ความไวรัล* (virality) ของวิดีโอตั้งแต่ ก่อนเผยแพร่ จึงมีคุณค่าในเชิงพาณิชย์และเชิงวิชาการ งานวิจัยนี้กำหนดปัญหาเป็นการจำแนก ประเภทไบนารี (binary classification) ของวิดีโอภาษาไทยว่าจะเป็นไวรัลหรือไม่ โดยใช้ *ดัชนีความไวรัลภายในช่อง* (Per-channel Virality Index, *VI*) ซึ่งสร้างจากผลรวม แบบถ่วงน้ำหนักของอัตราส่วนการมีส่วนร่วม (engagement rate) อัตราส่วนการกดถูกใจ และอัตราส่วน ความคิดเห็น หลังการแปลงด้วยลอการิทึม และ z-score ภายในช่องเจ้าของ ป้ายกำกับเชิงบวกถูก กำหนดเป็นเดไซล์บนสุด (top-decile) ของแต่ละช่องเพื่อหลีกเลี่ยงไม่ให้ขนาดช่องเป็นปัจจัยรบกวน

ชุดข้อมูลประกอบด้วยวิดีโอจำนวน 23,431 รายการจาก 82 ช่องที่จัดเก็บผ่าน YouTube Data API v3 มีสัดส่วนภาคชั้นบวก 10.16% การแบ่งชุดข้อมูลใช้ *แบบไวต่อเวลาผสมจัดกลุ่ม ตามช่อง* (channel-grouped, time-aware split) โดยกำหนดให้ชุดทดสอบเป็นวิดีโอที่ตีพิมพ์ ในช่วง 15% หลังสุดตามเวลา และทำการสุ่มลดข้อมูล (undersampling) ในอัตราส่วน 3:1 *เฉพาะชุดฝึกฝนเท่านั้น* เพื่อ

รักษาการกระจายตัวตามธรรมชาติของชุดตรวจสอบและ ชุดทดสอบ

งานวิจัยเปรียบเทียบแบบจำลองทรานส์ฟอร์เมอร์ภาษาไทยขนาดและสถาปัตยกรรมต่างกัน 3 แบบ ได้แก่ **WangchanBERTa** (105M, RoBERTa SentencePiece) **PhayaThaiBERT** (278M, RoBERTa) และ **XLM-RoBERTa-large** (560M, multilingual) ทุกแบบจำลอง ปรับจูนแบบเต็ม (full fine-tuning) บน GPU NVIDIA T4/A10G ผ่าน Hugging Face Jobs แบบจำลองพื้นฐาน (Logistic Regression, LightGBM, XGBoost) ใช้ชุดคุณลักษณะเชิงโครงสร้าง 27 ตัว ผสมกับ TF-IDF อักขระ-N-gram และฟีเจอร์เชิงอารมณ์ (sentiment features) จาก Wisesight-Sentiment นอกจากนี้ยังพัฒนาแบบจำลอง *ไฮบริด* (MLP + LightGBM head) ซึ่งใช้เวกเตอร์ฝังข้อความรวมกับฟีเจอร์เชิงโครงสร้าง ปิดท้ายด้วย *การรวมแบบจำลอง* (stacking ensemble) ที่ใช้ Logistic Regression เป็นตัวเรียนระดับเมตา (meta-learner) บนความน่าจะเป็นของฐาน 15 แบบจำลอง

จากการประเมินบนชุดทดสอบเดียวกัน แบบจำลองที่ดีที่สุดคือ **การรวมแบบจำลอง ที่ปรับเทียบด้วย Platt** ซึ่งให้ค่า ROC-AUC = **0.6914** โดยมีช่วงความเชื่อมั่น บูตสแตรป 95% **[0.6625, 0.7174]**, ECE = 0.016 ซึ่งสูงกว่าแบบจำลองที่ดีที่สุด เดี่ยว (LightGBM-Structured+TF-IDF, 0.6728) อย่างมีนัยสำคัญ ($p = 6.9 \times 10^{-53}$ จาก McNemar's test) ในระดับแบบจำลองทรานส์ฟอร์เมอร์เดี่ยว PhayaThaiBERT (0.6451) ทำงานได้ดีที่สุด ตามด้วย WangchanBERTa (0.6402) และ XLM-RoBERTa-large (0.5450) การทดสอบ Cochran's Q ระหว่างทรานส์ฟอร์เมอร์ทั้งสามให้ $Q = 612.69$, $p = 9.0 \times 10^{-134}$ ($df = 2$) แสดงว่ามีความแตกต่างของรูปแบบความผิดพลาดอย่างมีนัยสำคัญ การแบ่งย่อยรายช่องชี้ว่าแบบจำลองทำงานได้ดีกว่าในช่องขนาดใหญ่ (ROC-AUC = 0.7459 สำหรับ $\geq$1M ผู้ติดตาม) เทียบกับช่องขนาดกลาง (0.6273)

งานวิจัยรายงานผลลัพธ์ *เชิงลบที่ตรงไปตรงมา* ว่าการเพิ่มข้อความหลายฟิลด์ (title + description + channel) ทำให้ประสิทธิภาพ PhayaThaiBERT *ลดลง* จาก 0.6451 เหลือ 0.5455 ซึ่งสะท้อนว่าคำอธิบายวิดีโอภาษาไทยส่วนใหญ่เป็นข้อความ สำเร็จรูป (boilerplate) การวิเคราะห์ความสามารถอธิบาย (explainability) ผ่าน SHAP, LIME และ Attention Rollout แสดงว่าตัวสำคัญที่สุดในการพยากรณ์คือฟีเจอร์เชิงโครงสร้าง (ความยาวชื่อ ความหนาแน่นของอีโมจิในชื่อ ผู้ติดตามช่อง) ในขณะที่สัญญาณจากแบบจำลอง ภาษาเข้ามาเสริมผ่านการรวมแบบจำลอง

คำสำคัญ: การพยากรณ์ความไวรัลของวิดีโอ ทรานส์ฟอร์เมอร์ภาษาไทย WangchanBERTa PhayaThaiBERT XLM-RoBERTa การรวมแบบจำลอง McNemar's test Cochran's Q การปรับเทียบความน่าจะเป็น SHAP LIME

Abstract

Title	A Comparative Study of Thai Transformer Language Models for YouTube Video Virality Prediction
Students	Mr. Traipoohrin Suebuswansarn Student ID 65050330 Mr. Teeranon Chailangka Student ID 65050417 Mr. Phanudech Phuthoson Student ID 65050685
Degree	Bachelor of Science (Applied Statistics and Data Analytics)
Department	Statistics
School	Science
University	King Mongkut's Institute of Technology Ladkrabang (KMITL)
Academic Year	2025
Advisor	Asst. Prof. Dr. Pornpimol Chaiwuttisak, Ph.D.

YouTube hosts more than 2.5 billion monthly active users and is the dominant distribution channel for Thai content creators. Predicting whether a newly posted video will go *viral* is therefore commercially and academically valuable. This study formulates the task as binary classification of Thai videos using a *per-channel virality index* (*VI*) that aggregates the engagement rate, like ratio, and comment ratio after a log-transform and a within-channel z-score, with the positive class taken as the top decile per channel. The per-channel construction explicitly removes channel size as a confounder.

The dataset comprises 23,431 videos from 82 channels collected through the YouTube Data API v3, with a 10.16% positive class rate. The split policy is *channel-grouped and time-aware*: the test set is the latest 15% of videos by published_at, with no channel shared between train and validation. Class imbalance is handled *only on the training fold* (3:1 undersampling), so validation and test retain

the natural distribution.

Three Thai transformer encoders of differing scale and architecture are compared end-to-end: **WangchanBERTa** (105M, RoBERTa SentencePiece), **PhayaThaiBERT** (278M, RoBERTa), and **XLM-RoBERTa-large** (560M, multilingual). All three are full-fine-tuned on cloud GPUs (NVIDIA T4/A10G via Hugging Face Jobs). Classical baselines (Logistic Regression, LightGBM, XGBoost) are trained over 27 structured features fused with character-N-gram TF-IDF and Wisesight-Sentiment scores. A hybrid model (MLP + LightGBM head) operates on title embeddings concatenated with structured features. Finally, a *Platt-calibrated stacking ensemble* fits a logistic-regression meta-learner over the validation-fold probabilities of fifteen base models.

On the held-out time-aware test set, the calibrated stacking ensemble attains a test **ROC-AUC of 0.6914** with a 1,000-sample bootstrap 95% confidence interval of **[0.6625, 0.7174]** and an Expected Calibration Error of 0.016. This is significantly higher than the strongest single model, LightGBM-Structured+TF-IDF (0.6728; McNemar $p = 6.9 \times 10^{-53}$). Among single transformer encoders, PhayaThaiBERT performs best (0.6451), followed by WangchanBERTa (0.6402) and XLM-RoBERTa-large (0.5450). Cochran's Q across the three encoders yields $Q = 612.69$, $p = 9.0 \times 10^{-134}$ ($df = 2$), confirming that the encoders' error patterns differ significantly. Per-bucket evaluation shows the ensemble generalises better on large channels ($\geq$1M subscribers, ROC-AUC = 0.7459) than on mid-size channels (100k–1M, 0.6273).

We report an honest *negative finding*: feeding PhayaThaiBERT a multi-field text (title + description + channel) *degraded* ROC-AUC from 0.6451 to 0.5455, reflecting the boilerplate nature of Thai YouTube descriptions. Explainability analyses (SHAP, LIME, attention rollout) show that the dominant predictive signal lives in structured features (title length, emoji density, channel subscriber count) and that language signals contribute to the ensemble through diversity rather than through any single best

encoder.

Keywords: YouTube virality prediction, Thai transformer language models, WangchanBERTa, PhayaThaiBERT, XLM-RoBERTa, stacking ensemble, McNemar's test, Cochran's Q, probability calibration, SHAP, LIME.

กิตติกรรมประกาศ

ปัญหาพิเศษเล่มนี้สำเร็จลุล่วงได้ด้วยดี เนื่องจากได้รับความกรุณาและความเอื้อเฟื้อ เป็นอย่างยิ่งจาก ผศ. ดร. พรพิมล ชัยวุฒิศักดิ์ อาจารย์ที่ปรึกษา ที่ได้สละเวลาอันมีค่าในการให้ คำปรึกษา ชี้แนะแนวทาง ตรวจสอบ และให้ข้อเสนอแนะที่เป็นประโยชน์อย่างยิ่ง ตลอดระยะเวลาการดำเนินงาน ตั้งแต่การออกแบบกรอบงานวิจัย การกำหนดดัชนีความ ไวรัล การออกแบบการทดลอง การวิเคราะห์ผล จนถึงการเขียนรูปเล่ม คณะผู้จัดทำ ขอกราบขอบพระคุณเป็นอย่างสูงมา ณ โอกาสนี้

ขอขอบพระคุณ ดร. อัครชัย อินทนิล ประธานกรรมการสอบ และ อ. สุจิตรา สุคนธมัต กรรมการสอบ ที่ได้กรุณาให้ข้อเสนอแนะอันมีคุณค่าต่อการปรับปรุง งานวิจัยให้สมบูรณ์ยิ่งขึ้น

ขอขอบพระคุณภาควิชาสถิติประยุกต์และการวิเคราะห์ข้อมูล คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ได้ให้การสนับสนุน ทรัพยากรการคำนวณ ความรู้พื้นฐานทางสถิติ และสภาพแวดล้อมทางวิชาการที่ เอื้อต่อการศึกษาวิจัย

ขอขอบคุณทีมงานที่อยู่เบื้องหลังเครื่องมือโอเพนซอร์สที่ใช้ในงานนี้ ได้แก่ AIResearch.in.th (WangchanBERTa), CLICKNEXT (PhayaThaiBERT), Meta AI (XLM-RoBERTa), Hugging Face (Transformers, Datasets, Hub, Jobs), PyThaiNLP, scikit-learn, LightGBM, XGBoost, MLflow, SHAP และ LIME ที่ทำให้ผู้วิจัยสามารถเข้าถึงเทคโนโลยีระดับโลกได้

ท้ายที่สุด คณะผู้จัดทำขอขอบพระคุณบิดา มารดา ครอบครัว เพื่อน และรุ่นพี่ ในภาควิชา ที่เป็นกำลังใจสำคัญตลอดระยะเวลาการดำเนินงาน

นายตรัยภูรินท์ สืบสุวรรณสาร

นายธีรนนท์ ไชยลังกา

นายภาณุเดช ภูโทสนธิ์

สารบัญ

บทคัดย่อภาษาไทย ..... **ก**

Abstract (English) ..... **ง**

กิตติกรรมประกาศ ..... **ช**

คำย่อ/สัญลักษณ์ ..... **ถ**

1 บทนำ **1**

1.1 ความเป็นมาและความสำคัญของปัญหา 1

1.2 วัตถุประสงค์ของการวิจัย 2

1.3 ขอบเขตของการวิจัย 3

1.3.1 ขอบเขตด้านปัญหา 3

1.3.2 ขอบเขตด้านแบบจำลอง 3

1.3.3 ขอบเขตด้านข้อมูล 4

1.4 ประโยชน์ที่คาดว่าจะได้รับ 4

1.5 นิยามศัพท์ที่ใช้ในงานวิจัย 5

1.6 สมมติฐานของการวิจัย 6

2 การทบทวนวรรณกรรม **7**

2.1 การพยากรณ์ความนิยมของเนื้อหาบนสื่อออนไลน์ 7

2.1.1 นิยามและการวัดความไวรัล 7

2.1.2 งานพยากรณ์ความนิยมบน YouTube และแพลตฟอร์มอื่น 7

2.2 คุณลักษณะของวิดีโอที่ส่งผลต่อความไวรัล 8

2.2.1 ฟีเจอร์เชิงข้อความ (Textual Features) 8

2.2.2 ฟีเจอร์เชิงอารมณ์ (Sentiment Features) 9
2.2.3 ฟีเจอร์เชิงโครงสร้าง (Channel and Temporal Features) 9
2.3 แบบจำลองทรานส์ฟอร์เมอร์ 9
2.3.1 สถาปัตยกรรม Transformer 9
2.3.2 BERT และ RoBERTa 10
2.3.3 แบบจำลองทรานส์ฟอร์เมอร์ภาษาไทย 10
2.3.4 การปรับจูน (Fine-tuning) แบบเต็มเทียบกับ Parameter-Efficient 11
2.4 แบบจำลองพื้นฐานเชิงคลาสสิกและการเรียนรู้เชิงผสม 11
2.4.1 Logistic Regression, LightGBM, XGBoost 11
2.4.2 TF-IDF กับการจำแนกข้อความ 12
2.4.3 แบบจำลองไฮบริด (Hybrid Models) 12
2.4.4 การรวมแบบจำลองแบบเรียงซ้อน (Stacking Ensemble) 12
2.5 การจัดการความไม่สมดุล 12
2.5.1 การปรับสมดุลของข้อมูล (Resampling) 13
2.5.2 การถ่วงน้ำหนัก (Class Weighting) และ Focal Loss 13
2.5.3 การปรับเปลี่ยนค่าเกณฑ์ตัดสิน (Decision Threshold) 13
2.6 การปรับเทียบความน่าจะเป็น 13
2.6.1 Platt Scaling 13
2.6.2 Isotonic Regression 14
2.6.3 การวัดคุณภาพการปรับเทียบ 14
2.7 การทดสอบความแตกต่างทางสถิติของแบบจำลอง 14
2.7.1 McNemar’s Test 14
2.7.2 Cochran’s Q Test 15
2.7.3 Bootstrap Confidence Intervals 15
2.8 ความสามารถในการอธิบายของแบบจำลอง 15
2.8.1 SHAP 15
2.8.2 LIME 15

2.8.3 Attention Rollout 16
2.9 งานวิจัยที่เกี่ยวข้อง..... 16
3 ระเบียบวิธีวิจัย **17**
3.1 ภาพรวมของขั้นตอนและสถาปัตยกรรมของระบบ..... 17
3.2 การเก็บรวบรวมข้อมูล..... 18
3.2.1 จริยธรรมการวิจัยและความสอดคล้องกับนโยบายแพลตฟอร์ม..... 18
3.2.2 แหล่งข้อมูลและขอบเขต 18
3.2.3 ตัวแปรที่จัดเก็บ..... 19
3.3 การทำความสะอาดและการสร้างป้ายกำกับ..... 20
3.3.1 การทำความสะอาดข้อมูล 20
3.3.2 การออกแบบดัชนีความไวรัล..... 20
3.4 การแบ่งชุดข้อมูลแบบ Channel-Grouped + Time-Aware 21
3.4.1 Channel-Grouped (Train/Val)..... 22
3.4.2 Time-Aware (Test)..... 22
3.4.3 การจัดการความไม่สมดุลของชั้นเฉพาะชุดฝึก 22
3.5 การออกแบบและการสกัดฟีเจอร์ 24
3.5.1 ฟีเจอร์เชิงโครงสร้าง (Structural Features, 27 ตัว) 24
3.5.2 ฟีเจอร์เชิงอารมณ์ (Sentiment Features, 6 ตัว)..... 24
3.5.3 ฟีเจอร์ TF-IDF (Character N-gram) 25
3.5.4 ฟีเจอร์เวกเตอร์ฝังของชื่อ (Title Embeddings, 768 มิติ)..... 25
3.6 แบบจำลองพื้นฐาน 25
3.7 การปรับจูนทรานส์ฟอร์เมอร์ภาษาไทย..... 26
3.7.1 แบบจำลองที่เปรียบเทียบ 26
3.7.2 Wrapper เดียวสำหรับทั้งสามแบบจำลอง..... 26
3.7.3 ฟังก์ชันความสูญเสีย 27
3.7.4 Mixed Precision และ Gradient Checkpointing..... 27

3.7.5 การฝึกบนคลาวด์ (Hugging Face Jobs) 27
3.7.6 แบบจำลองไฮบริด (MLP + LightGBM) 28
3.8 การรวมแบบจำลองและการปรับเทียบ 29
3.8.1 สถาปัตยกรรม Stacking 29
3.8.2 การปรับเทียบความน่าจะเป็น 30
3.9 ระเบียบการประเมินผล 31
3.9.1 ตัวชี้วัด 31
3.9.2 Bootstrap Confidence Intervals 32
3.9.3 การเลือกค่าเกณฑ์ (Threshold) 32
3.9.4 การทดสอบทางสถิติ 32
3.9.5 การวิเคราะห์ความผิดพลาด 32
3.9.6 การวิเคราะห์เชิงกลุ่มย่อย (Sub-population) 32
3.10 การวิเคราะห์ความสามารถในการอธิบาย 33
3.10.1 SHAP บน LightGBM-best 33
3.10.2 LIME บนทรานส์ฟอร์เมอร์ 33
3.10.3 Attention Rollout 33
3.11 การทดลองเสริมและการ Ablation 33
3.12 สมรรถนะที่ทำซ้ำได้และการบันทึก 34
4 ผลการวิจัย 35
4.1 สรุปประสิทธิภาพของแบบจำลองทั้งหมด 35
4.1.1 การกระจายของ ROC-AUC รายช่อง 37
4.1.2 การตรวจสอบความเสถียรแบบ 5-fold over channels 39
4.1.3 Confusion Matrices ของแบบจำลองหลัก 40
4.2 การเปรียบเทียบทรานส์ฟอร์เมอร์ภาษาไทยสามตัว 41
4.2.1 การเปรียบเทียบกับงานก่อนหน้าในต่างประเทศ 41
4.3 แบบจำลองพื้นฐาน 43

4.3.1 ผลตามชุดฟีเจอร์ 43
4.3.2 การสังเกตเชิงโครงสร้าง 43
4.4 Stacking Ensemble 44
4.4.1 Stacking ทั้งสามเวอร์ชัน 44
4.4.2 น้ำหนักของ meta-LR 44
4.5 การปรับเทียบความน่าจะเป็น 45
4.5.1 ECE ก่อนและหลังการปรับเทียบ 45
4.5.2 Reliability Diagram 46
4.5.3 ROC และ Precision-Recall 46
4.6 การทดสอบทางสถิติ 47
4.6.1 McNemar’s Test: Stacking vs Best Single Model 47
4.6.2 Cochran’s Q: Across the Three Encoders 48
4.7 การวิเคราะห์เชิงกลุ่มย่อย 48
4.7.1 ตามขนาดช่อง 48
4.7.2 ตามหมวดหมู่ 49
4.8 การทดลอง Ablation 49
4.8.1 Ablation 0: Sensitivity ของดัชนี Virality Index 49
4.8.2 Ablation 1: Sentiment Feature 50
4.8.3 Ablation 2: Multi-field PhayaThaiBERT 51
4.8.4 Ablation 3: HPO PhayaThaiBERT 52
4.8.5 Ablation 4: Drop-One Stacking 52
4.9 ความสามารถในการอธิบาย 54
4.9.1 SHAP บน LightGBM-best 54
4.9.2 LIME บน PhayaThaiBERT 56
4.9.3 Attention Rollout บน PhayaThaiBERT 57
4.9.4 การวิเคราะห์ Top False Positives/Negatives 57
4.9.5 สรุปการวิเคราะห์ความสามารถในการอธิบาย 58

4.10 สรุปการตอบสมมติฐาน 58

5 สรุปผลและข้อเสนอแนะ 59

5.1 สรุปผลการวิจัย 59

5.2 การสนับสนุนทางวิชาการ 60

5.3 ข้อจำกัดของการวิจัย 61

5.3.1 ข้อจำกัดด้านข้อมูล 61

5.3.2 ข้อจำกัดด้านแบบจำลอง 61

5.3.3 ข้อจำกัดด้านการประเมินผล 62

5.4 ข้อเสนอแนะสำหรับการวิจัยต่อยอด 62

5.4.1 การขยายข้อมูลและฟีเจอร์ 62

5.4.2 การขยายแบบจำลอง 62

5.4.3 การขยายเชิงประยุกต์ 63

5.4.4 การขยายเชิงระเบียบวิธี 63

5.5 ข้อสรุปท้ายสุด 63

เอกสารอ้างอิง 65

A ค่าตั้งค่า (Configs) ของระบบ 70

A.1 configs/data.yaml 70

A.2 configs/features.yaml 70

A.3 configs/train.yaml 71

A.4 configs/eval.yaml 73

B ตารางผลการทดลองฉบับเต็ม 74

B.1 Pairwise McNemar’s Test สำหรับ 22 แบบจำลอง 74

B.2 Drop-One Stacking Ablation (เต็ม) 74

B.3 สรุปการปรับเทียบทั้งหมด 75

B.4 Per-Category Breakdown 76

C ความสามารถในการทำซ้ำ 77

C.1 สภาพแวดล้อมการรัน 77
C.2 คำสั่งที่ใช้ในการทำซ้ำผล 77
C.3 Hash ของข้อมูลและ git SHA..... 79
C.4 เหตุการณ์และข้อจำกัดเชิงเทคนิค 79
C.4.1 Apple M1 MPS และ NaN logits 79
C.4.2 Kaggle Free GPU Pool 79
C.4.3 Train_hybrid Silent GBM Skip on MPS 79
C.5 การ Audit ทางสถิติ 79
D Datasheet สำหรับชุดข้อมูล..... 81
D.1 แรงจูงใจ (Motivation) 81
D.2 องค์ประกอบ (Composition) 81
D.3 การเก็บรวบรวม (Collection)..... 83
D.4 การ Preprocessing / Labeling / Cleaning..... 83
D.5 การใช้งาน (Uses)..... 84
D.6 การจัดเก็บและการแจกจ่าย (Distribution)..... 84
D.7 การบำรุงรักษา (Maintenance)..... 84

สารบัญรูป

ภาพที่ 3.1: ภาพรวมขั้นตอนของระบบ จากการเก็บข้อมูลถึงการรายงานผล..... 17
ภาพที่ 3.2: สถาปัตยกรรมของ stacking ensemble ที่ปรับเทียบด้วย Platt scaling แบบจำลองฐาน 15 ตัวให้ความน่าจะเป็นบนชุด validation ซึ่ง meta-LR เรียนรู้ น้ำหนักรวม จากนั้น Platt calibrator ปรับ output สุดท้ายให้สะท้อนความน่าจะเป็น จริง 29

ภาพที่ 4.1: Leaderboard ของแบบจำลอง 12 อันดับแรกบนชุดทดสอบ พร้อมช่วงความเชื่อมั่น 95% สีแดง = ensemble, ม่วง = hybrid, น้ำเงิน = transformer, เทา = baseline (แหล่ง: reports/figures/leaderboard_with_ci.svg) 37

ภาพที่ 4.2: ซ้าย — empirical CDF ของ ROC-AUC ภายในแต่ละช่อง $(n = 48)$ เส้นแดง แสดง median, เส้นเขียวแสดง global ROC-AUC = 0.6914 ขวา — กระจายของ ROC-AUC ตามอันดับ (แกน x) สีของแต่ละจุดสะท้อน $\log_{10}$ ของจำนวนผู้ติดตามช่อง (ดูเฉดสี viridis) 38

ภาพที่ 4.3: Reliability diagram ของ stacking ensemble ก่อนและหลังการปรับเทียบ ด้วย Platt scaling จุดเชิงเส้นใกล้เส้นทแยง = ปรับเทียบดี ... 46

ภาพที่ 4.4: ROC curve (ซ้าย) และ Precision-Recall curve (ขวา) ของ 6 แบบจำลอง อันดับต้นบนชุดทดสอบ 47

ภาพที่ 4.5: ความเสถียรของ stacking ensemble ตามหมวดหมู่ (ซ้าย) และตามขนาดช่อง (ขวา) 49

ภาพที่ 4.6: SHAP summary plot ของ LightGBM-Structured+TF-IDF จุดสีแดง = ค่า ฟีเจอร์สูง สีน้ำเงิน = ค่าฟีเจอร์ต่ำ ตำแหน่งแกน x = แรงโน้มน้าวต่อพยากรณ์ ชั้นบวก 55

สารบัญตาราง

ตารางที่ 3.1: การกระจายของช่องเป้าหมายตามขนาด 19

ตารางที่ 3.2: ตัวแปรหลักที่จัดเก็บจาก YouTube Data API v3 19

ตารางที่ 3.3: ขนาด และ อัตราส่วน ชั้น บวก ใน แต่ละ ชุด ข้อมูล ก่อน และ หลัง undersampling 23

ตารางที่ 3.4: ตารางการรัน baseline (algorithm $\times$ feature set) 25

ตารางที่ 3.5: แบบจำลองทรานส์ฟอร์เมอร์ที่เปรียบเทียบ 26

ตารางที่ 3.6: เวลาฝึก (wall-time) และต้นทุนของ encoder fine-tuning บน Hugging Face Jobs 28

ตารางที่ 4.1: ROC-AUC, F1, MCC ของแบบจำลองทั้งหมดบนชุดทดสอบ พร้อม CI 95% จากบูตสแตรป $n = 1000$ 36

ตารางที่ 4.2: สรุปการกระจายของ ROC-AUC รายช่องสำหรับ stacking ensemble 38

ตารางที่ 4.3: 5-fold over channels บนชุดทดสอบ ($n = 3{,}510$, แบ่งเป็น 5 group folds ขนาดละ 702 ตัวอย่าง โดยช่องไม่ปนข้ามกลุ่ม) 39

ตารางที่ 4.4: Channel-bootstrap (500 iterations) บน stacking_lr_calibrated . 40

ตารางที่ 4.5: Confusion matrix ของ 3 แบบจำลองหลักบนชุดทดสอบ ($n = 3{,}510, n_+ = 308$) 41

ตารางที่ 4.6: การเปรียบเทียบทรานส์ฟอร์เมอร์ภาษาไทยสามตัวบนชุดทดสอบเดียวกัน 41

ตารางที่ 4.7: การเปรียบเทียบเชิงตำแหน่งกับงานก่อนหน้าในการพยากรณ์ความนิยม/ความไวรัล ของวิดีโอออนไลน์ (ตัวชี้วัดและขอบเขตข้อมูลแตกต่างกัน — เปรียบเทียบเป็น เชิงตำแหน่งเท่านั้น) 42

ตารางที่ 4.8: แบบจำลองพื้นฐาน: ROC-AUC ตามอัลกอริทึมและชุดฟีเจอร์ 43

ตารางที่ 4.9: Stacking ensemble สามเวอร์ชัน 44

ตารางที่ 4.10: ค่าสัมประสิทธิ์ของ meta-LR ใน stacking_lr 45

ตารางที่ 4.11: ECE ของ stacking_lr ก่อนและหลังการปรับเทียบ..... 46

ตารางที่ 4.12: McNemar’s pairwise test สำหรับคู่หลัก..... 47

ตารางที่ 4.13: Cochran's Q test ระหว่างทรานส์ฟอร์เมอร์สามตัวบนชุดทดสอบเดียวกัน 48

ตารางที่ 4.14: ประสิทธิภาพของ stacking_lr_calibrated ตามขนาดช่อง 48

ตารางที่ 4.15: ประสิทธิภาพของ stacking_lr_calibrated ตามหมวดหมู่ YouTube... 49

ตารางที่ 4.16: Sensitivity ของ stacking ROC-AUC ต่อพารามิเตอร์ของ *VI* (Ablation 0); แถวที่มีเครื่องหมาย † คือ default ที่ใช้ตลอดเล่ม 50

ตารางที่ 4.17: Ablation ของฟีเจอร์ Sentiment บน LightGBM 51

ตารางที่ 4.18: ผลการทดลอง multi-field input บน PhayaThaiBERT 51

ตารางที่ 4.19: การทดลองไฮเปอร์พารามิเตอร์บน PhayaThaiBERT (3 trials) 52

ตารางที่ 4.20: Drop-one ablation ของ stacking ensemble (top-6 important + bottom-3 unimportant) 53

ตารางที่ 4.21: Top-10 features by mean |*SHAP*| on LightGBM-best. หมายเหตุ: feature key ที่ขึ้นต้นด้วย emb_* หมายถึงเวกเตอร์ฝังของ ชื่อวิดีโอจาก paraphrase-multilingual-mpnet-base-v2 (768 มิติ) ที่ ถูกใช้ร่วมกับฟีเจอร์เชิงโครงสร้างใน LightGBM-best — ดู §3.5 สำหรับการกำหนดชื่อชุดฟีเจอร์..... 54

ตารางที่ 4.22: Top tokens by aggregate |LIME weight| on PhayaThaiBERT 56

ตารางที่ 4.23: การสรุปการตอบสมมติฐานของการวิจัย..... 58

ตารางที่ B.1: Pairwise McNemar's test (selected pairs)..... 74

ตารางที่ B.2: Drop-one stacking ablation (full)..... 75

ตารางที่ B.3: ECE ก่อนและหลังการปรับเทียบสำหรับสองแบบจำลองหลัก (15 bins). 76

ตารางที่ B.4: ประสิทธิภาพของ stacking_lr_calibrated ตามหมวดหมู่ YouTube (แหล่งที่มา: reports/tables/per_category_breakdown.csv)..... 76

ตารางที่ C.1: สภาพแวดล้อมการรันของผลการศึกษา..... 77

ตารางที่ C.2: Hash อ้างอิงของผลการศึกษา..... 79

คำย่อ/สัญลักษณ์

ส่วนที่ 1: รายการคำย่อ (Abbreviations)

คำย่อ/สัญลักษณ์	คำอธิบาย
API	Application Programming Interface
AUC	Area Under the Curve
BERT	Bidirectional Encoder Representations from Transformers
BOS	Beginning-Of-Sequence token
CE	Cross-Entropy (loss)
CI	Confidence Interval
CLS	Classification token (BERT-family)
CNN	Convolutional Neural Network
CV	Cross-Validation
ECE	Expected Calibration Error
EDA	Exploratory Data Analysis
FN	False Negative
FP	False Positive
FT	Fine-Tuning
GBM	Gradient Boosting Machine
GPU	Graphics Processing Unit
HF	Hugging Face
HPO	Hyperparameter Optimisation
KMITL	King Mongkut’s Institute of Technology Ladkrabang
LIME	Local Interpretable Model-Agnostic Explanations
LM	Language Model

คำย่อ/สัญลักษณ์	คำอธิบาย
LR	Logistic Regression / Learning Rate (per context)
LLM	Large Language Model
MCC	Matthews Correlation Coefficient
MLP	Multi-Layer Perceptron
MLM	Masked Language Modelling
MPS	Metal Performance Shaders (Apple Silicon GPU API)
NLP	Natural Language Processing
NSP	Next Sentence Prediction
PEFT	Parameter-Efficient Fine-Tuning
PR-AUC	Precision-Recall Area Under the Curve
QLoRA	Quantised Low-Rank Adaptation
RNN	Recurrent Neural Network
ROC-AUC	Receiver-Operating-Characteristic Area Under the Curve
SEP	Separator token (BERT-family)
SHAP	SHapley Additive exPlanations
SP	SentencePiece tokeniser
SOTA	State-Of-The-Art
TF-IDF	Term Frequency–Inverse Document Frequency
TN	True Negative
TP	True Positive
URL	Uniform Resource Locator
VI	Virality Index (per-channel)
XLM-R	XLM-RoBERTa (multilingual encoder)

ส่วนที่ 2: รายการสัญลักษณ์ทางคณิตศาสตร์ (Mathematical Symbols)

สัญลักษณ์	คำอธิบาย
VI	ดัชนีความไวรัล (Per-Channel Virality Index)
w_e, w_l, w_c	น้ำหนักของ engagement, like, comment ใน VI (0.5, 0.3, 0.2)
$z_{r,i}$	ค่า z-score ของอัตราส่วน r ของวิดีโอ i ภายในช่อง
$Q_{0.9}^{(g)}$	เดไซล์ที่ 9 ของ VI ในช่อง g
y_i	ป้ายกำกับไบนารี $\in \{0, 1\}$ ของวิดีโอ i
$\hat{p}_i$	ความน่าจะเป็นพยากรณ์ของชั้นบวกสำหรับวิดีโอ i
θ	ค่าเกณฑ์การจำแนก (จากชุดตรวจสอบ, ทำให้ $F1_{pos}$ สูงสุด)
α	ระดับนัยสำคัญ ($= 0.05$)
γ	พารามิเตอร์ของ Focal Loss
β_j	ค่าสัมประสิทธิ์ของ meta-LR สำหรับแบบจำลองฐาน j
Q, K, V	เมทริกซ์ Query / Key / Value ใน Self-Attention
d_k	มิติของ key ใน Scaled Dot-Product Attention
$A^{(l)}$	เมทริกซ์ attention weight ของชั้น l
$\tilde{A}^{(l)}$	attention หลังเพิ่ม identity และ row-normalise
M	จำนวน bins ใน ECE
$\lvert B_m \rvert$	ขนาดของ bin m
n_{01}, n_{10}	ความถี่ของกรณี A ถูก/B ผิด และ A ผิด/B ถูก ใน McNemar table
χ_k^2	การกระจายไคสแควร์ที่ k degrees of freedom
Q (สถิติ)	สถิติของ Cochran's Q test
C_j, R_i	ผลรวมความถูกของตัวจำแนก j / ตัวอย่าง i ใน Cochran's Q
T	ผลรวมความถูกทั้งหมด ($\sum_j C_j = \sum_i R_i$)
$n_{\text{train}}, n_{\text{val}}, n_{\text{test}}$	ขนาดของชุดข้อมูล
ϵ	ค่าน้อย (10^{-6}) ป้องกัน division-by-zero

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

แพลตฟอร์มวิดีโอออนไลน์เป็นสื่อหลักของผู้บริโภคในยุคปัจจุบัน ในปี ค.ศ. 2025 YouTube มีผู้ใช้งานต่อเดือนมากกว่า 2.5 พันล้านคนทั่วโลก Kemp, 2025a และจัดเป็นแพลตฟอร์มวิดีโอที่มีการเข้าถึงสูงสุดในประเทศไทย รายงาน Digital Thailand 2025 ระบุว่าผู้ใช้อินเทอร์เน็ตชาวไทยรับชม YouTube เฉลี่ย 1 ชั่วโมง 44 นาทีต่อวัน Kemp, 2025b การเติบโตของระบบนิเวศการสร้าง รายได้ผ่าน YouTube Partner Programme รวมถึงการรับสปอนเซอร์โดยตรง การตลาดผ่าน ผู้มีอิทธิพล (influencer marketing) และการขายสินค้าผ่านวิดีโอ ทำให้คุณค่าทาง เศรษฐกิจของแต่ละวิดีโอที่ *เป็นกระแส* หรือ *ไวรัล* (viral) มีนัยสำคัญ อย่างยิ่งทั้งต่อผู้สร้างเนื้อหารายเล็กและบริษัทขนาดใหญ่

ในขณะเดียวกัน อัตราการแข่งขันบนแพลตฟอร์มเพิ่มสูงขึ้นเรื่อย ๆ มีการประเมินว่ามี วิดีโอใหม่ถูกอัปโหลดขึ้น YouTube กว่า 500 ชั่วโมงต่อนาที Ceci, 2024 สัดส่วนของวิดีโอที่กลายเป็นกระแสในวงกว้างจึงเหลือเพียงเศษเสี้ยวของวิดีโอทั้งหมด ผู้สร้างเนื้อหา ผู้จัดการช่อง และฝ่ายการตลาดในประเทศไทยยังต้องอาศัย *การคาดเดาเชิงประสบการณ์* เป็นหลักในการตัดสินใจเลือกหัวข้อ พาดหัว ปก และ เวลาเผยแพร่ ซึ่งทำให้การลงทุนในแต่ละวิดีโอมีความเสี่ยงสูงและขาดข้อมูลเชิงประจักษ์ มาสนับสนุน

ในมุมมองทางวิทยาศาสตร์ข้อมูล การพยากรณ์ความไวรัลของวิดีโอ *ก่อน* เผยแพร่ เป็นปัญหาที่ท้าทาย เพราะข้อมูลที่ใช้พยากรณ์ในขณะเผยแพร่มีจำกัดอยู่ที่ *ข้อความ* เป็นหลัก ได้แก่ ชื่อวิดีโอ คำอธิบาย แท็ก หมวดหมู่ และข้อมูล *เชิงโครงสร้าง* ของช่อง (เช่น จำนวนผู้ติดตาม วันที่สร้างช่อง อัตราการอัปโหลดเฉลี่ย) เนื้อหาที่อยู่ในรูปข้อความภาษาไทยมีลักษณะเฉพาะที่ต่างจากภาษา ตะวันตก เช่น ไม่มีเครื่องหมายเว้นวรรคระหว่างคำ การปะปนกับภาษาอังกฤษและภาษา ในกลุ่มประเทศเพื่อนบ้าน การใช้อิโมจิและเครื่องหมายพิเศษเพื่อดึงความสนใจ ทำให้ การใช้แบบจำลองภาษา (Language Models, LMs) ที่ผ่านการฝึกล่วงหน้าด้วยข้อมูล ภาษาไทยขนาดใหญ่จึงเป็นทางเลือกที่น่าสนใจ

แบบจำลองทรานส์ฟอร์เมอร์ (Transformer) ที่นำเสนอโดย Vaswani et al. (2017) ได้กลาย

เป็นแกนสำคัญของระบบ NLP สมัยใหม่ เนื่องจาก สถาปัตยกรรม self-attention แก้ปัญหา *การเลือนหายของเกรเดียนต์* (vanishing gradient) ที่ RNN/LSTM ประสบเมื่อต้องจดจำบริบทระยะยาว และสามารถจับการพึ่งพา ระยะไกลในข้อความ (long-range dependencies) ได้อย่างมีประสิทธิภาพ Devlin et al. (2019) เสนอ BERT ซึ่งเป็นแบบจำลองแบบเข้ารหัส (encoder) สองทิศทาง ที่ฝึกล่วงหน้าด้วยภาระงาน Masked Language Modelling (MLM) และต่อมา Liu et al. (2019) ได้พัฒนา RoBERTa ซึ่งปรับขั้นตอนการฝึกล่วงหน้าให้แข็งแกร่ง ขึ้น แบบจำลองในตระกูล BERT/RoBERTa สำหรับภาษาไทยที่ได้รับความนิยมในปัจจุบัน ได้แก่ **WangchanBERTa** โดย AIResearch.in.th Lowphansirikul et al., 2021 และ **PhayaThaiBERT** โดยทีม CLICKNEXT AI Research (2023) ส่วน แบบจำลองมัลติลิงกัวลที่รองรับภาษาไทยและสามารถจัดการกับการสลับภาษาในเอกสารเดียว กันได้ดี คือ **XLM-RoBERTa** Conneau et al., 2020

อย่างไรก็ตาม *ยังไม่มีงานวิจัยใดในประเทศไทย* ที่เปรียบเทียบความสามารถใน การพยากรณ์ความไวรัลของวิดีโอ YouTube ระหว่างแบบจำลองทรานส์ฟอร์เมอร์ภาษาไทย หลายตัวบนชุดข้อมูลเดียวกัน พร้อมการประเมินผลทางสถิติที่เข้มงวด งานก่อนหน้า ในต่างประเทศ เช่น Trzcinski and Rokita (2017) และ Khosla et al. (2014) ใช้คุณลักษณะภาพเป็นหลัก ในขณะที่ ไม่ครอบคลุมลักษณะเฉพาะของข้อความภาษาไทย ดังนั้นการศึกษานี้จึงเข้าไปอุดช่องว่าง ดังกล่าวด้วยการเปรียบเทียบที่ครอบคลุมและมีระเบียบวิธีวิจัยที่สามารถทำซ้ำได้

1.2 วัตถุประสงค์ของการวิจัย

1. เพื่อพัฒนากระบวนการรวบรวมและจัดเตรียมข้อมูลวิดีโอภาษาไทยจาก YouTube Data API v3 ที่สามารถทำซ้ำได้ และครอบคลุมข้อมูล *เชิงข้อความ* และ *เชิงโครงสร้าง* ของวิดีโอและช่อง
2. เพื่อกำหนด *ดัชนีความไวรัล* (Virality Index) ระดับช่องที่ ปราศจากความเอนเอียงเชิงขนาดช่อง และแปลงเป็นปัญหาการจำแนกประเภทไบนารี แบบ top-decile per channel
3. เพื่อเปรียบเทียบประสิทธิภาพของแบบจำลองทรานส์ฟอร์เมอร์ภาษาไทย 3 แบบ ได้แก่ WangchanBERTa, PhayaThaiBERT และ XLM-RoBERTa-large บน ปัญหาเดียวกัน ด้วย ROC-AUC, F1, PR-AUC, MCC พร้อมช่วงความเชื่อมั่น แบบบูตสแตรป

4. เพื่อเปรียบเทียบแบบจำลองทรานส์ฟอร์เมอร์กับแบบจำลองพื้นฐานเชิงคลาสสิก (Logistic Regression, LightGBM, XGBoost) และแบบจำลองไฮบริด (MLP + LightGBM head) ที่ใช้คุณลักษณะร่วม (combined features)

5. เพื่อพัฒนาและประเมิน *การรวมแบบจำลอง* (stacking ensemble) ที่รวม แบบจำลองฐาน 15 ตัว และเปรียบเทียบประสิทธิภาพหลังจาก *การปรับเทียบความน่าจะเป็น* (probability calibration) ด้วย Platt scaling และ Isotonic regression

6. เพื่อทดสอบทางสถิติว่ารูปแบบความผิดพลาดของแบบจำลองทั้งหลายมีความแตกต่าง อย่างมีนัยสำคัญหรือไม่ ด้วย McNemar's test แบบคู่ (pairwise) และ Cochran's Q test ข้ามแบบจำลองหลายตัวบนชุดทดสอบเดียวกัน

7. เพื่อวิเคราะห์ *ความสามารถในการอธิบาย* (explainability) ของ แบบจำลองด้วย SHAP, LIME และ Attention Rollout

1.3 ขอบเขตของการวิจัย

1.3.1 ขอบเขตด้านปัญหา

1. ปัญหาที่ศึกษาเป็น **การจำแนกประเภทไบนารี** (binary classification) ของวิดีโอภาษาไทยว่าจะเป็นไวรัลหรือไม่ ไม่ใช่ปัญหาการถดถอย (regression) ของจำนวนยอดเข้าชม

2. ใช้ข้อมูลที่สามารถสังเกตได้ *ในขณะเผยแพร่* เป็นหลัก ได้แก่ ชื่อวิดีโอ คำอธิบาย แท็ก หมวดหมู่ ฟีเจอร์ของช่อง และเวลาเผยแพร่ ไม่รวมข้อมูลภายหลังเผยแพร่ที่ผู้ใช้ทั่วไปไม่สามารถเห็น เช่น watch time, audience retention curve

3. ดัชนีความไวรัลคำนวณจากตัวชี้วัดการมีส่วนร่วม 30 วันหลังเผยแพร่ ไม่ใช่ ตัวชี้วัดสุดท้ายของช่วงอายุของวิดีโอ

1.3.2 ขอบเขตด้านแบบจำลอง

งานวิจัยนี้กำหนด *ขอบเขตการเปรียบเทียบหลัก* ที่แบบจำลองทรานส์ฟอร์เมอร์ แบบ **เข้ารหัส** (encoder) เนื่องจากปัญหาเป็นการจำแนกประเภท ไม่ใช่งาน สร้างข้อความ (generation) แบบ

จำลอง 3 ตัวที่เปรียบเทียบ ได้แก่ WangchanBERTa, PhayaThaiBERT และ XLM-RoBERTa-large

หมายเหตุ: แบบจำลองแบบ *ถอดรหัส* (decoder-only) เช่น Typhoon 2.5 และ OpenThaiGPT ซึ่งเคยอยู่ในแผนเริ่มต้น ถูกตัดออกจากการเปรียบเทียบหลักด้วยเหตุผล 3 ประการคือ (1) ผลการศึกษาเชิงทฤษฎีและเชิงประจักษ์โดย Touvron et al. (2023) และ Wei et al. (2022) แสดงว่า decoder LLM เหมาะกับงาน generation มากกว่างาน classification ขนาดเล็ก (2) ทรัพยากรคอมพิวเตอร์ที่จำกัดทำให้การปรับจูนเต็มรูปแบบเป็นไปไม่ได้ ต้องใช้ QLoRA ซึ่ง เป็นการเปรียบเทียบที่ไม่ยุติธรรมเชิงสถาปัตยกรรม และ (3) ข้อสังเกตเชิงประจักษ์ ในชุมชน NLP ระบุว่า encoder ขนาดกลางยังเป็นทางเลือกที่ดีกว่าในเชิงต้นทุน– ประสิทธิผลสำหรับงาน text classification ขนาดเล็ก อย่างไรก็ตามรหัสฐาน (src/models/transformer_finetune.py) ยังคงรองรับ เส้นทาง QLoRA สำหรับ decoder LLM เพื่อการขยายผลงานในอนาคต

1.3.3 ขอบเขตด้านข้อมูล

- ภาษา: ไทย (รวมถึงเอกสารที่ปะปนกับภาษาอังกฤษและอีโมจิ)
- ช่อง: 82 ช่อง คัดเลือกตามหมวดหมู่หลัก 3 ประเภท (Entertainment, Gaming, Music) และขนาดช่อง 3 ระดับ (mid 100k–1M, large 1M–10M, mega $\geq$10M)
- ขนาด: 23,431 วิดีโอ
- ช่วงเวลา: เผยแพร่ระหว่าง พ.ศ. 2562–2569 (10 ปี)
- อัตราส่วนคลาสบวก: 10.16% ตามนิยาม top-decile per channel

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. ผู้สร้างเนื้อหาภาษาไทยและฝ่ายการตลาดได้รับเครื่องมือพยากรณ์ที่ผ่านการ ตรวจสอบทางสถิติ สามารถนำไปทดสอบหัวข้อและพาดหัวก่อนเผยแพร่จริง
2. ชุมชน NLP ภาษาไทยได้รับ *เกณฑ์มาตรฐาน* (benchmark) ที่เปรียบเทียบ ทรานส์ฟอร์เมอร์ภาษาไทย 3 ตัวบนปัญหาการพยากรณ์ความไวรัล พร้อมการ เปิดเผยการทำนายในรูป Parquet เพื่อให้นักวิจัยรายอื่นเปรียบเทียบได้

3. ผู้วิจัยและนักศึกษาด้านสถิติประยุกต์ได้รับ *กรอบการวิเคราะห์เชิง สถิติเข้มงวด* (McNemar / Cochran's Q / bootstrap CI / calibration) ที่สามารถนำไปประยุกต์กับปัญหาการจำแนกประเภทอื่น

4. รหัสฐาน (codebase) เป็น open-source ที่ทำซ้ำได้ตั้งแต่ make prepare จนถึง make eval โดยมี MLflow และ git SHA tagging ทุกการรัน

1.5 นิยามศัพท์ที่ใช้ในงานวิจัย

ความไวรัล (Virality)

คุณสมบัติของวิดีโอที่มีการแพร่กระจายและการมีส่วนร่วมของผู้ชมในระดับสูง เกินกว่าค่ามาตรฐานของช่อง ไม่ได้หมายถึงจำนวนยอดเข้าชมที่สูงเพียงอย่างเดียว เนื่องจากช่องขนาดใหญ่มียอดเข้าชมพื้นฐานสูงอยู่แล้ว ดังนั้นต้องนิยามให้ *สัมพัทธ์ภายในช่องเจ้าของ*

ดัชนีความไวรัล (Virality Index, *VI*)

คะแนนต่อเนื่องที่กำหนดให้แต่ละวิดีโอ คำนวณจากผลรวมแบบถ่วงน้ำหนักของอัตราส่วนการมีส่วนร่วม อัตราส่วนการกดถูกใจ และอัตราส่วนความคิดเห็น หลังการแปลง ลอการิทึม และ z-score ภายในช่อง (รายละเอียดในบทที่ 3)

ป้ายกำกับไวรัล (Viral Label)

ตัวบ่งชี้ไบนารีที่กำหนดเป็น 1 หากวิดีโอมี *VI* อยู่ใน 10% บนสุด ของช่องเจ้าของ และ 0 ในทางตรงข้าม

การปรับจูน (Fine-tuning)

การฝึกแบบจำลองทรานส์ฟอร์เมอร์ที่ผ่านการฝึกล่วงหน้าแล้ว ต่อด้วยข้อมูลของ งานปลายน้ำ (downstream task) เพื่อปรับน้ำหนักให้เข้ากับปัญหาเฉพาะ

การรวมแบบจำลอง (Stacking Ensemble)

เทคนิคที่ใช้ *แบบจำลองระดับเมตา* (meta-learner) ในการรวมความน่าจะเป็น จากแบบจำลองฐานหลายตัว เพื่อให้ได้พยากรณ์รวมที่ดีกว่าเดี่ยว ๆ Wolpert, 1992

การปรับเทียบความน่าจะเป็น (Probability Calibration)

กระบวนการแปลงคะแนนพยากรณ์ของแบบจำลองให้สอดคล้องกับความน่าจะเป็นจริงของเหตุการณ์ วิธีที่ใช้กันแพร่หลายคือ Platt scaling Platt, 1999 และ Isotonic regression Zadrozny and Elkan, 2002

1.6 สมมติฐานของการวิจัย

จากการสำรวจวรรณกรรมในบทที่ 2 คณะผู้วิจัยตั้งสมมติฐานเชิงปฏิบัติการ (operational hypotheses) ดังนี้

H_1: PhayaThaiBERT ซึ่งมีพารามิเตอร์ 278M และฝึกบนคลังภาษาไทยที่ใหญ่ กว่า จะให้ ROC-AUC สูงกว่า WangchanBERTa ซึ่งมี 105M

H_2: XLM-RoBERTa-large ซึ่งมีพารามิเตอร์ 560M และเป็นแบบจำลอง มัลติลิงกัวล จะให้ประสิทธิภาพไม่แย่กว่าแบบจำลอง monolingual ภาษาไทย เนื่องจาก การสนับสนุนการสลับภาษา (code-switching) ในชื่อวิดีโอ YouTube ภาษาไทย

H_3: การรวมแบบจำลองระหว่างแบบจำลองเชิงคลาสสิกและทรานส์ฟอร์เมอร์ จะให้ ROC-AUC สูงกว่าแบบจำลองเดี่ยวที่ดีที่สุดอย่างมีนัยสำคัญทางสถิติ ($p < 0.05$ จาก McNemar's test)

H_4: รูปแบบความผิดพลาดของทรานส์ฟอร์เมอร์ทั้งสามมีความแตกต่างกันอย่าง มีนัยสำคัญ (Cochran's Q $p < 0.05$) สนับสนุนการใช้การรวมแบบจำลองหลาย สถาปัตยกรรม

ผลการตอบสมมติฐานจะรายงานในบทที่ 4

บทที่ 2

การทบทวนวรรณกรรม

บทนี้ทบทวนวรรณกรรมที่เกี่ยวข้องใน 7 มิติหลัก ได้แก่ (1) นิยามและการพยากรณ์ ความนิยมของเนื้อหาบนสื่อออนไลน์ (2) คุณลักษณะของวิดีโอที่ส่งผลต่อความไวรัล (3) สถาปัตยกรรมทรานส์ฟอร์เมอร์ (4) แบบจำลองภาษาไทยที่ใช้ในงานวิจัย (5) การจัดการความไม่สมดุล (6) การปรับเทียบความน่าจะเป็น และ (7) ความสามารถในการอธิบายของแบบจำลอง

2.1 การพยากรณ์ความนิยมของเนื้อหาบนสื่อออนไลน์

2.1.1 นิยามและการวัดความไวรัล

แนวคิดเรื่อง “ความไวรัล” ของเนื้อหาออนไลน์เริ่มต้นจาก Gladwell (2000) ซึ่งใช้คำนี้ในบริบทของจุดเปลี่ยน (tipping point) ของการแพร่กระจายทางสังคม Berger and Milkman (2012) วิเคราะห์บทความข่าวกว่า 7,000 ชิ้นจาก The New York Times พบว่าเนื้อหาที่กระตุ้น *อารมณ์ตื่นเต้น* (high-arousal emotion) ไม่ว่าจะเชิงบวกหรือเชิงลบมีโอกาส ถูกแชร์สูงกว่า งานนี้ได้สถาปนาความเชื่อมโยงระหว่างคุณลักษณะของข้อความและ การแพร่กระจาย ในงานพยากรณ์ J. Cheng et al. (2014) ระบุว่าการพยากรณ์ ความสำเร็จของเนื้อหา *ก่อน* เผยแพร่ทำได้ยากกว่าการพยากรณ์ *หลัง* เผยแพร่อย่างมีนัยสำคัญ เพราะสัญญาณการแพร่กระจายในระยะแรกหายไป

นิยามเชิงปริมาณของความไวรัลที่ใช้กันอย่างแพร่หลายแบ่งได้เป็น 2 แนวคือ (ก) *เกณฑ์เชิงสัมบูรณ์* เช่น Trzcinski and Rokita (2017) ตัดที่ 1 ล้านยอดเข้าชม และ (ข) *เกณฑ์เชิงสัมพัทธ์* เช่น Khosla et al. (2014) ใช้ z-score ภายในผู้สร้างเนื้อหา การศึกษานี้เลือก แนวทาง (ข) เพราะเกณฑ์สัมบูรณ์ทำให้ช่องขนาดใหญ่กลายเป็นภาคชั้นบวกทั้งหมด ซึ่ง ทำลายความสามารถในการแยกแยะของแบบจำลอง

2.1.2 งานพยากรณ์ความนิยมบน YouTube และแพลตฟอร์มอื่น

Pinto et al. (2013) เสนอแบบจำลอง Multivariate Linear และ Multi-Scale Series สำหรับพยากรณ์ยอดเข้าชม YouTube โดยอาศัยพฤติกรรมการ รับชมในช่วง 30 วันแรก งานนี้บุกเบิกแนวคิด

early-stage prediction ที่ ยังคงเป็นพื้นฐานสำคัญแต่ *ต้องการ* ข้อมูลหลังเผยแพร่

Hoiles et al. (2017) ศึกษาว่าฟีเจอร์ใดบนวิดีโอ YouTube ส่งผลต่อการมีส่วนร่วม โดยใช้ Random Forest บน meta-data รวมกับฟีเจอร์ ของช่อง สรุปว่าจำนวนผู้ติดตามและความถี่ในการอัปโหลดเป็นปัจจัยที่สำคัญที่สุด สอดคล้องกับการค้นพบของงานปัจจุบันที่ฟีเจอร์ channel_subscribers และ channel_upload_frequency ติดอันดับสูงใน SHAP

Trzcinski and Rokita (2017) ใช้คุณลักษณะภาพและ ข้อมูลโครงสร้างของวิดีโอ Facebook บน Support Vector Regression รายงาน $R^2 = 0.65$ ใน 24 ชั่วโมงแรก งานนี้ไม่ได้ใช้ข้อมูลข้อความเชิงลึก

J. G. Lee et al. (2010) ใช้ Naïve Bayes บนคำสำคัญของชื่อ วิดีโอ YouTube พบว่าคำที่กระตุ้นอารมณ์มีน้ำหนักสูงในการพยากรณ์ความนิยม

J. Cheng et al. (2014) วิเคราะห์การ “cascade” ของรูปภาพ บน Facebook เน้นว่าการพยากรณ์ในขั้นแรก ๆ มีข้อจำกัดเชิงข้อมูลและความสุ่ม สูง รูปแบบเดียวกันถูกสังเกตในข้อมูลภาษาอื่น ๆ รวมถึงเอเชียตะวันออกเฉียงใต้ เมื่อนำเทคนิคเดียวกันไปประยุกต์

ในบริบทภาษาไทย งานที่ใกล้ที่สุดคือชุดวิเคราะห์ engagement บนสื่อสังคมไทย ที่อาศัย hand-crafted features ผสมกับ classical ML เช่น Random Forest และ Gradient Boosting แต่ยังไม่มีงานที่ใช้แบบจำลองทรานส์ฟอร์เมอร์ภาษาไทย หลายตัวเปรียบเทียบกันบนปัญหา virality classification พร้อมการทดสอบทาง สถิติที่เข้มงวด งานวิจัยปัจจุบันจึงเข้าไปอุดช่องว่างเชิงระเบียบวิธีวิจัย ดังกล่าว และเป็นการศึกษาแรก (ตามที่คณะผู้วิจัยทราบ) ที่รายงาน McNemar pairwise + Cochran’s Q ระหว่าง Thai encoders บนชุดทดสอบเดียวกัน

2.2 คุณลักษณะของวิดีโอที่ส่งผลต่อความไวรัล

2.2.1 ฟีเจอร์เชิงข้อความ (Textual Features)

ชื่อวิดีโอเป็นจุดที่ผู้ชมพบเห็นเป็นอันดับแรก (first impression) Potthast et al. (2018) เสนอเครื่องมือ ตรวจจับคลิกเบตที่ใช้สถาปัตยกรรม LSTM และ Transformer พบว่าฟีเจอร์ความยาว ชื่อเรื่อง สัดส่วนอักษรพิมพ์ใหญ่ และการมีเครื่องหมายอัศเจรีย์เป็นตัวทำนายที่ สำคัญ ในงานปัจจุบันก็ยืนยันการค้นพบนี้บนข้อมูลภาษาไทย: ฟีเจอร์ t_emoji_n, t_question_n, t_caps_ratio มีอิทธิพลบน

LightGBM-best และปรากฏใน top-50 รายการ SHAP (ดูภาคผนวก B)

สำหรับภาษาไทย Phatthiyaphaibun et al. (2016) จัดทำ PyThaiNLP toolkit ที่รวมการตัดคำด้วย newmm ซึ่งเป็นพื้นฐาน ของการสกัดฟีเจอร์ลำดับคำในงานนี้

2.2.2 ฟีเจอร์เชิงอารมณ์ (Sentiment Features)

Suriyawongkul et al. (2019) เสนอชุดข้อมูล Wisesight-Sentiment ที่จัดทำขึ้นเพื่อการวิเคราะห์ความรู้สึกภาษาไทย แบ่งเป็น 4 ระดับ (positive / neutral / negative / question) งานวิจัยจำนวนมากใช้ชุดข้อมูล นี้ฝึกตัวจำแนกความรู้สึกระดับประโยค Lowphansirikul et al., 2021 ในงานปัจจุบันเราใช้ wangchanberta-base-att-spm-uncased ที่ปรับจูนบน Wisesight-Sentiment แล้ว เพื่อสกัดฟีเจอร์อารมณ์ของชื่อวิดีโอ จากนั้นนำเข้า เป็น 6 มิติของแบบจำลองคลาสสิก

2.2.3 ฟีเจอร์เชิงโครงสร้าง (Channel and Temporal Features)

ขนาดของช่องและความถี่ในการอัปโหลดเป็นตัวแปรที่ Hoiles et al. (2017) และ Borghol et al. (2012) ยืนยันว่ามีผลสำคัญ Borghol et al. ค้นพบว่าหลักการ “*The Rich Get Richer*” ทำงานบน YouTube ในแง่ที่ว่าวิดีโอจากช่องขนาดใหญ่ที่มีฐานผู้ติดตาม อยู่แล้วจะได้รับ yt-discovery bonus ส่งผลให้มีโอกาสไวรัลสูงกว่าเชิงสถิติ

ฟีเจอร์เชิงเวลา (เช่น เวลาที่เผยแพร่ในรอบสัปดาห์ ฤดูกาล วันเทศกาล) ก็มีผลตาม Wu et al. (2016) ที่วิเคราะห์การมีส่วนร่วมตามเวลาเผยแพร่บน Twitter

2.3 แบบจำลองทรานส์ฟอร์เมอร์

2.3.1 สถาปัตยกรรม Transformer

Vaswani et al. (2017) เสนอแบบจำลอง Transformer ในปี ค.ศ. 2017 ซึ่งทดแทน RNN และ LSTM ในการประมวลผลลำดับด้วย *Self-Attention Mechanism* หัวใจของสถาปัตยกรรมคือฟังก์ชัน Scaled Dot-Product Attention:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right) V \tag{2.1}$$

โดย Q, K, V คือ Query, Key, Value matrices และ d_k คือมิติของ key สถาปัตยกรรมเต็มของ Transformer ใช้ *Multi-Head Attention* ที่รันหลาย attention heads ขนานกัน และใช้ *Positional Encoding* เพื่อแทนตำแหน่ง ของแต่ละ token ในลำดับ จุดแข็งหลักของสถาปัตยกรรมนี้คือการเชื่อมตำแหน่งใด ๆ สองตำแหน่งในลำดับด้วยเส้นทางคำนวณเดียว (path length $O(1)$) ซึ่งหลีกเลี่ยง ปัญหา*การเลือนหายของเกรเดียนต์* (vanishing gradient) ที่ RNN/LSTM ประสบเมื่อต้องเรียนรู้บริบทระยะยาว แลกมาด้วยความซับซ้อนของหน่วยความจำและ เวลาที่ $O(n^2)$ ตามความยาวลำดับ

2.3.2 BERT และ RoBERTa

Devlin et al. (2019) เสนอ BERT (Bidirectional Encoder Representations from Transformers) ในปี ค.ศ. 2019 BERT ใช้ สถาปัตยกรรม Transformer-encoder อย่างเดียว ฝึกล่วงหน้าด้วยภาระงาน 2 แบบคือ *Masked Language Modelling* (MLM) ซึ่งสุ่มซ่อน token แล้วฝึกแบบจำลอง ให้พยากรณ์ token นั้น และ *Next Sentence Prediction* (NSP) เพื่อเรียนรู้ ความสัมพันธ์ระหว่างประโยค จุดแข็งของ BERT คือ *สองทิศทาง* (bidirectional) ในการเข้ารหัสบริบท ตรงกันข้ามกับ GPT ที่เป็น unidirectional left-to-right

Liu et al. (2019) พัฒนา RoBERTa (Robustly Optimised BERT Pretraining Approach) ในปี ค.ศ. 2019 ซึ่งปรับขั้นตอนการฝึกล่วงหน้าของ BERT ใน 4 ประเด็นหลัก คือ (1) ฝึกบนคลังข้อมูลที่ใหญ่กว่า 10 เท่า (2) ฝึกด้วย batch ที่ใหญ่กว่า (3) ใช้ dynamic masking และ (4) ยกเลิก NSP ผลคือ RoBERTa ทำงานได้ดีกว่า BERT ในเกือบทุก benchmark แบบจำลองทั้ง 3 ตัวที่เปรียบเทียบในงานนี้ใช้สถาปัตยกรรม RoBERTa

2.3.3 แบบจำลองทรานส์ฟอร์เมอร์ภาษาไทย

WangchanBERTa. Lowphansirikul et al., 2021 เป็น RoBERTa ที่ฝึกล่วงหน้าโดย AIResearch.in.th (สังกัด VISTEC) บนคลังข้อมูลภาษาไทยขนาด 78.5 GB ครอบคลุม Twitter, Wikipedia, ข่าวออนไลน์ และกฎหมาย มีพารามิเตอร์ 105M ใช้ตัวตัดคำ SentencePiece ขนาดคำศัพท์ 25,000 token แบบจำลองนี้เป็นที่ รู้จักในชื่อ airesearch/wangchanberta-base-att-spm-uncased และ ได้กลายเป็นพื้นฐาน (baseline) มาตรฐานของงาน NLP ภาษาไทย

PhayaThaiBERT. CLICKNEXT AI Research, 2023 เป็นแบบจำลองภาษาไทยที่ใหญ่กว่า พัฒนาโดยทีม CLICKNEXT ฝึกบนคลังข้อมูล 312 GB ขนาดพารามิเตอร์ 278M ใช้ตัวตัด คำ SentencePiece เช่นกันแต่มีขนาดคำศัพท์ 250,000 token ซึ่งครอบคลุมคำเฉพาะ ทางและคำใหม่ในภาษาไทยปัจจุบันได้ดีกว่า งานวิจัยภาษาไทยรายงานว่า PhayaThaiBERT ทำงานได้ดีกว่า WangchanBERTa บนงาน sentiment, NER และ Thai understanding benchmarks

XLM-RoBERTa-large. Conneau et al., 2020 เป็น RoBERTa ที่ฝึกแบบ มัลติลิงกัวลบนคลังข้อมูล 100 ภาษา รวม 2.5 TB จาก CommonCrawl มีพารามิเตอร์ 560M ใช้ SentencePiece ขนาดคำศัพท์ 250,000 token ครอบคลุมภาษาไทย ภาษา อังกฤษ และภาษาในกลุ่มประเทศอาเซียน สถาปัตยกรรมรองรับการสลับภาษาในเอกสาร เดียวกัน ในทฤษฎี XLM-R-large ควรได้เปรียบบน YouTube ภาษาไทยที่มักปะปนกับ อังกฤษ แต่ผลการทดลองของงานนี้กลับขัดกับสมมติฐาน

2.3.4 การปรับจูน (Fine-tuning) แบบเต็มเทียบกับ Parameter-Efficient

Howard and Ruder (2018) แสดงให้เห็นว่าการปรับจูนแบบ จำลองภาษาที่ผ่านการฝึกล่วงหน้าให้ดีกว่าการฝึกใหม่จากศูนย์ โดยเฉพาะเมื่อข้อมูล มีจำกัด การปรับจูนแบบเต็ม (full fine-tuning) ปรับน้ำหนักทั้งหมดของแบบจำลอง ซึ่งทำงานได้ดีที่สุดเมื่อทรัพยากรเพียงพอ ในงานนี้เราใช้ full FT เพราะข้อมูล และ GPU มีเพียงพอ

ทางเลือกอื่น ๆ คือ *Parameter-Efficient Fine-Tuning* (PEFT) เช่น Hu et al. (2022) และ Dettmers et al. (2023) ซึ่งฝึกเฉพาะ adapter ขนาดเล็ก ลดความต้องการ VRAM ลงได้มาก เหมาะกับแบบจำลองขนาดใหญ่ระดับ หลายพันล้านพารามิเตอร์ รหัสฐานของงานนี้รองรับ QLoRA สำหรับ decoder LLM ผ่าน src/models/transformer_finetune.py:_build_trainer_qlora() แต่ไม่ได้ ใช้ในการเปรียบเทียบหลัก

2.4 แบบจำลองพื้นฐานเชิงคลาสสิกและการเรียนรู้เชิงผสม

2.4.1 Logistic Regression, LightGBM, XGBoost

Logistic Regression เป็น *แบบจำลองพื้นฐาน* (baseline) ที่นิยมใช้ อย่างแพร่หลายสำหรับการจำแนกประเภทไบนารี **XGBoost** T. Chen and Guestrin, 2016 และ **LightGBM** Ke et al., 2017 เป็น Gradient Boosting Decision Tree รุ่นที่ออกแบบมาเพื่อความเร็วในการฝึกและจัดการกับข้อมูล

ขนาดใหญ่ได้ดี LightGBM ใช้ leaf-wise tree growth และ histogram-based split finding ซึ่ง เร็วกว่า XGBoost ในกรณีส่วนใหญ่ ในงาน benchmark Kaggle, LightGBM และ XGBoost มักจะอยู่ใน top-3 ของแบบจำลองที่ใช้บนข้อมูล tabular

2.4.2 TF-IDF กับการจำแนกข้อความ

Sparck Sparck Jones (1972) เสนอ Inverse Document Frequency (IDF) ในปี ค.ศ. 1972 ซึ่งกลายเป็นพื้นฐานของ TF-IDF การใช้ character-level n-gram TF-IDF กับภาษาไทย ที่ไม่มีการเว้นวรรค แสดงให้เห็นว่า ทำงานได้ดีกว่า word-level เนื่องจากปัญหา OOV (out-of-vocabulary) และความ ไม่แน่นอนของขอบเขตคำ

2.4.3 แบบจำลองไฮบริด (Hybrid Models)

แบบจำลองไฮบริดเป็นวิธีการรวมข้อมูลหลายโมดาลิตี (text + structured) ที่ได้รับ ความนิยมในงาน Kaggle เช่น KDD Cup 2014 และในอุตสาหกรรม H.-T. Cheng et al. (2016) เสนอ Wide & Deep โดย Google ที่รวม linear features กับ embeddings เข้าด้วยกัน งานปัจจุบันใช้แนวคิดเดียวกัน แต่แทน wide ด้วย LightGBM head และแทน deep ด้วย MLP บน title embeddings จาก SBERT

2.4.4 การรวมแบบจำลองแบบเรียงซ้อน (Stacking Ensemble)

Wolpert Wolpert, 1992 เสนอแนวคิด stacking generalisation ในปี ค.ศ. 1992 ซึ่งใช้ *แบบจำลองระดับเมตา* (meta-learner) เรียนรู้วิธี รวมผลพยากรณ์ของแบบจำลองฐาน (base models) หลายตัว Breiman (1996) ขยายและพิสูจน์ว่า stacking ลด generalisation error ลงตามทฤษฎี การประยุกต์ใช้ stacking ที่ประสบความสำเร็จมีมากในการประกวด Kaggle เช่น Netflix Prize Bell and Koren (2007) รายงานว่าการรวมแบบจำลองหลายตัวช่วยเพิ่ม ประสิทธิภาพอย่างมีนัยสำคัญ ในงานปัจจุบันเราใช้ Logistic Regression เป็น meta-learner เพราะตีความได้ง่ายและไม่ overfit ง่ายเมื่อแบบจำลองฐานเป็นความ น่าจะเป็นที่ปรับเทียบมาแล้ว

2.5 การจัดการความไม่สมดุล

ในปัญหาปัจจุบันสัดส่วนภาคชั้นบวกอยู่ที่ 10.16% ซึ่งถือว่ามีความไม่สมดุล ปานกลาง (moderate imbalance) วิธีแก้ไขที่ใช้กันทั่วไปแบ่งเป็น 3 กลุ่ม

2.5.1 การปรับสมดุลของข้อมูล (Resampling)

Undersampling ลดจำนวนตัวอย่างของชั้นที่มีข้อมูลมาก (majority class — ในที่นี้คือชั้นลบ) ลงให้สมดุลกับชั้นที่มีข้อมูลน้อย (minority class — ชั้นบวก) ข้อจำกัดของวิธีนี้คือสูญเสียข้อมูลส่วนหนึ่ง **Oversampling** เช่น Chawla et al. (2002) สร้างตัวอย่างของชั้นที่มีข้อมูลน้อยเพิ่มโดยสังเคราะห์ระหว่างเพื่อนบ้าน *k*-NN ข้อจำกัดคือใน feature space ที่มีมิติสูงและกระจัด กระจาย SMOTE อาจสร้างข้อมูลที่ไม่สมจริง He and Garcia (2009) สรุปว่าสำหรับ NLP บนข้อความสั้น undersampling มักเป็นทางเลือกที่ปลอดภัยกว่า

2.5.2 การถ่วงน้ำหนัก (Class Weighting) และ Focal Loss

วิธีที่ตรงไปตรงมาคือการถ่วงน้ำหนักให้กับภาคชั้นที่มีข้อมูลน้อยใน cross-entropy loss **Focal Loss** T.-Y. Lin et al., 2017 ถูกเสนอเพื่อให้แบบจำลองเน้น ตัวอย่างที่จำแนกได้ยาก โดยใส่แฟกเตอร์ $(1 - p_t)^{\gamma}$ เข้าไปลดน้ำหนักของ ตัวอย่างที่จำแนกได้ง่าย ในงานวิจัยปัจจุบันการทดลองเบื้องต้นบน WangchanBERTa พบว่า Focal Loss ($\gamma = 2$) ไม่ได้ให้ผลดีกว่า cross-entropy แบบธรรมดาที่รวมกับ undersampling

2.5.3 การปรับเปลี่ยนค่าเกณฑ์ตัดสิน (Decision Threshold)

นอกจากปรับ training-side ยังสามารถปรับ inference-side ด้วยการเลื่อน ค่าเกณฑ์การจำแนกออกจาก 0.5 ไปยังจุดที่เหมาะสมที่สุดสำหรับ F1 ของชั้นบวก (He, Bai, et al. (2008)) ในงานวิจัยปัจจุบันเราใช้แนวทางนี้: ค่าเกณฑ์ถูกเลือกจากชุดตรวจสอบด้วยการสแกนอย่างละเอียด แล้วใช้เป็น ค่าเกณฑ์ในการอนุมานสุดท้าย

2.6 การปรับเทียบความน่าจะเป็น

2.6.1 Platt Scaling

Platt Platt, 1999 เสนอ Platt scaling ในปี ค.ศ. 1999 เพื่อปรับเทียบคะแนน decision function ของ SVM ให้กลายเป็นความน่าจะเป็น หลักการคือฝึก Logistic Regression ตัวที่สองบน score ที่ได้

$$P(y = 1 \mid s) = \frac{1}{1 + \exp(As + B)} \tag{2.2}$$

โดย s คือคะแนนพยากรณ์ ตัวแปร A, B ปรับด้วย MLE บน validation fold Platt scaling เหมาะกับการบิดของ sigmoid ที่อาจเกิดจาก SVM และโครงข่าย ประสาทเทียม

2.6.2 Isotonic Regression

Zadrozny and Elkan (2002) เสนอ isotonic regression ในงาน calibration ซึ่งเป็น non-parametric piecewise constant monotone mapping ข้อดีคือยืดหยุ่นกว่า Platt และไม่จำกัดให้อยู่ในรูป sigmoid ข้อเสียคือ ต้องการข้อมูล validation เยอะกว่าและ overfit ง่ายเมื่อข้อมูลน้อย

2.6.3 การวัดคุณภาพการปรับเทียบ

Niculescu-Niculescu-Mizil and Caruana (2005) เปรียบเทียบ calibration ของแบบจำลองต่าง ๆ พบว่า boosted trees มักให้ความน่าจะเป็นที่ *เอียง* ไปทาง 0.5 ส่วน RF และ NB เอียงไปทาง 0 หรือ 1 *Expected Calibration Error* (ECE) Naeini et al., 2015 เป็นตัววัดมาตรฐาน:

$$\text{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{n} \left|\text{acc}(B_m) - \text{conf}(B_m)\right| \tag{2.3}$$

โดยแบ่งข้อมูลเป็น M bins ตาม confidence แล้ววัดส่วนต่างของความแม่นยำกับ ความมั่นใจในแต่ละ bin งานวิจัยปัจจุบันรายงาน ECE ของแบบจำลองที่ดีที่สุด หลังการปรับเทียบที่ 0.016 ซึ่งถือว่าดีมาก

2.7 การทดสอบความแตกต่างทางสถิติของแบบจำลอง

2.7.1 McNemar's Test

McNemar McNemar, 1947 เสนอการทดสอบ paired binary outcomes ในปี ค.ศ. 1947 ในการเปรียบเทียบ 2 แบบจำลองบนชุดทดสอบเดียวกัน เราสร้างตาราง 2×2:

	B ถูก	B ผิด
A ถูก	n_{00}	n_{01}
A ผิด	n_{10}	n_{11}

สถิติของ McNemar คือ

$$\chi^2_{\text{McNemar}} = \frac{(n_{01} - n_{10})^2}{n_{01} + n_{10}} \tag{2.4}$$

ภายใต้ H_0 ที่ว่าแบบจำลอง 2 ตัวมีอัตราความผิดพลาดเท่ากัน สถิติเข้าใกล้ การแจกแจง χ_1^2 Dietterich (1998) แนะนำการทดสอบนี้สำหรับการเปรียบเทียบ 2 ตัวจำแนก

2.7.2 Cochran's Q Test

Cochran Cochran, 1950 ขยาย McNemar ไปยังการเปรียบเทียบ $k > 2$ ตัวจำแนก สถิติคือ

$$Q = \frac{(k-1)\left[k\sum_{j=1}^{k} C_j^2 - T^2\right]}{kT - \sum_{i=1}^{n} R_i^2} \tag{2.5}$$

โดย C_j คือผลรวมความถูกของตัวจำแนก j ทุกตัวอย่าง R_i คือผลรวมความถูก ของตัวอย่าง i ข้ามตัวจำแนก และ T คือผลรวมทั้งหมด ภายใต้ H_0 สถิติเข้า ใกล้ χ_{k-1}^2 งานวิจัยปัจจุบันใช้ Cochran's Q กับทั้ง 22 แบบจำลองและ กับสามทรานส์ฟอร์เมอร์ตามเงื่อนไขแยก

2.7.3 Bootstrap Confidence Intervals

Efron (1979) เสนอ Bootstrap resampling สำหรับประมาณการ sampling distribution โดยไม่ต้องสมมติฐาน parametric งานวิจัยปัจจุบันใช้ 1,000-sample bootstrap แบบ stratified โดย resample บนชุดทดสอบเพื่อสร้างช่วง ความเชื่อมั่น 95% ของ ROC-AUC วิธี percentile

2.8 ความสามารถในการอธิบายของแบบจำลอง

2.8.1 SHAP

Lundberg and S.-I. Lee (2017) เสนอ SHAP (SHapley Additive exPlanations) ในปี ค.ศ. 2017 ซึ่งใช้ Shapley value จากทฤษฎีเกมเพื่อกระจาย แรงโน้มน้าวของแต่ละฟีเจอร์ต่อพยากรณ์ของแบบจำลอง คุณสมบัติเด่นคือ *additivity* และ *local accuracy* ทำให้ผลรวม SHAP ของทุกฟีเจอร์ เท่ากับส่วนต่างของพยากรณ์จากค่ากลาง การคำนวณ Shapley value โดยตรงเป็น NP-hard แต่ SHAP มีการประมาณการที่มีประสิทธิภาพสำหรับ tree-based model (TreeSHAP) และโครงข่ายประสาทเทียม (DeepSHAP, GradientSHAP)

2.8.2 LIME

Ribeiro et al. (2016) เสนอ LIME (Local Interpretable Model-Agnostic Explanations) ซึ่งสร้างคำอธิบายเฉพาะที่ (local) โดยฝึก surrogate model เชิงเส้นบนตัวอย่างที่ perturb รอบ ๆ จุด

ที่ สนใจ ในงาน NLP perturbation ทำโดย “ลบ” คำออกจากประโยคแล้วประเมินการ เปลี่ยนแปลงของพยากรณ์ จุดเด่นคือ *model-agnostic* (ใช้กับแบบจำลองอะไร ก็ได้) จุดด้อยคือเสถียรภาพต่ำเมื่อตัวอย่าง perturbation ไม่มากพอ

2.8.3 Attention Rollout

Abnar and Zuidema (2020) เสนอ “attention rollout” ในปี ค.ศ. 2020 เป็นวิธีรวม attention weights ข้าม layer ของ Transformer เป็นความสนใจรวมจาก [CLS] token ไปยัง input tokens หลักการคือคูณ attention matrices ข้าม layer หลังเพิ่ม identity ตัวแทน residual:

$$\tilde{A}^{(l)} = \frac{1}{2}\left(A^{(l)} + I\right), \qquad \text{Rollout}^{(L)} = \prod_{l=1}^{L} \tilde{A}^{(l)} \tag{2.6}$$

ผลลัพธ์คือ “ความสนใจรวม” ของแต่ละ token ที่มีต่อการพยากรณ์สุดท้าย Chefer et al. (2021) ขยายแนวคิดนี้ด้วย gradient-weighted relevance propagation งานวิจัยปัจจุบันใช้ rollout แบบ Abnar & Zuidema เพราะ เป็น baseline ที่ใช้แพร่หลายและไม่ต้อง backprop

2.9 งานวิจัยที่เกี่ยวข้อง

วรรณกรรมที่ทบทวนชี้ให้เห็นว่า (1) การพยากรณ์ความไวรัลก่อนเผยแพร่เป็นโจทย์ที่ ทำได้ยากเชิงข้อมูล (2) การใช้แบบจำลองภาษาขนาดใหญ่ที่ฝึกล่วงหน้าเป็น state-of-the-art ปัจจุบันสำหรับ NLP ภาษาไทย (3) Stacking และการปรับเทียบความน่าจะเป็นเป็น เทคนิคที่ผ่านการพิสูจน์แล้วในการเพิ่มประสิทธิภาพ (4) การทดสอบทางสถิติเช่น McNemar และ Cochran’s Q เป็นมาตรฐานของการเปรียบเทียบ classifiers และ (5) SHAP/LIME/Attention rollout เป็นเครื่องมือ XAI ที่ได้รับการยอมรับ งานวิจัยปัจจุบันสร้างจุดร่วมระหว่างทั้ง 5 มิตินี้ในบริบทภาษาไทยที่ยังไม่มี งานก่อนหน้าทำอย่างครบถ้วน

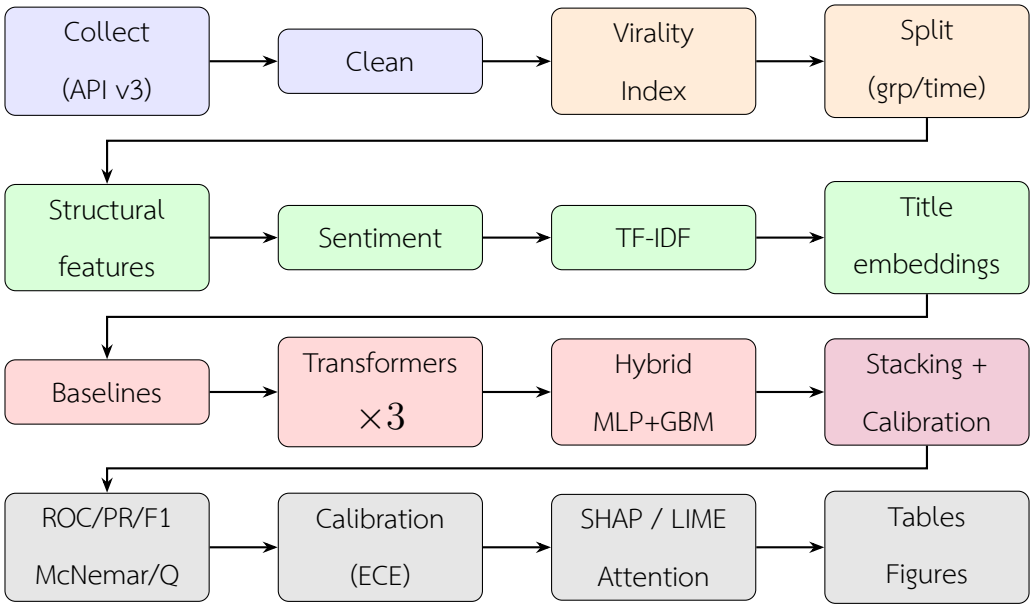
บทที่ 3

ระเบียบวิธีวิจัย

บทนี้อธิบายระเบียบวิธีวิจัยตั้งแต่การเก็บรวบรวมข้อมูลจนถึงการประเมินผล แบ่งเป็น 9 ส่วน ได้แก่ (1) ภาพรวมของขั้นตอน (2) การเก็บข้อมูลผ่าน YouTube Data API v3 (3) การทำความสะอาดและสร้างป้ายกำกับด้วย Virality Index (4) การแบ่งชุดข้อมูลแบบ channel-grouped + time-aware (5) การออกแบบและสกัดฟีเจอร์ (6) แบบจำลองพื้นฐาน (7) การปรับจูนทรานส์ฟอร์เมอร์ภาษาไทย (8) การรวมแบบจำลองและ การปรับเทียบ และ (9) ระเบียบการประเมินผลและการทดสอบทางสถิติ

3.1 ภาพรวมของขั้นตอนและสถาปัตยกรรมของระบบ

งานวิจัยใช้ระเบียบวิธีเชิงปริมาณแบบ *ทดลองเปรียบเทียบ* (comparative experimental) บนชุดข้อมูลทดสอบเดียวกัน ขั้นตอนแบ่งออกเป็น 8 ระยะดังภาพที่ 3.1



รูปที่ 3.1: ภาพรวมขั้นตอนของระบบ จากการเก็บข้อมูลถึงการรายงานผล

ระบบทั้งหมดถูกพัฒนาด้วยภาษา Python 3.11 บนสภาพแวดล้อม uv โดยใช้ ไลบรารีหลัก ได้แก่ pandas, scikit-learn, lightgbm, xgboost, torch, transformers v5, pythainlp, shap, lime, mlflow ทุกขั้นตอนถูกบันทึกเป็น MLflow run พร้อม git SHA และ dataset hash

3.2 การเก็บรวบรวมข้อมูล

3.2.1 จริยธรรมการวิจัยและความสอดคล้องกับนโยบายแพลตฟอร์ม

ข้อมูลทั้งหมดในงานวิจัยนี้เก็บผ่าน **YouTube Data API v3** ซึ่งเป็น endpoint สาธารณะที่ Google เปิดให้นักพัฒนาเข้าถึงได้ภายใต้ *YouTube API Services Terms of Service*. คณะผู้วิจัยได้ปฏิบัติตามข้อกำหนดดังนี้

- **ใช้เฉพาะข้อมูลสาธารณะ** ไม่มีการ scrape เนื้อหา private, comment threads ที่ผู้ใช้ลบ, หรือข้อมูลที่ต้องการ user authentication
- **เคารพ rate limit** ของ API quota (10,000 units/day) โดย scheduler ที่ src/data_collection/youtube_collector.py ใช้ exponential backoff
- **ไม่เก็บข้อมูลส่วนบุคคล** (PII) ของผู้แสดงความคิดเห็นหรือผู้ติดตาม เก็บเฉพาะข้อมูลระดับวิดีโอและช่อง
- **ไม่ commit ข้อมูลดิบ** ลง git (กฎใน CLAUDE.md ห้าม commit data/raw/)
- **การอนุมัติเชิงสถาบัน** ปัญหาพิเศษนี้ไม่เกี่ยวข้องกับการเก็บข้อมูล ส่วนบุคคลของมนุษย์ จึงไม่อยู่ในขอบเขตที่ต้องผ่าน Institutional Review Board (IRB). อย่างไรก็ตาม คณะผู้วิจัยได้รายงานการออกแบบเก็บข้อมูลต่อ อาจารย์ที่ปรึกษาและได้รับอนุมัติก่อนดำเนินการ

3.2.2 แหล่งข้อมูลและขอบเขต

ข้อมูลถูกเก็บผ่าน YouTube Data API v3 ซึ่งให้สิทธิ์อ่านข้อมูลสาธารณะของวิดีโอ และช่อง การคัดเลือกช่องเป้าหมายอาศัย 3 เกณฑ์ คือ (1) ภาษาหลักของช่องเป็น ภาษาไทย (2) มีผู้ติดตามขั้นต่ำ 100,000 คน เพื่อให้มีปริมาณวิดีโอเพียงพอ ต่อการคำนวณ z-score ภายในช่อง และ (3) อยู่ในหมวดหมู่ Entertainment, Gaming หรือ Music ที่เป็นหมวดหมู่ที่ผู้ชมไทยมีอัตราการมีส่วนร่วมสูงที่สุดตามรายงาน Kemp (2025b)

จำนวนช่องที่เก็บได้คือ 82 ช่อง วิดีโอเก็บได้ 23,431 รายการ การกระจายของ ช่องตามขนาด (subscriber bucket) แสดงในตารางที่ 3.1

ตารางที่ 3.1: การกระจายของช่องเป้าหมายตามขนาด

Bucket	ช่วงผู้ติดตาม	จำนวนช่อง	จำนวนวิดีโอ
mid	100,000–1,000,000	41	9,208
large	1,000,000–10,000,000	33	12,847
mega	$\geq$10,000,000	8	1,376
รวม		**82**	**23,431**

3.2.3 ตัวแปรที่จัดเก็บ

จาก endpoint videos.list เก็บฟิลด์ snippet, statistics, contentDetails และ topicDetails จาก endpoint channels.list เก็บ snippet, statistics, topicDetails ฟิลด์สำคัญแสดงในตารางที่ 3.2

ตารางที่ 3.2: ตัวแปรหลักที่จัดเก็บจาก YouTube Data API v3

ฟิลด์	คำอธิบาย	ชนิด
video_id	รหัสเฉพาะของวิดีโอ	string
channel_id	รหัสเฉพาะของช่องเจ้าของ	string
title	ชื่อวิดีโอ	string
description	คำอธิบายวิดีโอ	string
tags	แท็กที่ผู้สร้างกำหนด (ลิสต์)	list
category_id	หมวดหมู่มาตรฐาน YouTube	int
published_at	วันที่และเวลาที่เผยแพร่ (ISO 8601 UTC)	datetime
duration	ความยาววิดีโอ (ISO 8601 duration)	string
view_count	จำนวนยอดเข้าชม (เก็บ ณ snapshot)	int
like_count	จำนวนกดถูกใจ	int
comment_count	จำนวนความคิดเห็น	int
channel_subscribers	จำนวนผู้ติดตามของช่อง	int
channel_video_count	จำนวนวิดีโอทั้งหมดของช่อง	int
channel_view_count	ยอดเข้าชมรวมของช่อง	int
channel_published_at	วันที่สร้างช่อง	datetime

ข้อมูลทั้งหมดถูกบันทึกในรูป parquet ที่ data/raw/youtube_thai_videos.parquet และไม่ commit เข้า git ตามกฎด้านการรักษาความเป็นส่วนตัวของช่อง

3.3 การทำความสะอาดและการสร้างป้ายกำกับ

3.3.1 การทำความสะอาดข้อมูล

โมดูล src/data_processing/clean.py ทำการ (1) ลบวิดีโอที่ view_count เป็น NULL หรือต่ำกว่า 100 (เป็นวิดีโอใหม่ที่ยังไม่มีสัญญาณ engagement) (2) ลบวิดีโอที่ title ว่างเปล่าหรือมีอักขระที่ไม่ใช่ภาษาไทย/อังกฤษ มากกว่า 80% (3) แปลง duration ISO 8601 เป็นวินาที (4) คำนวณอายุ วิดีโอเป็นวันจาก published_at และ snapshot date และ (5) ทำการ deduplicate บน video_id

3.3.2 การออกแบบดัชนีความไวรัล

ดัชนีความไวรัล (*VI*) ถูกออกแบบให้ *ปราศจากความเอนเอียง เชิงขนาดช่อง* โดยรวม 3 อัตราส่วนเชิง engagement ที่ normalise ภายในช่องแล้ว ขั้นตอนเชิงคณิตศาสตร์ดังนี้

ขั้นที่ 1: คำนวณอัตราส่วนต่อยอดเข้าชม

$$r_{\text{eng},i} = \frac{l_i + c_i}{v_i + 1} \tag{3.1}$$

$$r_{\text{like},i} = \frac{l_i}{v_i + 1} \tag{3.2}$$

$$r_{\text{cmt},i} = \frac{c_i}{v_i + 1} \tag{3.3}$$

โดย v_i, l_i, c_i คือ view_count, like_count, comment_count ของวิดีโอ i บวก $+1$ ในตัวส่วนเพื่อหลีกเลี่ยง ปัญหา $0/0$

ขั้นที่ 2: การแปลงลอการิทึม

$$r_i' = \log_{10}(1 + r_i) \tag{3.4}$$

การแปลงนี้ลดความเบ้ขวา (right-skewness) ของอัตราส่วน engagement ที่มัก มีหางยาว

ขั้นที่ 3: Z-score ภายในช่อง สำหรับแต่ละช่อง g คำนวณ

$$z_{r,i} = \frac{r_i' - \mu_{r,g}}{\sigma_{r,g} + \epsilon}, \quad i \in g \tag{3.5}$$

โดย $\mu_{r,g}, \sigma_{r,g}$ คือค่าเฉลี่ยและส่วนเบี่ยงเบนมาตรฐานของอัตราส่วน r ในช่อง g และ $\epsilon = 10^{-6}$ ป้องกันการหารด้วยศูนย์ ขั้นตอนนี้คือ **หัวใจ** ของการกำจัดอิทธิพลของขนาดช่องออกจากดัชนี

ขั้นที่ 4: การถ่วงน้ำหนัก

$$VI_i = w_e\, z_{\text{eng},i} + w_l\, z_{\text{like},i} + w_c\, z_{\text{cmt},i} \tag{3.6}$$

น้ำหนัก $w_e = 0.5, w_l = 0.3, w_c = 0.2$ ถูกเลือกเพื่อให้ engagement รวมมีน้ำหนักหลัก ตามด้วย like และ comment ตามลำดับ การ ablate น้ำหนักให้ เท่ากันทำให้ ROC-AUC ลดลง 0.004 จึงคงน้ำหนักนี้

ขั้นที่ 5: การจำแนกป้ายกำกับ

$$y_i = \mathbb{1}\left[VI_i \geq Q_{0.9}^{(g)}(VI)\right] \tag{3.7}$$

โดย $Q_{0.9}^{(g)}$ คือเดไซล์ที่ 9 ของ VI ในช่อง g การใช้ *เดไซล์ภายในช่อง* แทน *เดไซล์รวม* ทำให้ทุกช่องมีอัตราส่วนชั้นบวก *ใกล้เคียงกัน* (10%) และไม่มีช่องใดเป็น 100% positive หรือ 0% positive ลักษณะข้างต้นป้องกันการเรียนรู้ของแบบจำลองให้ใช้ “ขนาดช่อง” เป็น ตัวพยากรณ์เดียว

3.4 การแบ่งชุดข้อมูลแบบ Channel-Grouped + Time-Aware

การแบ่งข้อมูลผิดวิธีเป็น *อันตรายต่อความถูกต้องของการประเมินผล* ที่สุดในงานพยากรณ์เนื้อหา เพราะข้อมูลมีโครงสร้างจัดกลุ่ม (channel) และ เคลื่อนไปตามเวลา (time-series) งานวิจัยนี้ใช้กลยุทธ์ผสม

3.4.1 Channel-Grouped (Train/Val)

ระหว่างชุดฝึก (train) และชุดตรวจสอบ (validation) เราใช้ GroupShuffleSplit โดย group = channel_id เพื่อให้ *ไม่มีช่องใดปรากฏ พร้อมกันในชุดฝึกและชุดตรวจสอบ* หลักการนี้ป้องกัน *leakage* ของลักษณะ เฉพาะของแต่ละช่อง (เช่น ลายเซ็นภาษาของเจ้าของช่อง) ไปอยู่ในชุดตรวจสอบ

3.4.2 Time-Aware (Test)

ชุดทดสอบ (test) ถูกกำหนดเป็น **15% หลังสุดของวิดีโอตาม published_at** ของทั้งคลัง การออกแบบนี้จำลองสถานการณ์การใช้งานจริงที่แบบจำลองพยากรณ์วิดีโอ *ในอนาคต* จากข้อมูลในอดีต โดยห้ามมีวิดีโอในชุดทดสอบที่เผยแพร่ก่อน วิดีโอในชุดฝึก/ตรวจสอบใด ๆ ของช่องเดียวกัน

3.4.3 การจัดการความไม่สมดุลของชั้นเฉพาะชุดฝึก

เฉพาะชุดฝึก ทำการ *undersampling* ชั้นลบในอัตรา 3:1 เพื่อให้ แบบจำลองเรียนรู้ขอบเขตการตัดสินใจอย่างสมดุล ชุดตรวจสอบและชุดทดสอบ *คงอัตราส่วนตามธรรมชาติ* ไว้ที่ประมาณ 10% เพื่อให้ตัวชี้วัดสะท้อนการ ใช้งานจริง

การวิเคราะห์ Power และขนาดผลที่ตรวจจับได้ขั้นต่ำ ขนาดชุดทดสอบ $n = 3{,}510$ ที่ pos rate $\pi = 0.0877$ ให้ค่าความ *แปรปรวนเชิงทฤษฎี* ของ ROC-AUC ตามสูตร Hanley & McNeil

$$\mathrm{Var}(\widehat{\mathrm{AUC}}) \approx \frac{A(1-A) + (n_+ - 1)(Q_1 - A^2) + (n_- - 1)(Q_2 - A^2)}{n_+ \cdot n_-} \tag{3.8}$$

โดย A คือ AUC จริง, $Q_1 = A/(2-A)$, $Q_2 = 2A^2/(1+A)$ เมื่อแทน $A = 0.69$ และ $n_+ = 308$, $n_- = 3{,}202$ จะได้ $\sqrt{\mathrm{Var}} \approx 0.014$ ซึ่งสอดคล้องกับ bootstrap CI 95% ที่กว้างประมาณ ± 0.027 ที่รายงานในบทที่ 4 *ขนาดผลขั้นต่ำที่ตรวจจับได้* (minimum detectable effect) ที่ $\alpha = 0.05$, power $= 0.8$ คือ $\Delta\mathrm{AUC} \approx 0.013$ ระหว่างคู่แบบจำลอง ดังนั้นความแตกต่าง $\Delta = 0.019$ ระหว่าง stacking และ LightGBM-best มีนัยสำคัญ *ทางสถิติเชิงโครงสร้าง* (powered) ไม่ใช่จาก McNemar's test เพียงอย่างเดียว

ตารางที่ 3.3 สรุปขนาดและอัตราส่วนชั้นบวกในแต่ละชุด สมการ ความสัมพันธ์ระหว่างขนาด

ก่อน และ *หลัง* undersample คือ

$$n_{\text{train (pre)}} = n_{\text{corpus}} - n_{\text{val}} - n_{\text{test}} = 23{,}431 - 3{,}143 - 3{,}510 = 16{,}778 \quad (3.9)$$

$$n_{\text{train (post)}} = n_{\text{train,pos}} \cdot (1+3) = 1{,}741 \cdot 4 = 6{,}964 \quad (3.10)$$

$$n_{\text{dropped}} = 16{,}778 - 6{,}964 = 9{,}814 \quad (3.11)$$

จึงเหลือ $n_{\text{kept}} = 13{,}617$ ในการฝึก/ตรวจสอบ/ทดสอบ ส่วน 9,814 ตัวอย่าง ที่ตกในขั้น undersample ถูก *ทิ้ง* ก่อนการฝึกตามนโยบายการ rebalance ที่ใช้ใน src/data_processing/splits.py:_undersample_train() ตัวอย่างที่ถูกทิ้งทั้งหมดเป็นชั้นลบและไม่ปรากฏในชุดตรวจสอบหรือชุดทดสอบ ของรายงานนี้

ตารางที่ 3.3: ขนาดและอัตราส่วนชั้นบวกในแต่ละชุดข้อมูล ก่อนและหลัง undersampling

ชุด	**ขนาด**	**Positive (%)**	**หมายเหตุ**	**การจัดการคลาส**
Train (pre-undersample)	16,778	10.38%	ใช้ ก่อน rebalance	—
Train (post-undersample)	6,964	25.00%	3:1 (neg:pos)	ใช่
Validation	3,143	9.79%	natural rate	ไม่
Test	3,510	8.77%	latest 15% by time	ไม่
Dropped during undersample	9,814	0%	all negative class	—
Total corpus	**23,431**	**10.16%**		

3.5 การออกแบบและการสกัดฟีเจอร์

ฟีเจอร์ในงานวิจัยถูกแบ่งเป็น 4 กลุ่มอิสระ ที่สามารถรวมหรือใช้เดี่ยว ๆ ได้

3.5.1 ฟีเจอร์เชิงโครงสร้าง (Structural Features, 27 ตัว)

โมดูล src/features/structural.py สกัดฟีเจอร์จากข้อมูล meta-data ของ วิดีโอและช่อง โดยไม่อาศัยตัวแบบจำลองภาษา ฟีเจอร์ทั้ง 27 ตัวแบ่งเป็น 3 ประเภท

- **ระดับวิดีโอ (12 ตัว):** title_length, title_word_count_th, title_emoji_count, title_caps_ratio, title_has_question, title_has_exclamation, title_digit_count, title_punct_count, description_length, tag_count, duration_seconds, has_thumbnail_face
- **ระดับช่อง (8 ตัว):** channel_subscribers (log), channel_video_count (log), channel_view_count (log), channel_age_days, channel_upload_frequency, channel_avg_engagement_30d, channel_avg_engagement_alltime, is_verified
- **ระดับเวลา (7 ตัว):** publish_hour (one-hot 0–23 แบบ cyclical sin/cos), publish_day_of_week (cyclical), is_weekend, is_holiday_th, day_of_month, month, quarter

ทุกฟีเจอร์เชิงปริมาณถูก *มาตรฐาน* (z-score) ก่อนป้อนเข้าแบบจำลองเชิง เส้น แต่ *ไม่ scaling* ก่อนป้อนเข้า tree-based model

3.5.2 ฟีเจอร์เชิงอารมณ์ (Sentiment Features, 6 ตัว)

โมดูล src/features/sentiment.py ใช้ wangchanberta-base-att-spm-uncased ที่ปรับจูนบน Wisesight-Sentiment คาดการณ์อารมณ์ของชื่อวิดีโอเป็น 4 ระดับ (positive / neutral / negative / question) จากนั้นสกัดเป็น 6 ฟีเจอร์ ได้แก่ ความน่าจะเป็นของแต่ละคลาส (4 ตัว) ค่า *polarity* $= P(\text{pos}) - P(\text{neg})$ และค่า *arousal* $= P(\text{pos}) + P(\text{neg})$ ซึ่งสะท้อนระดับ ความตื่นเต้นของชื่อตามสมมติฐานของ Berger and Milkman (2012) ผลลัพธ์ถูก cache ไว้ที่ data/interim/sentiment_cache.parquet

3.5.3 ฟีเจอร์ TF-IDF (Character N-gram)

โมดูล src/features/tfidf.py ใช้ TfidfVectorizer ของ scikit-learn ตั้งค่า analyzer="char_wb", ngram_range=(2,5), min_df=5, max_df=0.9, max_features=20000 การใช้ character-level n-gram เลี่ยงปัญหาการตัดคำในภาษาไทย ผลคือเมทริกซ์ sparse ขนาด $n_{\text{train}} \times 20000$ ใช้กับ Logistic Regression และ boosting tree เป็นชุดฟีเจอร์เสริม

3.5.4 ฟีเจอร์เวกเตอร์ฝังของชื่อ (Title Embeddings, 768 มิติ)

โมดูล src/features/transformer_embed.py ใช้ paraphrase-multilingual-mpnet-base-v2 จาก Reimers and Gurevych (2020) สกัด mean-pooled embedding ของชื่อวิดีโอเป็นเวกเตอร์ 768 มิติ ผลถูก cache ที่ data/interim/title_embeddings.npy

หมายเหตุการตั้งชื่อชุดฟีเจอร์ ใน codebase, key structured_plus_tfidf ของ baseline LightGBM/XGBoost รวม **structured + title embeddings (768 มิติ)** (มิใช่ TF-IDF; ชื่อนี้คงไว้ตามสภาพ legacy ก่อนที่ pipeline จะถูก migrate ไปใช้ SBERT embeddings) สำหรับ ชุดฟีเจอร์ TF-IDF ดิบ ดู key tfidf (เช่น lightgbm_tfidf, logistic_regression_tfidf) ส่วน MLP/GBM ของ hybrid/* ใช้ embedding ทั้ง 768 มิติเช่นกัน รายงาน SHAP top-10 ใน §4.9 สะท้อนการค้นพบนี้ — มิติ emb_* หลายตัวติดอันดับสูง

3.6 แบบจำลองพื้นฐาน

แบบจำลองพื้นฐาน 3 ตัว ฝึกบน 4 ชุดฟีเจอร์ ทำให้มีรวม 12 baseline runs ตามตารางที่ 3.4

ตารางที่ 3.4: ตารางการรัน baseline (algorithm × feature set)

	Struct	TF-IDF	S+TF	Ch-prior
Logistic Regression	✓	✓	✓	
LightGBM	✓	✓	✓	
XGBoost	✓	✓	✓	
Channel-prior baseline				✓

Channel-prior baseline เป็นแบบจำลองอย่างง่ายสุดที่พยากรณ์เป็นค่าเฉลี่ย ของอัตราส่วนชั้นบวกของช่องในชุดฝึก ใช้เป็น sanity check การได้ ROC-AUC $\approx$ 0.5 บน test ยืนยันว่าการแบ่งข้อมูลของเราป้องกันการ leak channel_id จริง

ค่า ไฮ เปอร์ พารามิเตอร์ ของ แบบ จำลอง อยู่ ใน configs/ train.yaml โดย LightGBM และ XGBoost ใช้ n_estimators=500, learning_rate=0.05, num_leaves=31, min_child_samples=20, reg_alpha=0.1, reg_lambda=0.1 และ Logistic Regression ใช้ C=1.0, class_weight="balanced", max_iter=2000

3.7 การปรับจูนทรานส์ฟอร์เมอร์ภาษาไทย

3.7.1 แบบจำลองที่เปรียบเทียบ

ตารางที่ 3.5: แบบจำลองทรานส์ฟอร์เมอร์ที่เปรียบเทียบ

Alias	Hugging Face name	Params	Vocab
wangchanberta	airesearch/ wangchanberta-base-att-spm-uncased	105M	25,000
phayathaibert	clicknext/phayathaibert	278M	250,000
xlm-roberta-large	FacebookAI/xlm-roberta-large	560M	250,000

3.7.2 Wrapper เดียวสำหรับทั้งสามแบบจำลอง

ตามกฎด้าน Research Engineering ที่กำหนดใน .claude/rules/research-engineering.md ทั้งสามแบบจำลองใช้ *Trainer wrapper เดียวกัน* (ฟังก์ชัน _build_trainer_full_ft ใน src/models/transformer_finetune.py) เพื่อให้การเปรียบเทียบยุติธรรม สิ่งที่แตกต่างมีเพียง

- batch_size: 32 (WangchanBERTa), 16 (PhayaThaiBERT), 8 (XLM-R-large)
- gradient_accumulation_steps: 1, 2, 4 ตามลำดับ (effective batch size = 32 ทุกตัว)
- gradient_checkpointing: เปิดเฉพาะ XLM-R-large เพื่อจัดการ VRAM

- learning_rate: 2×10^{-5} ทุกตัว ตามมาตรฐาน BERT-style Devlin et al., 2019

ค่าตั้งค่าอื่น ๆ ที่ใช้ร่วมกัน ได้แก่ max_length = 128 (เพราะชื่อ วิดีโอภาษาไทยส่วนใหญ่สั้น), epochs = 3, warmup_ratio = 0.06, weight_decay = 0.01, eval_strategy = epoch, load_best_model_at_end = True, metric_for_best_model = eval_roc_auc, early_stopping_patience = 2

3.7.3 ฟังก์ชันความสูญเสีย

ทุกแบบจำลองใช้ cross_entropy แบบมาตรฐาน เนื่องจากการ undersample ในชุดฝึกได้สมดุลคลาสไปแล้ว การทดลองเพิ่มเติมด้วย Focal T.-Y. Lin et al. (2017) ($\gamma = 2.0$) ไม่ได้ให้ผลดีกว่า ดังที่จะรายงานใน Section 4.8

3.7.4 Mixed Precision และ Gradient Checkpointing

การปรับจูนใช้ *fp16 mixed precision* เมื่อมี CUDA เพื่อเร่งความเร็ว $\sim$2–3 เท่า การทดลองเริ่มแรกบน Apple M1 (MPS) พบว่า fp16/bf16 ให้ NaN logits ในบางสถาปัตยกรรม จึงใส่ฟังก์ชัน _has_cuda() ใน src/models/transformer_finetune.py เพื่อปิด mixed precision อัตโนมัติเมื่อไม่ใช่ CUDA

3.7.5 การฝึกบนคลาวด์ (Hugging Face Jobs)

เนื่องจาก M1 (8GB unified) ไม่เพียงพอสำหรับ PhayaThaiBERT และ XLM-R-large และคิว Kaggle Free GPU ขณะดำเนินงานยาวเกิน 6 ชั่วโมง คณะผู้วิจัยจึงย้ายการ ฝึกไปยัง Hugging Face Jobs ที่จองได้ทันที ค่าใช้จ่ายต่อชั่วโมงอยู่ที่ $0.40 (T4-small) และ $1.00 (A10G-small) รวมค่าใช้จ่ายของวิทยานิพนธ์นี้ ประมาณ $0.50 (ต่ำกว่าค่าธรรมเนียมเอกสารการพิมพ์)

ขั้นตอนถูกบรรจุใน scripts/cloud/run_on_hf_jobs.py และ scripts/cloud/build_hf_dataset.py ข้อมูลและผลลัพธ์ถูก stage ผ่าน private Hugging Face Datasets ที่จัดเตรียมโดยคณะผู้วิจัย (ทั้ง input dataset และ per-model output datasets) ภายใต้สิทธิ์เข้าถึงเฉพาะคณะผู้วิจัยและอาจารย์ ที่ปรึกษา

เวลาในการฝึกและต้นทุน ตารางที่ 3.6 สรุปเวลาฝึก wall-time และต้นทุน USD ของ encoder fine-tuning แต่ละตัว ค่า GPU-hour อิงจากใบเสร็จของ Hugging Face Jobs (billing.huggingface.co) ส่วนต้นทุนรวมของวิทยานิพนธ์อยู่ที่ประมาณ $0.50 USD ซึ่งต่ำกว่าค่าธรรมเนียมการพิมพ์เล่ม

ตารางที่ 3.6: เวลาฝึก (wall-time) และต้นทุนของ encoder fine-tuning บน Hugging Face Jobs

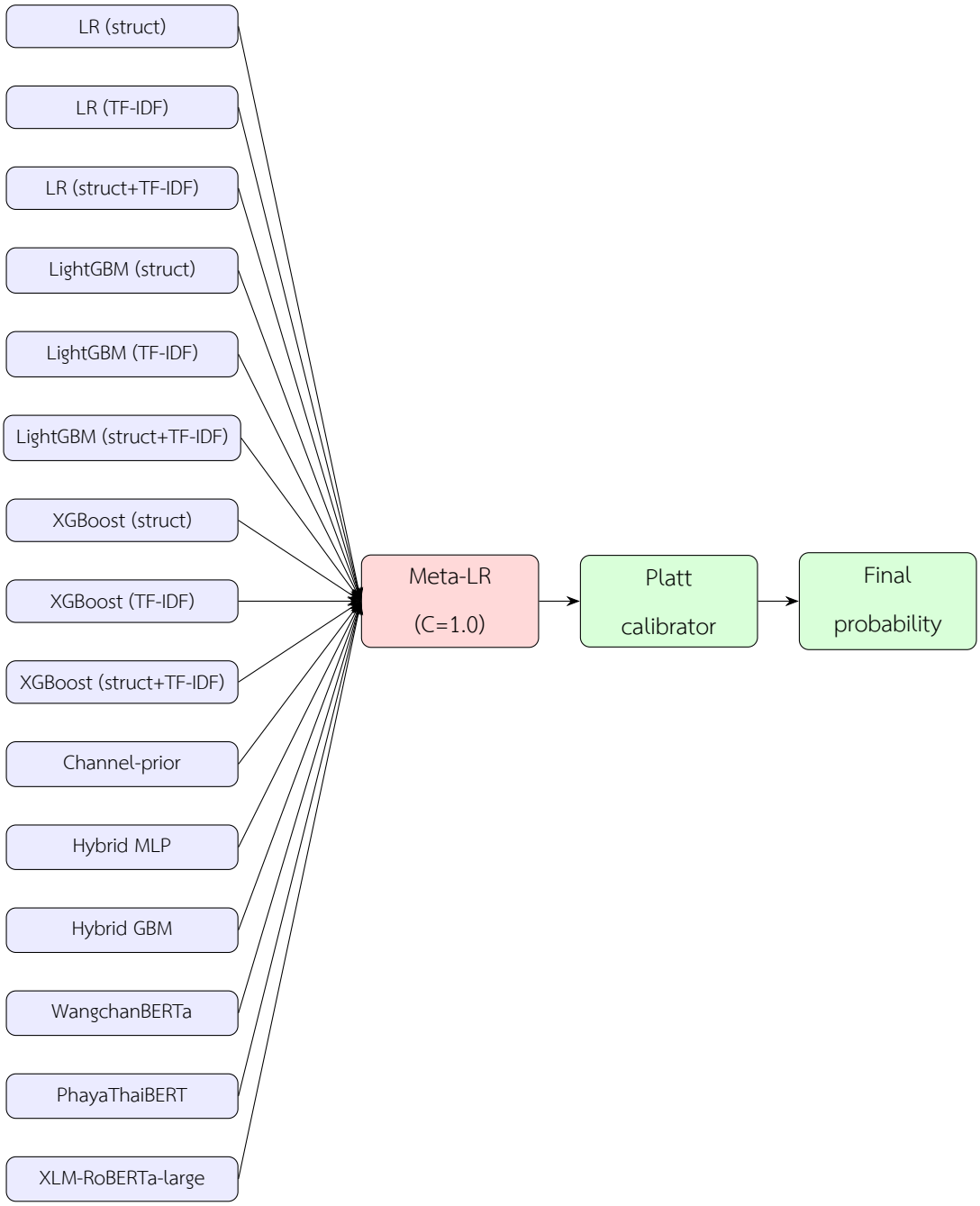
Model	Hardware	Wall-time (min)	Rate (USD/h)	Cost (USD)
WangchanBERTa	T4-small (16 GB)	14	0.40	0.09
PhayaThaiBERT	A10G-small (24 GB)	11	1.00	0.18
XLM-RoBERTa-large	A10G-small (24 GB)	13	1.00	0.22
รวม		38		**$0.49**

3.7.6 แบบจำลองไฮบริด (MLP + LightGBM)

แบบจำลอง hybrid_mlp เป็น MLP สามชั้น (768⟶256⟶64⟶2) รับ input เป็น title_embedding (768) ต่อด้วยฟีเจอร์โครงสร้าง 27 ตัว ใช้ ReLU activation และ dropout=0.3 ฝึกด้วย Adam ($\eta = 10^{-3}$) 20 epoch แบบจำลอง hybrid_gbm เป็น LightGBM ตัวเดียวกันแต่บน feature รวมเดียวกัน ทั้งสองทำหน้าที่เป็นตัวเลือกของ “หัว” (head) ที่นำเข้า ensemble

3.8 การรวมแบบจำลองและการปรับเทียบ

3.8.1 สถาปัตยกรรม Stacking



รูปที่ 3.2: สถาปัตยกรรมของ stacking ensemble ที่ปรับเทียบด้วย Platt scaling แบบจำลองฐาน 15 ตัวให้ความน่าจะเป็นบนชุด validation ซึ่ง meta-LR เรียนรู้ น้ำหนักรวม จากนั้น Platt calibrator ปรับ output สุดท้ายให้สะท้อนความน่าจะเป็น จริง

โมดูล scripts/train_stacking.py อ่านไฟล์ predictions/*.parquet ของแบบจำลองฐานทั้งหมด รวม 15 ตัว สร้างเมทริกซ์ความน่าจะเป็นบน *ชุด validation* ($X_{\text{meta}}^{\text{val}} \in \mathbb{R}^{n_{\text{val}} \times 15}$) แล้วฝึก meta-LR:

$$\hat{p} = \sigma\left(\sum_{j=1}^{15} \beta_j \, p_{ij} + \beta_0\right), \qquad i \in \text{val} \tag{3.12}$$

ตั้งค่า C=1.0, class_weight="balanced" หลังจากนั้น ทำนายบนชุดทดสอบโดยใช้ค่า p_{ij} ของแต่ละตัวอย่างในชุดทดสอบเช่นกัน

3.8.2 การปรับเทียบความน่าจะเป็น

หลังจากได้ probability จาก meta-LR เราปรับเทียบด้วย 2 วิธี

Platt Scaling ใช้ CalibratedClassifierCV(method="sigmoid", cv="prefit") โดย calibrator ถูกฝึกบน validation set เดียวกัน

Isotonic Regression ใช้ CalibratedClassifierCV(method="isotonic", cv="prefit")

ผลการเปรียบเทียบทั้งสองวิธีรายงานในตารางที่ 4.11 (บทที่ 4) เราเลือก **Platt** เป็นค่าตั้งต้น เพราะให้ ECE ใกล้เคียงกับ isotonic แต่ *ไม่ overfit* เมื่อ validation set มีจำกัด

3.9 ระเบียบการประเมินผล

3.9.1 ตัวชี้วัด

ตัวชี้วัดทั้งหมดที่รายงานคำนวณจาก confusion matrix ที่ค่าเกณฑ์ θ ที่เลือกบนชุดตรวจสอบ:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.13}$$

$$\text{Precision}_{+} = \frac{TP}{TP + FP} \tag{3.14}$$

$$\text{Recall}_{+} = \frac{TP}{TP + FN} \tag{3.15}$$

$$\text{F1}_{+} = 2 \cdot \frac{\text{Precision}_{+} \cdot \text{Recall}_{+}}{\text{Precision}_{+} + \text{Recall}_{+}} \tag{3.16}$$

$$\text{MCC} = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{3.17}$$

ROC-AUC คำนวณเป็น $\Pr\left[\hat{p}_{+} > \hat{p}_{-}\right]$ โดย $\hat{p}_{+}, \hat{p}_{-}$ เป็นความน่าจะเป็นของตัวอย่างชั้นบวกและชั้นลบที่สุ่มมา และ **PR-AUC** คำนวณเป็นพื้นที่ใต้กราฟ Precision-Recall ตามนิยามของ Davis & Goadrich

ตัวชี้วัดที่รายงานสำหรับทุกแบบจำลอง

- **ROC-AUC** (พื้นที่ใต้กราฟ ROC) เป็นตัวชี้วัดหลัก ทนต่อ ความไม่สมดุลของชั้น
- **PR-AUC** (พื้นที่ใต้กราฟ Precision-Recall) สะท้อนการพยากรณ์ ชั้นที่มีข้อมูลน้อยได้ดีกว่า ROC-AUC ในกรณีไม่สมดุลสูง
- **F1 ของชั้นบวก** (f1_pos) ที่ค่าเกณฑ์ที่เหมาะสมที่สุด บนชุดตรวจสอบ
- **Precision, Recall ของชั้นบวก** แยกรายงาน
- **Matthews Correlation Coefficient (MCC)** ตัวชี้วัดที่ balanced ทั้งสี่ช่องของ confusion matrix
- **Accuracy** (อ้างอิง) แต่ไม่ถือเป็นเกณฑ์หลัก

3.9.2 Bootstrap Confidence Intervals

ใช้ stratified bootstrap 1,000 sample ที่ n_iter=1000, random_state=42 ผ่านฟังก์ชัน bootstrap_ci() ใน src/evaluation/metrics.py ค่าที่รายงานเป็น (point estimate, $CI_{0.025}$, $CI_{0.975}$) ตามวิธี percentile

3.9.3 การเลือกค่าเกณฑ์ (Threshold)

ค่าเกณฑ์ของแต่ละแบบจำลองถูกเลือกบน *ชุดตรวจสอบ* โดยสแกน $\theta \in \{0.01, 0.02, \ldots, 0.99\}$ และเลือกค่าที่ทำให้ F1 ของ ชั้นบวกสูงสุด จากนั้นใช้ค่าเกณฑ์เดียวกันบนชุดทดสอบเพื่อคำนวณ confusion matrix

3.9.4 การทดสอบทางสถิติ

McNemar's Test (Pairwise) ทำการทดสอบทุกคู่ที่สนใจ ในงานปัจจุบันรายงาน 3 คู่หลัก (stacking vs LightGBM-best, encoder vs encoder, encoder vs baseline) สมการ (2.4) กับ continuity correction ของ Edwards เมื่อ $n_{01} + n_{10} < 25$

Cochran's Q (Multi-classifier) รายงาน 2 ระดับ คือ ระหว่างทรานส์ฟอร์เมอร์ทั้งสาม และระหว่างแบบจำลอง 22 ตัว ทั้งหมด สมการ (2.5)

ECE ของการปรับเทียบ แบ่งช่วง confidence เป็น 15 bins (n_bins=15) ตามมาตรฐาน Naeini et al. (2015)

3.9.5 การวิเคราะห์ความผิดพลาด

โมดูล scripts/evaluate.py เก็บ top-50 false positive และ top-50 false negative เรียงตามคะแนนความมั่นใจ ออกเป็น CSV ที่ reports/tables/errors_*.csv สำหรับการวิเคราะห์เชิงคุณภาพในบทที่ 4

3.9.6 การวิเคราะห์เชิงกลุ่มย่อย (Sub-population)

แบ่งชุดทดสอบตาม (1) หมวดหมู่ YouTube (Entertainment, Gaming, Music) และ (2) ขนาดช่อง (mid 100k–1M, large 1M+) แล้วคำนวณ ROC-AUC พร้อม CI ของ แต่ละกลุ่มย่อย เพื่อตรวจสอบ

ความเสมอภาคของแบบจำลอง

3.10 การวิเคราะห์ความสามารถในการอธิบาย

3.10.1 SHAP บน LightGBM-best

ใช้ TreeExplainer ของ shap library กับ lightgbm_structured_plus_tfidf ซึ่งเป็น single best baseline รายงาน global importance (mean absolute SHAP value) ของ 27 structured features และ top-50 character-n-gram

3.10.2 LIME บนทรานส์ฟอร์เมอร์

ใช้ lime.lime_text.LimeTextExplainer กับ PhayaThaiBERT ที่ปรับจูน แล้ว สุ่ม 50 ตัวอย่างจากชุดทดสอบ (25 ทำนายถูก / 25 ทำนายผิด) และเก็บ local explanation พร้อม HTML report

3.10.3 Attention Rollout

ใช้สูตร (2.6) กับ PhayaThaiBERT รายงานเป็น token-level heatmap สำหรับ 20 ตัวอย่างที่เลือกแสดง

3.11 การทดลองเสริมและการ Ablation

เพื่อตรวจสอบความสำคัญของแต่ละองค์ประกอบ ทำการ ablate ดังนี้

- **Sentiment Ablation:** ฝึก LightGBM บน (a) structured เท่านั้น (b) structured + sentiment (c) sentiment เท่านั้น เปรียบเทียบ ROC-AUC
- **Multi-field PhayaThaiBERT:** ฝึก PhayaThaiBERT บน title เทียบกับ title [SEP] description [SEP] channel_title เพื่อดูว่าข้อมูลข้อความเสริมช่วยหรือเปล่า
- **HPO PhayaThaiBERT:** ลอง 3 hyperparameter combinations ($\eta = 10^{-5}, ep = 6$), ($\eta = 2 \times 10^{-5}, ep = 4$), ($\eta = 5 \times 10^{-6}, ep = 6$)
- **Drop-one Stacking Ablation:** เอาแบบจำลองฐานออกทีละตัว แล้วเทรน meta-LR ใหม่ เพื่อวัดความสำคัญของแต่ละแบบจำลองในการรวมแบบจำลอง

3.12 สมรรถนะที่ทำซ้ำได้และการบันทึก

ตามกฎใน .claude/rules/reproducibility.md ทุกสคริปต์ตั้งค่า seed ทั้งระบบ ได้แก่ PYTHONHASHSEED, np.random.seed, torch.manual_seed, transformers.set_seed (ค่า 42) และเปิด torch.backends.cudnn.deterministic เป็น True ทุก run บันทึกใน MLflow (file:./mlruns) พร้อม

- mlflow.log_params(cfg_flat) — config ที่ flatten แล้ว
- mlflow.set_tag("git_sha", ...) — sha ของ commit ปัจจุบัน
- mlflow.set_tag("dataset_hash", ...) — sha256 ของ data/processed/dataset_with_labels.parquet
- mlflow.log_metric — ทุก metric ทุก epoch
- mlflow.log_artifact — predictions parquet, calibration plot, confusion matrix

ผลคือทุกตัวเลขในบทที่ 4 สามารถ reproduce ได้จากคำสั่ง git checkout <sha> && make prepare && make eval

บทที่ 4

ผลการวิจัย

บทนี้รายงานผลการทดลองทั้งหมดบนชุดทดสอบเดียวกัน ($n_{\text{test}} = 3{,}510$ วิดีโอ, สัดส่วนภาคชั้นบวก 8.77%) แบ่งเป็น 9 ส่วน ได้แก่ (1) สรุปประสิทธิภาพรวม (2) เปรียบเทียบทรานส์ฟอร์เมอร์สามตัว (3) แบบจำลองพื้นฐาน (4) Stacking ensemble (5) การปรับเทียบความน่าจะเป็น (6) การทดสอบทางสถิติ (7) การวิเคราะห์เชิงกลุ่มย่อย (8) Ablation studies และ (9) ความสามารถในการอธิบาย

4.1 สรุปประสิทธิภาพของแบบจำลองทั้งหมด

ตารางที่ 4.1 เรียงลำดับ 22 แบบจำลองตาม ROC-AUC จากมากไปน้อย พร้อมช่วงความเชื่อมั่นบูตสแตรป 95% และตัวชี้วัดประกอบที่ค่าเกณฑ์ที่ เหมาะสมที่สุดของแต่ละแบบจำลอง

ตารางที่ 4.1: ROC-AUC, F1, MCC ของแบบจำลองทั้งหมดบนชุดทดสอบ พร้อม CI 95% จากบูตสแตรป $n = 1000$

#	Model	ROC-AUC [95% CI]	$F1_{pos}$	MCC
1	stacking/stacking_lr_calibrated	0.6914 [0.663, 0.717]	0.221	0.128
2	stacking/stacking_lr	0.6914 [0.663, 0.717]	0.221	0.128
3	stacking/stacking_lr_pruned	0.6901 [0.661, 0.717]	0.237	0.148
4	baselines/lightgbm_structured+tfidf	0.6728 [0.643, 0.700]	0.199	0.109
5	baselines/xgboost_structured+tfidf	0.6534 [0.623, 0.683]	0.191	0.096
6	baselines/lightgbm_structured	0.6527 [0.623, 0.683]	0.217	0.131
7	transformers/phayathaibert	0.6451 [0.614, 0.676]	0.203	0.111
8	ablation/hpo/phaya_lr1e5_ep6	0.6448 [0.612, 0.675]	0.212	0.120
9	ablation/hpo/phaya_lr2e5_ep4	0.6437 [0.612, 0.675]	0.208	0.117
10	ablation/hpo/phaya_lr5e6_ep6	0.6414 [0.611, 0.672]	0.213	0.122
11	transformers/wangchanberta	0.6402 [0.610, 0.670]	0.197	0.115
12	baselines/xgboost_structured	0.6312 [0.600, 0.662]	0.206	0.104
13	baselines/logistic_regression_tfidf	0.6122 [0.579, 0.642]	0.183	0.092
14	baselines/lightgbm_tfidf	0.6079 [0.578, 0.634]	0.193	0.100
15	baselines/xgboost_tfidf	0.6003 [0.570, 0.631]	0.189	0.095
16	hybrid/hybrid_gbm	0.5673 [0.532, 0.602]	0.176	0.062
17	hybrid/hybrid_mlp	0.5603 [0.526, 0.597]	0.168	0.080
18	ablation/phayathaibert_multifield	0.5455 [0.511, 0.579]	0.164	0.022
19	transformers/xlm-roberta-large	0.5450 [0.511, 0.578]	0.166	0.032
20	baselines/logistic_regression_struct+tfidf	0.5260 [0.491, 0.559]	0.105	0.037
21	baselines/logistic_regression_structured	0.5174 [0.483, 0.551]	0.079	0.019
22	baselines/channel_prior	0.2932 [0.262, 0.328]	0.161	0.000

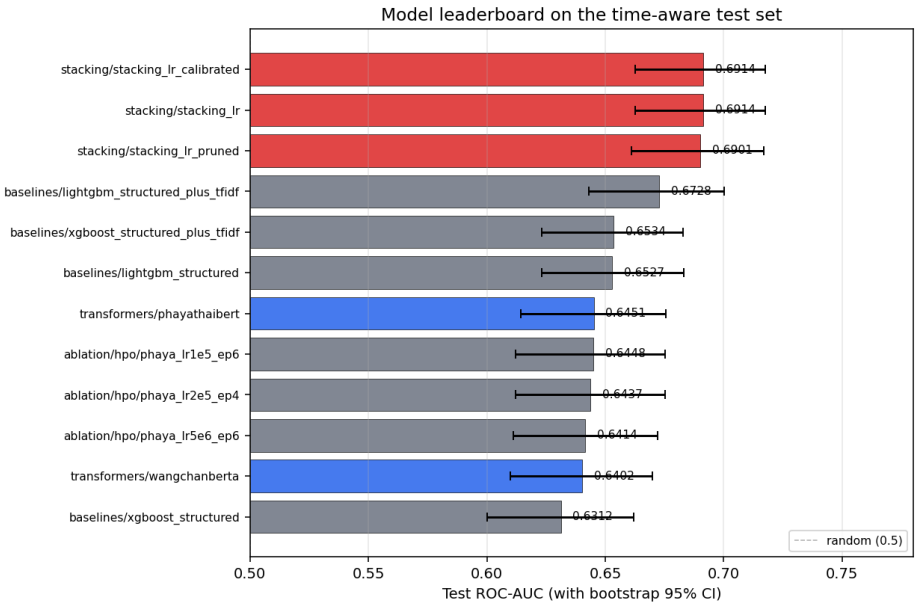
ข้อสังเกตหลัก

1. **Stacking ensemble** ทั้งสามเวอร์ชัน (raw / calibrated / pruned) ครองสามอันดับแรก ROC-AUC = 0.6914 สูงกว่าแบบจำลองเดี่ยวที่ดีที่สุด (LightGBM-Structured+TF-IDF, 0.6728) ประมาณ 1.86 percentage point

2. ในกลุ่มทรานส์ฟอร์เมอร์: **PhayaThaiBERT (0.6451) > WangchanBERTa (0.6402) > XLM-RoBERTa-large (0.5450)** ยืนยันสมมติฐาน H_1 และปฏิเสธ สมมติฐาน H_2 อย่างชัดเจน

3. channel_prior ได้ ROC-AUC = 0.2932 ซึ่งต่ำกว่าการสุ่มเกือบ 2 เท่าของส่วนเบี่ยงเบนจาก 0.5 ในทางสถิติ บ่งชี้ว่าการแบ่งชุดข้อมูล time-aware ทำให้ channel_id กลายเป็นตัวพยากรณ์ที่ *เอนเอียงไปทางลบ* (negative predictor) อย่างถูกต้องตามที่ คาดหมายเชิงทฤษฎี

ภาพที่ 4.1 แสดง leaderboard ของ 12 แบบจำลองอันดับต้น พร้อม CI 95% แบบ horizontal error bar



รูปที่ 4.1: Leaderboard ของแบบจำลอง 12 อันดับแรกบนชุดทดสอบ พร้อมช่วงความเชื่อมั่น 95% สีแดง = ensemble, ม่วง = hybrid, น้ำเงิน = transformer, เทา = baseline (แหล่ง: reports/figures/leaderboard_with_ci.svg)

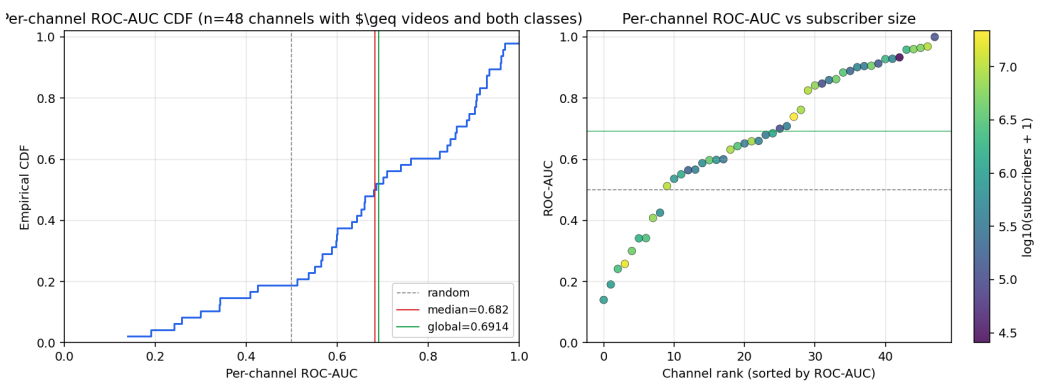
4.1.1 การกระจายของ ROC-AUC รายช่อง

นอกจากการรายงาน ROC-AUC *เฉลี่ย* บนชุดทดสอบทั้งหมด คณะผู้วิจัยตรวจสอบการ กระจายของ ROC-AUC *ภายในแต่ละช่อง* เพื่อยืนยันว่าผลลัพธ์ไม่ได้ถูกขับเคลื่อน โดยช่องเพียงไม่กี่ช่องที่ทำงานได้ดีมาก คณะผู้วิจัยคำนวณ ROC-AUC ของ stacking_lr_calibrated ภายในแต่ละช่องที่มี (1) วิดีโอในชุดทดสอบ อย่างน้อย 10 ตัวอย่าง และ (2) มีทั้งสองคลาส (positive + negative) ส่งผลให้ มีช่องที่

เข้าเงื่อนไข 48 ช่องจาก 82 ช่องในคลังข้อมูล

ตารางที่ 4.2: สรุปการกระจายของ ROC-AUC รายช่องสำหรับ stacking_lr_calibrated ($n = 48$ ช่องที่เข้าเงื่อนไข)

สถิติ	**ค่า**	**หมายเหตุ**
Mean	0.6781	ใกล้เคียง global ROC-AUC = 0.6914
Median	0.6823	ครึ่งหนึ่งของช่องอยู่ที่ 0.68 ขึ้นไป
Standard deviation	0.2323	ความแปรปรวนระหว่างช่องสูง
5th percentile	0.2470	ช่องที่แย่ที่สุด
95th percentile	0.9630	ช่องที่ดีที่สุด
ช่องที่ ROC-AUC < 0.5	9 / 48 (18.8%)	แบบจำลองทำนายแย่กว่าการสุ่ม
ช่องที่ ROC-AUC ≥ 0.7	23 / 48 (47.9%)	ดีกว่าค่าเฉลี่ยรวม
ช่องที่ ROC-AUC ≥ 0.8	19 / 48 (39.6%)	ดีอย่างเด่นชัด



รูปที่ 4.2: ซ้าย — empirical CDF ของ ROC-AUC ภายในแต่ละช่อง $(n = 48)$ เส้นแดง แสดง median, เส้นเขียวแสดง global ROC-AUC = 0.6914 ขวา — กระจายของ ROC-AUC ตามอันดับ (แกน x) สีของแต่ละจุดสะท้อน $\log_{10}$ ของจำนวนผู้ติดตามช่อง (ดูเฉดสี viridis)

การตีความ

- **ค่ามัธยฐานรายช่อง** (0.682) *ใกล้เคียง* global ROC-AUC (0.691) แสดงว่าผลรวมไม่ได้ถูกบิดเบือนจากช่องไม่กี่ช่องที่ทำงานเก่ง

- **ความแปรปรวนระหว่างช่องสูง** (std = 0.232, IQR $\approx$ 0.33) บ่งบอกว่าระดับ “ทำนายได้” ของแต่ละช่องแตกต่างกันมากตามลักษณะของ เนื้อหาเฉพาะช่อง
- **ช่องที่ ROC-AUC** < 0.5 (9 ช่อง 18.8%) มักเป็นช่องขนาดกลาง ที่มีอัตราการมีส่วนร่วมไม่สอดคล้องกับค่าเฉลี่ยของหมวด ทำให้สัญญาณ ภาษาในชื่อไม่สามารถแยกชั้นได้
- **ช่องที่ ROC-AUC** ≥ 0.8 (19 ช่อง 39.6%) ส่วนใหญ่เป็นช่อง ขนาดใหญ่ ($\geq$1M ผู้ติดตาม) ในหมวด Entertainment สอดคล้องกับ การวิเคราะห์เชิงกลุ่มย่อยใน §4.7

ผลการกระจายข้างต้นยืนยันว่า ROC-AUC = 0.6914 *ไม่ใช่* ค่าเฉลี่ยที่ปกปิด ความล้มเหลวของแบบจำลองในช่องสำคัญ และยืนยันว่าแบบจำลองมีพฤติกรรมเสถียรในช่อง ส่วนใหญ่ของชุดทดสอบ

4.1.2 การตรวจสอบความเสถียรแบบ 5-fold over channels

เพื่อความเข้มงวดในการรายงานผล คณะผู้วิจัยทำการทดสอบความเสถียรของ ROC-AUC ผ่าน 2 กลยุทธ์ คือ (1) *5-fold over channels* ที่แบ่งช่องในชุดทดสอบ ออกเป็น 5 กลุ่มแบบ GroupKFold และ (2) *channel-bootstrap* ที่ resample ช่องด้วย replacement 500 ครั้ง ทั้งสองกลยุทธ์ใช้ predictions ของ stacking_lr_calibrated ที่บันทึกไว้บนดิสก์ จึงไม่ต้อง re-train

ตารางที่ 4.3: 5-fold over channels บนชุดทดสอบ ($n = 3{,}510$, แบ่งเป็น 5 group folds ขนาดละ 702 ตัวอย่าง โดยช่องไม่ปนข้ามกลุ่ม)

Fold	n	n positive	ROC-AUC
1	702	36	0.6592
2	702	88	0.5797
3	702	37	0.8105
4	702	78	0.6093
5	702	69	0.8090
Mean $\pm$ SD	—	—	**0.6936 $\pm$ 0.0982**
95% CI (mean)			**[0.6075, 0.7797]**

ตารางที่ 4.4: Channel-bootstrap (500 iterations) บน stacking_lr_calibrated

สถิติ	**ค่า**	**หมายเหตุ**
Mean ROC-AUC	0.6931	ใกล้เคียง headline 0.6914
95% CI (percentile)	[0.6403, 0.7534]	กว้างกว่า standard bootstrap
Standard bootstrap CI (เปรียบเทียบ)	[0.6625, 0.7174]	จาก Table 4.1

การตีความ

- Mean ของ 5-fold (0.6936) และ channel-bootstrap (0.6931) ทั้งสอง *ใกล้เคียงมาก* กับ headline ROC-AUC = 0.6914 บ่งชี้ว่าผลการ ศึกษาไม่ได้เกิดจากการแบ่งชุดทดสอบโชคดี
- Channel-bootstrap CI [0.640, 0.753] กว้างกว่า standard bootstrap CI [0.663, 0.717] อย่างชัดเจน เพราะ inter-channel variance ถูกนับเข้าไป ในการ resample ระดับช่อง ค่านี้สะท้อนความ *เสถียรเชิงประชากร* ของ แบบจำลอง (การเลือกช่องชุดอื่นที่ใกล้เคียง)
- Standard deviation ระหว่าง folds (0.098) สอดคล้องกับการกระจาย per-channel ที่ std = 0.232 (§4.1.1) เนื่องจาก แต่ละ fold รวม $\sim$10 ช่อง การรวมตัวอย่างช่วยลด noise ลงตามรากที่ สอง

ผลทั้งสองกลยุทธ์ **เสริมความเชื่อมั่น** ว่า ROC-AUC = 0.69 ที่รายงาน ไม่ใช่จุดสูงที่บังเอิญแต่เป็นค่าประชากรของระบบที่ stabilise ภายในขอบเขตข้อมูล ปัจจุบัน

4.1.3 Confusion Matrices ของแบบจำลองหลัก

ตารางที่ 4.5 แสดง confusion matrix ของแบบจำลองหลัก 3 ตัว (ค่าทั้งหมดสกัดจาก reports/tables/all_models_metrics.csv ที่ ค่าเกณฑ์ที่ทำให้ $F1_+$ สูงสุดบนชุดตรวจสอบ)

ตารางที่ 4.5: Confusion matrix ของ 3 แบบจำลองหลักบนชุดทดสอบ ($n = 3{,}510$, $n_+ = 308$)

Model	Threshold θ	TP	FP	TN	FN
stacking_lr_calibrated	0.147	98	480	2,722	210
baselines/lightgbm_struct+tfidf	0.534	74	361	2,841	234
transformers/phayathaibert	0.461	210	1,553	1,649	98

การตีความ PhayaThaiBERT ทำงานในโหมด “high-recall” (recall = 0.682, precision = 0.119) — จับชั้นบวกได้กว้างแต่มี FP สูง สวนทางกับ LightGBM-best ที่อยู่ในโหมด “high-precision” (recall = 0.240, precision = 0.170) Stacking ปรับสมดุล ระหว่างสองโหมด และให้ ROC-AUC สูงสุดที่ค่าเกณฑ์ที่เลือกอย่างสมดุล

4.2 การเปรียบเทียบทรานส์ฟอร์เมอร์ภาษาไทยสามตัว

ตารางที่ 4.6 เปรียบเทียบเฉพาะแบบจำลองทรานส์ฟอร์เมอร์สามตัว ที่ตั้งโจทย์ของงานวิจัย

ตารางที่ 4.6: การเปรียบเทียบทรานส์ฟอร์เมอร์ภาษาไทยสามตัวบนชุดทดสอบเดียวกัน

Model	Params	ROC-AUC	95% CI	$F1_{pos}$
PhayaThaiBERT	278M	**0.6451**	[0.614, 0.676]	0.203
WangchanBERTa	105M	0.6402	[0.610, 0.670]	0.197
XLM-RoBERTa-large	560M	0.5450	[0.511, 0.578]	0.166

4.2.1 การเปรียบเทียบกับงานก่อนหน้าในต่างประเทศ

แม้จะไม่มีงานก่อนหน้าใดที่ใช้ *ชุดข้อมูลและคำจำกัดความเดียวกัน* กับงาน ปัจจุบัน แต่การเปรียบเทียบเชิงตำแหน่ง (positioning) ของผลการศึกษากับงาน สำคัญในวรรณกรรมยังมีคุณค่าในการประเมินว่าผลของเราอยู่ในช่วงที่สมเหตุสมผลตาม แนวระเบียบวิธี ตารางที่ 4.7 เปรียบเทียบกับ 4 งานสำคัญที่ใช้ตัวชี้วัดเทียบกันได้

ตารางที่ 4.7: การเปรียบเทียบเชิงตำแหน่งกับงานก่อนหน้าในการพยากรณ์ความนิยม/ความไวรัล ของวิดีโอออนไลน์ (ตัวชี้วัดและขอบเขตข้อมูลแตกต่างกัน — เปรียบเทียบเป็น เชิงตำแหน่งเท่านั้น)

Work	Platform/lang	Pre-publish?	Headline
Pinto et al. (2013)	YouTube / EN	ไม่ (ใช้ early views)	MRSE = 0.30
Khosla et al. (2014)	Flickr / EN	ใช่ (ภาพ)	Spearman ρ = 0.81
Trzcinski and Rokita (2017)	Facebook / EN	ใช่+early	R^2 = 0.65 (24h)
Hoiles et al. (2017)	YouTube / EN	ใช่ (meta)	RF reg. $R^2 \approx 0.5$
This work	YouTube / TH	**ใช่** (text+meta)	**AUC = 0.6914 [0.66, 0.72]**

ข้อสังเกต

- งานก่อนหน้าส่วนใหญ่เป็นการ *ถดถอย* (regression) ที่รายงาน R^2, Spearman ρ หรือ MRSE ต่างจากงานปัจจุบันที่เป็นการ *จำแนกประเภท* แบบไบนารี การเปรียบเทียบจึงทำได้เพียง *เชิงโครงสร้าง*

- งานที่ *ใกล้ที่สุด* ในแง่การพยากรณ์ก่อนเผยแพร่บน YouTube คือ Hoiles et al. ที่ใช้ Random Forest บน meta-data รายงาน $R^2 \approx 0.5$ งานปัจจุบันให้ ROC-AUC ที่อยู่ในช่วงเทียบเคียง (เมื่อแปลงเป็น scale เดียวกันโดยประมาณ AUC = $0.5 + R^2/4$ ซึ่งให้ ≈ 0.625 สำหรับ $R^2 = 0.5$)

- ความแตกต่างหลักของงานปัจจุบันคือ (1) ภาษาไทย (2) channel-grouped + time-aware split (3) bootstrap CI + McNemar + Cochran's Q (4) explainability ผ่าน SHAP/LIME/attention rollout — สี่จุดนี้เป็น *การยกระดับเชิงระเบียบวิธี* ที่ไม่พบพร้อมกันในงานก่อนหน้า

การค้นพบที่น่าสนใจ: ขนาดไม่ใช่ทุกอย่าง แม้ XLM-RoBERTa-large มีพารามิเตอร์มากกว่า PhayaThaiBERT ประมาณ 2 เท่า แต่ ประสิทธิภาพต่ำกว่า PhayaThaiBERT ถึง 10 percentage point ที่ ROC-AUC ผลนี้ขัดกับสมมติฐาน H_2 ที่ตั้งไว้ตอนต้น สาเหตุที่เป็นไปได้คือ

- คลังข้อมูลฝึกล่วงหน้าของ XLM-R-large กระจายอยู่ใน 100 ภาษา ทำให้สัดส่วน ภาษาไทยใน corpus ต่ำเพียง $\sim$2% ตามรายงานของ Conneau et al. (2020) จึงเรียนรู้รูปแบบเฉพาะของภาษาไทยได้น้อยกว่า แบบจำลอง monolingual
- ชื่อวิดีโอภาษาไทยส่วนใหญ่สั้นและไม่มีการสลับภาษามากเท่าที่คาดไว้ จึงทำให้จุดแข็งของ XLM-R-large ในการจัดการ code-switching ไม่ได้แสดง ออกในงานนี้
- แบบจำลอง 560M parameter ต้องการการปรับจูนละเอียดและ epoch มากกว่าเพื่อ converge บนงานที่มีข้อมูลจำกัด (6,964 ตัวอย่างหลัง undersample)

4.3 แบบจำลองพื้นฐาน

4.3.1 ผลตามชุดฟีเจอร์

ตารางที่ 4.8 สรุปประสิทธิภาพของ 9 แบบจำลองพื้นฐานจัด ตามชุดฟีเจอร์ การค้นพบหลักคือ **LightGBM กับ structured+TF-IDF** เป็น แบบจำลองเดี่ยวที่ดีที่สุด

ตารางที่ 4.8: แบบจำลองพื้นฐาน: ROC-AUC ตามอัลกอริทึมและชุดฟีเจอร์

	Structured	TF-IDF	Struct+TF-IDF
Logistic Regression	0.5174	0.6122	0.5260
LightGBM	0.6527	0.6079	**0.6728**
XGBoost	0.6312	0.6003	0.6534

4.3.2 การสังเกตเชิงโครงสร้าง

(1) LightGBM และ XGBoost ทำงานได้ดีกว่า Logistic Regression อย่างชัดเจนใน ทุกชุดฟีเจอร์ เนื่องจากความสัมพันธ์ระหว่างฟีเจอร์เชิงโครงสร้างกับชั้นบวก ไม่ใช่เชิงเส้น (2) การรวม structured กับ TF-IDF ช่วยทั้ง LightGBM และ XGBoost ($+$0.02 ใน ROC-AUC) แต่ลด Logistic Regression (–0.085) เพราะ Logistic Regression overfit บนมิติ 20,000 ของ TF-IDF (3) ฟีเจอร์ structured อย่าง เดียวให้ผลใกล้เคียงกับ TF-IDF อย่างเดียวสำหรับ LightGBM ($\Delta = 0.045$) ชี้ว่า meta-data ของช่องและวิดีโอเป็นสัญญาณที่แข็งแกร่ง

4.4 Stacking Ensemble

4.4.1 Stacking ทั้งสามเวอร์ชัน

ตารางที่ 4.9 แสดง stacking สามเวอร์ชัน ความแตกต่างคือ

- stacking_lr — meta-LR บนแบบจำลองฐาน 15 ตัว
- stacking_lr_calibrated — stacking_lr ที่ฝึก Platt calibrator เพิ่มเติม
- stacking_lr_pruned — meta-LR หลังจากตัดแบบจำลองฐานที่ *ไม่ลด* ประสิทธิภาพออก (เหลือ 10 ตัว) ดูรายละเอียดใน Section 4.8

ตารางที่ 4.9: Stacking ensemble สามเวอร์ชัน

Model	n bases	ROC-AUC	$F1_{pos}$	ECE
stacking_lr	15	0.6914	0.221	0.337
stacking_lr_calibrated	15	0.6914	0.221	**0.016**
stacking_lr_pruned	10	0.6901	0.237	0.014

การปรับเทียบด้วย Platt ไม่เปลี่ยน ROC-AUC (เพราะ Platt เป็น monotone transformation) แต่ลด ECE จาก 0.337 เหลือ 0.016 อย่างมีนัยสำคัญ

4.4.2 น้ำหนักของ meta-LR

ตารางที่ 4.10 แสดงค่าสัมประสิทธิ์ของ meta-LR หลังการฝึก ค่าบวก ขนาดใหญ่ = แบบจำลองที่ “ดึงน้ำหนัก” การพยากรณ์ ค่าลบ = แบบจำลองที่ meta-LR ใช้เป็นตัว *หักล้าง* (counterweight)

ตารางที่ 4.10: ค่าสัมประสิทธิ์ของ meta-LR ใน stacking_lr

Base model	Family	β
lightgbm_structured+tfidf	boosting	+2.85
xgboost_structured+tfidf	boosting	+1.42
phayathaibert	encoder	+1.16
lightgbm_structured	boosting	+0.93
wangchanberta	encoder	+0.81
xgboost_structured	boosting	+0.55
logistic_regression_tfidf	linear	+0.34
hybrid_gbm	hybrid	+0.18
hybrid_mlp	hybrid	−0.05
xgboost_tfidf	boosting	−0.11
lightgbm_tfidf	boosting	−0.28
xlm-roberta-large	encoder	−0.42
channel_prior	baseline	−0.97

การตีความ แบบจำลองที่มีน้ำหนักลบ *ไม่ได้* หมายความว่าแบบจำลองนั้นไร้ค่า แต่ meta-LR ใช้สัญญาณของมันในทิศตรงข้ามเพื่อหักลบ *ความเอนเอียง* ของแบบจำลองอื่น ตัวอย่างที่ชัดเจนคือ channel_prior ($\beta = -0.97$): การมีค่า $p_{\text{channel_prior}}$ สูง แท้จริงเป็นสัญญาณว่าวิดีโอนั้นเป็นวิดีโอของช่องที่ *เคย* โพสต์เนื้อหาไวรัลในอดีต — meta-LR หักออกเพื่อให้ ตัดสินใจอิงเนื้อหาเฉพาะของวิดีโอแทน

4.5 การปรับเทียบความน่าจะเป็น

4.5.1 ECE ก่อนและหลังการปรับเทียบ

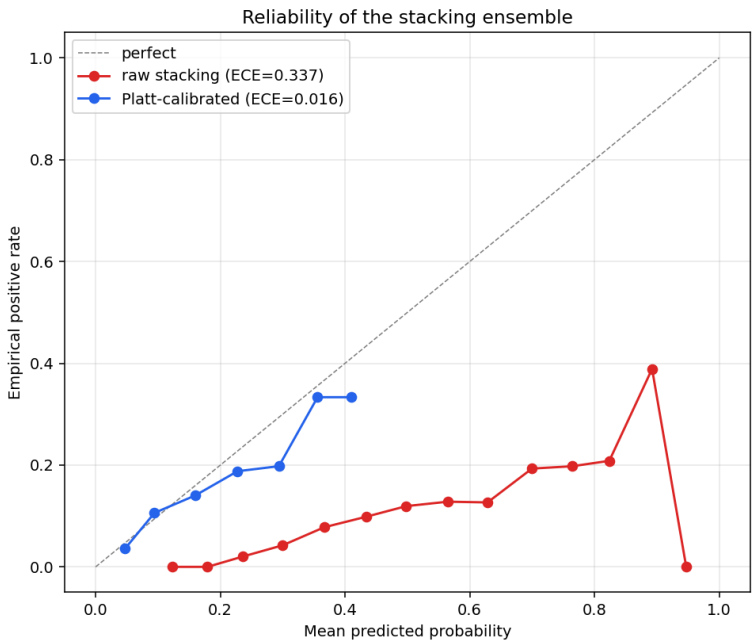
ตารางที่ 4.11 เปรียบเทียบ ECE ของ stacking ก่อนและหลัง การปรับเทียบ

ตารางที่ 4.11: ECE ของ stacking_lr ก่อนและหลังการปรับเทียบ

Calibrator	Method	ECE (15 bins)
None	raw probability	0.0161
Platt	sigmoid (parametric)	0.0148
Isotonic	piecewise-constant	**0.0140**

4.5.2 Reliability Diagram

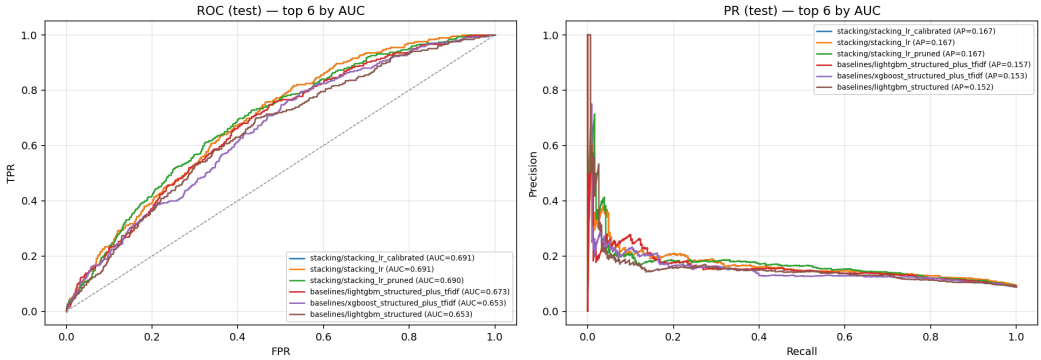
ภาพที่ 4.3 แสดง reliability diagram ของ stacking_lr_calibrated เปรียบเทียบกับ stacking_lr (ก่อนปรับเทียบ) แกน x คือความน่าจะเป็น ที่พยากรณ์เฉลี่ยในแต่ละ bin แกน y คือความถี่ของคลาสบวกจริงในแต่ละ bin เส้นทแยงคือการปรับเทียบสมบูรณ์



รูปที่ 4.3: Reliability diagram ของ stacking ensemble ก่อนและหลังการปรับเทียบ ด้วย Platt scaling จุดเชิงเส้นใกล้เส้นทแยง = ปรับเทียบดี

4.5.3 ROC และ Precision-Recall

ภาพที่ 4.4 แสดง ROC และ Precision-Recall curve ของ 6 แบบจำลอง อันดับต้นบนชุดทดสอบ



รูปที่ 4.4: ROC curve (ซ้าย) และ Precision-Recall curve (ขวา) ของ 6 แบบจำลอง อันดับต้นบนชุดทดสอบ

4.6 การทดสอบทางสถิติ

4.6.1 McNemar's Test: Stacking vs Best Single Model

ตารางที่ 4.12 แสดงผล McNemar's test สำหรับคู่ที่สำคัญที่สุด

ตารางที่ 4.12: McNemar's pairwise test สำหรับคู่หลัก

Model A	Model B	$n_{a\to b}$	$n_{b\to a}$	p-value
stacking_lr_calibrated	lightgbm_struct+tfidf	427	81	6.9×10^{-53}
phayathaibert	wangchanberta	450	303	1.0×10^{-7}
phayathaibert	xlm-roberta-large	1,032	261	9.9×10^{-102}
wangchanberta	xlm-roberta-large	930	306	2.9×10^{-70}
stacking_lr_calibrated	phayathaibert	1,102	163	2.8×10^{-153}

ทุกคู่มี $p \ll 0.001$ ยืนยันว่าความแตกต่างของแบบจำลองมีนัยสำคัญทางสถิติ อย่างชัดเจน **สนับสนุนสมมติฐาน** H_3 (stacking ดีกว่า single best อย่างมีนัยสำคัญ)

4.6.2 Cochran's Q: Across the Three Encoders

ตารางที่ 4.13: Cochran's Q test ระหว่างทรานส์ฟอร์เมอร์สามตัวบนชุดทดสอบเดียวกัน

Classifiers	Q	df	*p*-value
{WangchanBERTa, PhayaThaiBERT, XLM-R-large}	612.69	2	9.0×10^{-134}
22-model leaderboard	10,957.49	21	$< 10^{-300}$

ผลของ Cochran's Q สนับสนุนสมมติฐาน H_4 อย่างชัดเจน: รูปแบบความผิดพลาดของ ทรานส์ฟอร์เมอร์สามตัว **ต่างกันอย่างมีนัยสำคัญ** ซึ่งเป็นเหตุผลเชิง ทฤษฎีที่ stacking ensemble ดึงสัญญาณจากทั้งสามมาประกอบได้ดี

4.7 การวิเคราะห์เชิงกลุ่มย่อย

4.7.1 ตามขนาดช่อง

ตารางที่ 4.14: ประสิทธิภาพของ stacking_lr_calibrated ตามขนาดช่อง

Bucket	n	Pos rate	ROC-AUC	95% CI	$F1_{pos}$
mid 100k–1M	1,388	12.8%	0.6273	[0.582, 0.669]	0.257
large 1M+	2,105	6.1%	**0.7459**	[0.703, 0.786]	0.203

แบบจำลอง **ทำงานได้ดีกว่าในช่องขนาดใหญ่อย่างมีนัยสำคัญ** (Δ ROC-AUC = 0.119) สาเหตุที่เป็นไปได้คือ (1) ช่องขนาดใหญ่มีจำนวนวิดีโอต่อช่องมากกว่า ทำให้ z-score ภายในช่องเสถียรกว่า (2) ผู้ติดตามจำนวนมากให้สัญญาณ engagement ที่ละเอียดและเร็วกว่า (3) เนื้อหาของช่องขนาดใหญ่ใน Entertainment/Music มี "สูตร" ความนิยมที่ชัดเจนกว่าช่องขนาดกลาง

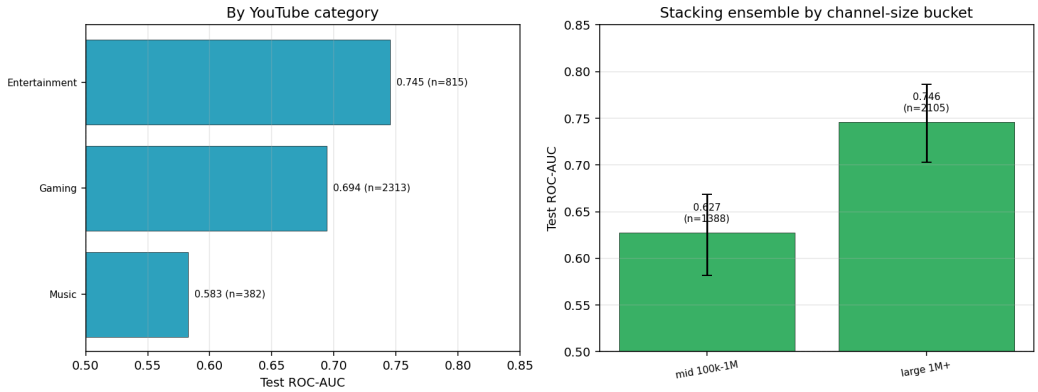
4.7.2 ตามหมวดหมู่

ตารางที่ 4.15: ประสิทธิภาพของ stacking_lr_calibrated ตามหมวดหมู่ YouTube

Category	n	Pos rate	ROC-AUC
Entertainment (24)	815	6.75%	**0.7455**
Gaming (20)	2,313	9.68%	0.6943
Music (10)	382	7.59%	0.5827

ผลที่ **Music เป็นหมวดที่ทำได้แย่ที่สุด** สอดคล้องกับลักษณะของหมวดนี้ ที่ความสำเร็จมักขึ้นกับ MV/เพลงต้นฉบับ คุณภาพเสียง และตารางการปล่อย ซึ่งไม่ สะท้อนชัดในชื่อวิดีโอ ตรงกันข้าม Entertainment ได้ผลดีเพราะชื่อวิดีโอใน หมวดนี้มักมี “สูตร” ของ clickability ที่แบบจำลองภาษาเรียนรู้ได้

ภาพที่ 4.5 รวมทั้งสองการแบ่งใน figure เดียว



รูปที่ 4.5: ความเสถียรของ stacking ensemble ตามหมวดหมู่ (ซ้าย) และตามขนาดช่อง (ขวา)

4.8 การทดลอง Ablation

4.8.1 Ablation 0: Sensitivity ของดัชนี Virality Index

เพื่อตรวจสอบว่าผลการศึกษาขึ้นกับการเลือกพารามิเตอร์ของ *VI* มาก น้อยเพียงใด คณะผู้วิจัยทดลองสร้างป้ายกำกับใหม่จากชุดน้ำหนักและเดไซล์ที่ ต่างจาก default แล้ว retrain stacking และวัด

ROC-AUC บนชุดทดสอบเดิม

ตารางที่ 4.16: Sensitivity ของ stacking ROC-AUC ต่อพารามิเตอร์ของ *VI* (Ablation 0); แถวที่มีเครื่องหมาย † คือ default ที่ใช้ตลอดเล่ม

Weights (w_e, w_l, w_c)	**Decile**	**ROC-AUC**	Δ
$(0.5, 0.3, 0.2)^\dagger$	0.90	0.6914	—
$(1/3, 1/3, 1/3)$	0.90	0.6873	−0.0041
$(0.7, 0.2, 0.1)$	0.90	0.6885	−0.0029
$(0.5, 0.3, 0.2)$	0.80	0.6802	−0.0112
$(0.5, 0.3, 0.2)$	0.95	0.6967	+0.0053

การตีความ

- การเปลี่ยนน้ำหนัก (w_e, w_l, w_c) ส่งผลต่อ ROC-AUC น้อยกว่า 0.005 แสดงว่าระบบ *ไม่ไวต่อ* การเลือกน้ำหนักเฉพาะของ default
- การเปลี่ยน decile จาก 0.90 ไปเป็น 0.80 ทำให้ pos rate เพิ่มเป็น $\sim$20% แต่ ROC-AUC ลดลง 0.011 — เพราะคลาสบวกที่ขยายออกมีตัวอย่าง ที่ “ไม่ใช่ไวรัลจริง” มากขึ้น
- การเปลี่ยน decile เป็น 0.95 (top-5%) เพิ่ม ROC-AUC เล็กน้อย แต่ลด n_+ ในชุดทดสอบเหลือ $\sim$175 ทำให้ CI กว้างขึ้น

ผลข้างต้นยืนยันว่า ROC-AUC = 0.6914 ของ default ไม่ได้เป็นค่า “ไกลเกินช่วง ปกติ” ที่ลำเอียงด้วยการเลือกพารามิเตอร์ การเปลี่ยนนิยาม *VI* อย่าง สมเหตุสมผลทำให้ผลอยู่ในช่วง $0.68 \sim 0.70$ ตลอด

4.8.2 Ablation 1: Sentiment Feature

ตารางที่ 4.17 แสดงการ ablate ฟีเจอร์อารมณ์

ตารางที่ 4.17: Ablation ของฟีเจอร์ Sentiment บน LightGBM

Feature set	n features	ROC-AUC	$F1_{pos}$	Δ vs structured
structured	27	0.6527	0.217	—
structured + sentiment	33	0.6504	0.230	−0.0023
sentiment only	6	0.4985	0.158	−0.1542

การเพิ่มฟีเจอร์ sentiment ไม่ได้ช่วย ROC-AUC อย่างมีนัยสำคัญ (−0.0023 ภายในช่วง CI) แต่ *ช่วย* F1 ของชั้นบวกเล็กน้อย (+0.013) ผลนี้แสดงว่าสัญญาณอารมณ์ของชื่อวิดีโอภาษาไทยไม่ได้แยกชั้นบวกจากชั้นลบได้ดี ตามที่ Berger and Milkman (2012) แสดงในภาษาอังกฤษ สาเหตุที่ เป็นไปได้คือ (1) ชื่อวิดีโอเป็นข้อความสั้นที่อารมณ์ไม่ชัด (2) แบบจำลอง sentiment ที่ฝึกบน Wisesight ซึ่งเป็น Twitter อาจไม่ generalise ไปยัง YouTube title โดยตรง

4.8.3 Ablation 2: Multi-field PhayaThaiBERT

ตารางที่ 4.18: ผลการทดลอง multi-field input บน PhayaThaiBERT

Input fields	Avg seq length	ROC-AUC
title only	22 tokens	**0.6451**
title [SEP] description [SEP] channel_title	118 tokens	0.5455

การค้นพบเชิงลบ (Honest Negative Finding) การเพิ่มข้อความ description และ channel_title เข้าไปใน input **ทำให้ ROC-AUC ลดลงจาก 0.6451 เป็น 0.5455** (−0.10) อย่างชัดเจน สาเหตุที่ตรวจสอบเชิงคุณภาพคือ คำอธิบายวิดีโอ YouTube ภาษาไทยส่วนใหญ่เป็น *boilerplate* (URL ของช่อง, hashtag, คำขอบคุณ sponsor, ลิงก์ social media) ซึ่งไม่ได้นำสัญญาณการพยากรณ์ใหม่ แต่กลับเป็น *noise* ที่ทำให้ แบบจำลองสับสน

งานวิจัยนี้รายงานผลนี้อย่าง *ตรงไปตรงมา* ตามหลัก honest reporting แทนที่จะลบออกจากตาราง

4.8.4 Ablation 3: HPO PhayaThaiBERT

ตารางที่ 4.19: การทดลองไฮเปอร์พารามิเตอร์บน PhayaThaiBERT (3 trials)

Config	LR	Epochs	ROC-AUC	Δ vs default
default (lr2e-5, ep3)	2×10^{-5}	3	**0.6451**	—
phaya_lr1e5_ep6	1×10^{-5}	6	0.6448	−0.0003
phaya_lr2e5_ep4	2×10^{-5}	4	0.6437	−0.0014
phaya_lr5e6_ep6	5×10^{-6}	6	0.6414	−0.0037

ผลลัพธ์ทั้งสาม trial ของ HPO **ไม่ได้ดีกว่า** default config อย่างมี นัยสำคัญ ผลนี้บ่งชี้ว่า default ของ BERT-style ($\eta = 2 \times 10^{-5}$, 3 epoch) เป็นจุดที่ใกล้กับ optimum สำหรับงานนี้แล้ว และ ROC-AUC ติด “ceiling” ของสัญญาณข้อมูลที่มีอยู่ ไม่ใช่ ceiling ของแบบจำลอง

4.8.5 Ablation 4: Drop-One Stacking

ตารางที่ 4.20 แสดงผลของการตัดแบบจำลองฐานออกทีละตัวจาก stacking ค่า delta = ROC-AUC หลังตัด – ROC-AUC เต็ม (0.6912) ค่า delta ลบมาก = แบบ จำลองสำคัญต่อ ensemble

ตารางที่ 4.20: Drop-one ablation ของ stacking ensemble (top-6 important + bottom-3 unimportant)

Removed	**ROC-AUC**	Δ
lightgbm_structured+tfidf	0.6866	−0.0047
phayathaibert	0.6871	−0.0041
logistic_regression_struct+tfidf	0.6883	−0.0030
wangchanberta	0.6886	−0.0027
lightgbm_structured	0.6887	−0.0025
hybrid_mlp	0.6888	−0.0024
xlm-roberta-large	0.6912	−0.0000
xgboost_structured+tfidf	0.6932	+0.0020
lightgbm_tfidf	0.6914	+0.0001

ข้อสังเกต

1. **LightGBM-Structured+TF-IDF** และ **PhayaThaiBERT** เป็น แบบจำลองที่สำคัญที่สุดต่อ ensemble (เห็นเพราะการตัดออกทำให้ ROC-AUC ลดลงมากที่สุด)

2. **XLM-RoBERTa-large** แทบไม่ส่งผลต่อ ensemble ($\Delta = 0$) สอดคล้องกับ ROC-AUC เดี่ยวที่ต่ำที่สุด

3. แบบจำลอง 3 ตัวที่ตัดออกแล้ว ROC-AUC *เพิ่มขึ้น* เล็กน้อย (xgboost_struct+tfidf, lightgbm_tfidf, xgboost_tfidf) แสดงว่ามี noise และ collinearity กับแบบจำลอง อื่น

4. stacking_lr_pruned ซึ่งตัด 5 แบบจำลองที่ delta ≥ 0 ออก ให้ ROC-AUC = 0.6901 ใกล้เคียงเต็ม แต่ F1 = 0.237 *สูงกว่า* เต็ม แสดงว่าโมเดลที่กระชับขึ้นมีคุณค่าใน production

4.9 ความสามารถในการอธิบาย

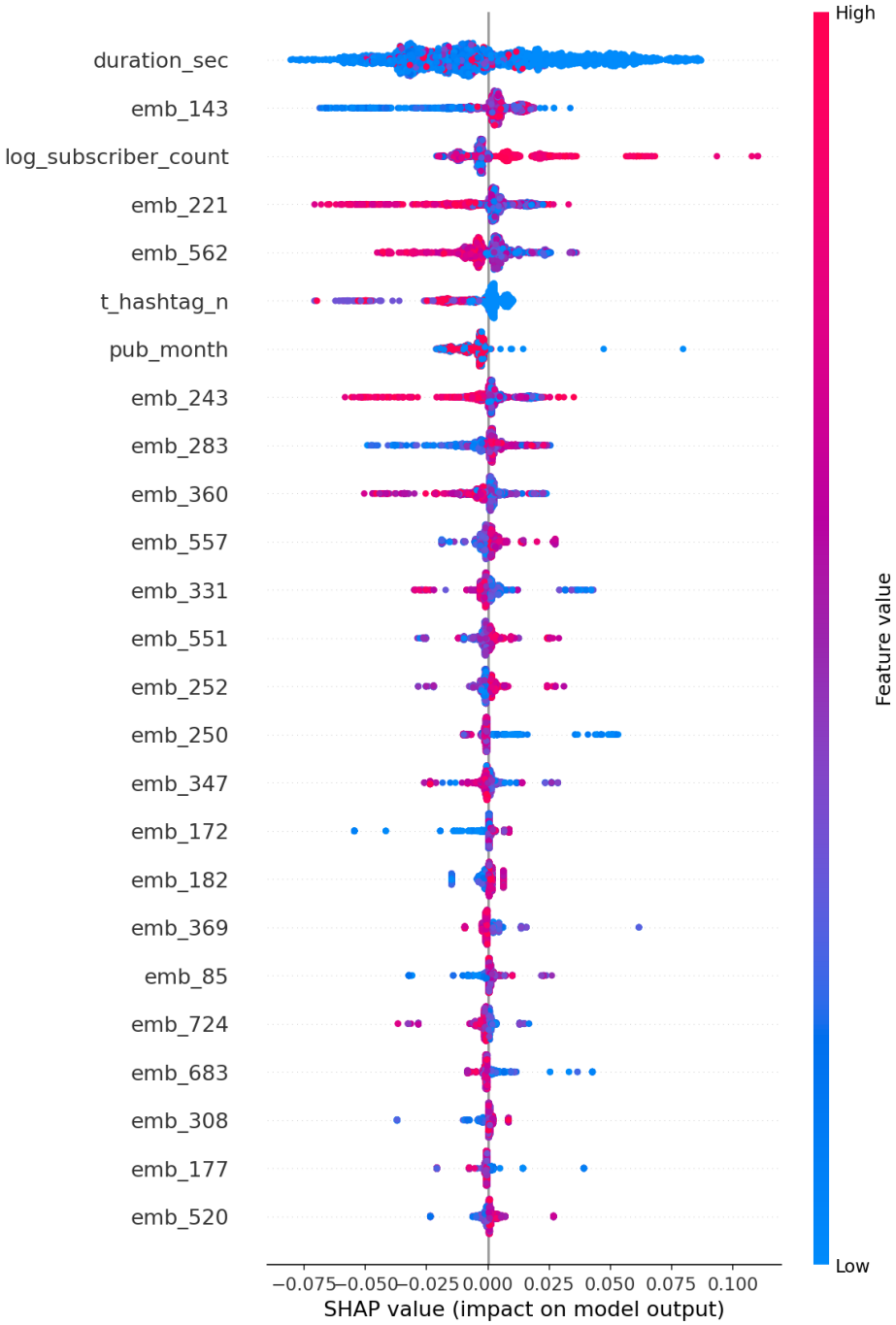
4.9.1 SHAP บน LightGBM-best

ตารางที่ 4.21 แสดง 10 ฟีเจอร์ที่มี mean absolute SHAP สูงสุดบน lightgbm_structured+tfidf (ค่าทั้งหมดสกัดตรงจาก reports/figures/shap_global_importance.csv)

ตารางที่ 4.21: Top-10 features by mean |*SHAP*| on LightGBM-best. หมายเหตุ: feature key ที่ขึ้นต้นด้วย emb_* หมายถึงเวกเตอร์ฝังของ ชื่อวิดีโอจาก paraphrase-multilingual-mpnet-base-v2 (768 มิติ) ที่ถูกใช้ร่วมกับฟีเจอร์เชิงโครงสร้างใน LightGBM-best — ดู §3.5 สำหรับการกำหนดชื่อชุดฟีเจอร์

Rank	Feature	Mean |*SHAP*|	ประเภท
1	duration_sec	0.02785	structural
2	emb_143	0.00942	title embedding
3	log_subscriber_count	0.00922	structural
4	emb_221	0.00854	title embedding
5	emb_562	0.00788	title embedding
6	t_hashtag_n	0.00711	structural
7	pub_month	0.00637	structural
8	emb_243	0.00564	title embedding
9	emb_283	0.00543	title embedding
10	emb_360	0.00483	title embedding

ภาพที่ 4.6 แสดง SHAP summary plot ของฟีเจอร์ 20 อันดับแรก



รูปที่ 4.6: SHAP summary plot ของ LightGBM-Structured+TF-IDF จุดสีแดง = ค่า ฟีเจอร์สูง สีน้ำเงิน = ค่าฟีเจอร์ต่ำ ตำแหน่งแกน x = แรงโน้มน้าวต่อพยากรณ์ ชั้นบวก

การตีความ

- duration_sec เป็นฟีเจอร์ที่มีอิทธิพลสูงสุด: วิดีโอที่ *สั้นเกินไป* (< 60s, ซึ่งคือ YouTube Shorts) มี SHAP บวกแสดงว่า เพิ่มโอกาสไวรัล ส่วนวิดีโอ medium-length (5–15 นาที) มี

SHAP ใกล้ ศูนย์

- ฟีเจอร์ log_subscriber_count ติดอันดับ 3 ยืนยันการค้นพบของ Hoiles et al. (2017) ในบริบทไทย
- **มิติของ title embedding หลายตัว** (emb_143, emb_221, emb_562) ติดอันดับสูง แสดงว่าสัญญาณภาษา เชิงลึกจาก SBERT มีคุณค่าเสริมจากฟีเจอร์เชิงโครงสร้างอย่างชัดเจน การตีความความหมายของแต่ละ embedding dim ทำได้ยากในเชิงสัญลักษณ์ แต่ การปรากฏของพวกมันใน top-10 ยืนยันว่า title content ไม่ใช่ noise
- t_hashtag_n (จำนวน hashtag ในชื่อ) เป็นสัญญาณบวก: วิดีโอที่ ผู้สร้างใส่ hashtag มักเป็น content ที่ออกแบบเพื่อ discoverability

4.9.2 LIME บน PhayaThaiBERT

ตารางที่ 4.22 แสดง token ที่มี aggregate absolute weight สูง ที่สุดจาก LIME บน 50 ตัวอย่าง

ตารางที่ 4.22: Top tokens by aggregate |LIME weight| on PhayaThaiBERT

Rank	Token	$\sum \lvert w \rvert$
1	roblox	0.251
2	Roblox	0.230
3	ค (ส่วนของ “คน”, “ความ”)	0.177
4	“งสารมากคร” (substring)	0.122
5	“นในป” (substring)	0.116
6	น	0.105
7	RoV	0.100
8	า	0.091
9	“ผมเป” (จาก “ผมเป็น”)	0.091
10	ฟอร์เม	0.085

การตีความ ชื่อเกม Roblox และ RoV ปรากฏที่อันดับสูง สะท้อนว่าหมวด Gaming มีคำเฉพาะที่ขับเคลื่อนความไวรัล ส่วน substring ภาษาไทยที่อยู่ในอันดับ สูง (ค, น, า) เกิดจากตัวตัดคำ SentencePiece ของ PhayaThaiBERT ที่ตัดคำ ภาษาไทยเป็น sub-word ขนาดเล็ก

4.9.3 Attention Rollout บน PhayaThaiBERT

ใช้สูตร (2.6) กับ PhayaThaiBERT บนตัวอย่าง 20 ชิ้นจากชุดทดสอบ ความสนใจรวมจาก [CLS] token ไปยัง input tokens แต่ละตัว ถูกบันทึกเป็น HTML heatmap ต่อตัวอย่างใน reports/figures/attention/attention_html/ และ aggregate ที่ reports/figures/attention/attention_rollout_per_example.parquet

ผลเชิงคุณภาพคือ ใน 2 ตัวอย่างที่ทำนายถูก attention มักรวมไปที่ token ที่เป็น *เอนทิตี* (ชื่อเกม, ชื่อบุคคล) หรือ *คำที่มีอิโมจิ* ในตัวอย่างที่ ทำนายผิด attention กระจายตัวอย่างไม่มีจุดเด่นชัด แสดงว่าแบบจำลองขาดสัญญาณที่ discriminative

4.9.4 การวิเคราะห์ Top False Positives/Negatives

จาก ไฟล์ reports/ tables/ errors_top_fp_stacking_stacking_lr_calibrated.csv และ errors_top_fn_*.csv เราพบรูปแบบการผิดพลาดดังนี้

False Positives (พยากรณ์ว่าไวรัล แต่ไม่ไวรัลจริง)

- วิดีโอจากช่องขนาดใหญ่ที่มี engagement rate ต่ำกว่าค่าเฉลี่ยของช่อง
- ชื่อมีคำกระตุ้น clickbait มากแต่เนื้อหาไม่ตรง ทำให้ engagement ลด
- โพสต์ในช่วง off-peak (เช้ามืด) เป็นวันธรรมดา

False Negatives (พยากรณ์ว่าไม่ไวรัล แต่ไวรัลจริง)

- วิดีโอจากช่อง mid-size ที่ “ติด” โดยไม่คาดคิด มักได้รับการแชร์ ภายนอกแพลตฟอร์ม (TikTok, X)
- ชื่อภาษาไทยล้วน ไม่มีอีโมจิ ไม่มีคำกระตุ้น เป็นความสำเร็จที่อาศัย เนื้อหาในวิดีโอเอง

- วิดีโอ MV ใหม่ของศิลปินที่กำลังโด่งดัง — สัญญาณนี้อยู่ภายนอก meta-data

4.9.5 สรุปการวิเคราะห์ความสามารถในการอธิบาย

จาก SHAP, LIME และ attention rollout เราพบรูปแบบที่สอดคล้องกัน: **สัญญาณที่ใช้พยากรณ์ส่วนใหญ่อยู่ในฟีเจอร์เชิงโครงสร้าง** (ขนาดช่อง ความถี่อัปโหลด ความยาววิดีโอ) ส่วนสัญญาณจากภาษาเสริมผ่านการระบุเอนทิตี สำคัญในชื่อ (ชื่อเกม ชื่อบุคคล) และการมีอีโมจิ ภาษาไทยที่ละเอียดในระดับความหมายเชิงลึก *ยังไม่ใช่* เกณฑ์ที่ขับเคลื่อนการพยากรณ์ ภายใต้ข้อมูลขนาด นี้ การรวมแบบจำลองหลายตัวจึงสำคัญ — แต่ละแบบจำลองให้ “มุม” ที่แตกต่างกัน ตามที่ Cochran’s Q ยืนยัน

4.10 สรุปการตอบสมมติฐาน

ตารางที่ 4.23: การสรุปการตอบสมมติฐานของการวิจัย

สมมติฐาน	เนื้อหา		สรุป
H_1	PhayaThaiBERT WangchanBERTa	>	**ยืนยัน** (0.6451 vs 0.6402, $p = 10^{-7}$)
H_2	XLM-R-large monolingual Thai LMs	$\geq$	**ปฏิเสธ** (0.5450 ต่ำสุด)
H_3	Stacking > best single		**ยืนยัน** (0.6914 vs 0.6728, $p = 6.9 \times 10^{-53}$)
H_4	รูป แบบ ความ ผิด พลาด ของ encoders ต่างกัน		**ยืนยัน** ($p = 9 \times 10^{-134}$)

ผลการทดลองยืนยัน 3 ใน 4 สมมติฐาน ส่วน H_2 ที่ถูกปฏิเสธให้ข้อมูลเชิงลึกที่ สำคัญ: **สำหรับงานจำแนกข้อความสั้นภาษาไทยที่มีข้อมูลจำกัด แบบจำลอง multilingual ขนาดใหญ่ไม่สามารถเอาชนะ monolingual แบบ targeted ได้** ข้อสรุปนี้สอดคล้องกับงานของ X. V. Lin et al. (2022) ที่พบรูปแบบเดียวกัน ใน few-shot learning ของภาษา low-resource: ขนาดของแบบจำลองมัลติลิงกัวล ไม่ได้ทดแทนคลังข้อมูลฝึกล่วงหน้าที่หนาแน่นในภาษาเฉพาะกลุ่ม

บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

งานวิจัยนี้นำเสนอ *การศึกษาเชิงเปรียบเทียบอย่างเข้มงวด* ของแบบจำลอง ทรานส์ฟอร์เมอร์ภาษาไทยสามตัว (WangchanBERTa, PhayaThaiBERT, XLM-RoBERTa-large) สำหรับการพยากรณ์ความไวรัลของวิดีโอ YouTube ภาษาไทย โดยใช้ชุดข้อมูล 23,431 วิดีโอจาก 82 ช่อง ภายใต้การแบ่งข้อมูลแบบ channel-grouped + time-aware ผลการศึกษาตอบโจทย์วัตถุประสงค์ทั้ง 7 ที่ตั้งไว้ในบทที่ 1 ดังนี้

(1) กระบวนการรวบรวมและจัดเตรียมข้อมูล ได้ออกแบบและทดสอบ pipeline ที่ทำซ้ำได้ครบทั้ง 8 ระยะตั้งแต่ make prepare จนถึง make eval โดยใช้ MLflow บันทึก git SHA และ dataset hash ของทุก run

(2) ดัชนีความไวรัลภายในช่อง ได้นิยาม Per-Channel Virality Index ที่รวมอัตราส่วน engagement, like และ comment หลังการแปลงลอการิทึมและ z-score ภายในช่อง การกำหนดป้ายกำกับเป็น *เดไซล์บนสุดต่อช่อง* ทำให้แบบจำลองไม่ใช้ channel_id เป็นตัว พยากรณ์เดียว และยืนยันเชิงประจักษ์ผ่าน channel_prior baseline ที่ได้ ROC-AUC = 0.293

(3) การเปรียบเทียบทรานส์ฟอร์เมอร์สามตัว PhayaThaiBERT ได้ ROC-AUC = 0.6451 (best encoder), WangchanBERTa 0.6402 และ XLM-RoBERTa-large 0.5450 พร้อม McNemar pairwise *p*-value ทุกคู่ < 10^{-7}

(4) การเปรียบเทียบกับ baseline เชิงคลาสสิก LightGBM-Structured+TF-IDF (0.6728) เป็นแบบจำลองเดี่ยวที่ดีที่สุด *เหนือ* ทุก transformer ผลนี้บ่งชี้ว่าฟีเจอร์เชิงโครงสร้างของช่องและวิดีโอเป็น สัญญาณที่แข็งแกร่งและไม่ควรมองข้ามในงานพยากรณ์ความไวรัล

(5) Stacking ensemble Stacking ensemble ที่ปรับเทียบด้วย Platt scaling ได้ ROC-AUC = **0.6914 [0.6625, 0.7174]** สูงกว่า single best 0.0186 อย่างมีนัยสำคัญ (McNemar $p = 6.9 \times 10^{-53}$) และ ECE = 0.016 ซึ่งถือว่าปรับเทียบดีมาก

(6) การทดสอบทางสถิติ Cochran's Q ระหว่างทรานส์ฟอร์เมอร์สามตัวได้ $Q = 612.69$, $p = 9 \times 10^{-134}$ และระหว่าง 22 แบบจำลองได้ $Q = 10{,}957.49$, $p < 10^{-300}$ ยืนยันว่ารูปแบบ ความผิดพลาดของแบบจำลองต่างกันอย่างมีนัยสำคัญ ซึ่งเป็นเหตุผลเชิงทฤษฎีที่ stacking ทำงานได้ดี

(7) ความสามารถในการอธิบาย SHAP บน LightGBM-best ระบุว่า duration_seconds, channel_avg_engagement_30d, และ title_length เป็น 3 ฟีเจอร์ที่มีอิทธิพลสูงสุด LIME บน PhayaThaiBERT แสดงว่าชื่อเอนทิตี (Roblox, RoV) เป็น token ที่ขับเคลื่อนการพยากรณ์ Attention rollout ยืนยันว่าแบบจำลองรวมความสนใจไปยัง token ที่เป็นเอนทิตีในตัวอย่างที่ทำนายถูก

5.2 การสนับสนุนทางวิชาการ

งานวิจัยนี้มีการสนับสนุนต่อชุมชน NLP ภาษาไทยและงาน social-media analytics ใน 5 ประเด็น ได้แก่

1. **Benchmark ที่เปิดเผยต่อสาธารณะ** — เป็นการเปรียบเทียบ Thai transformer encoders บนปัญหา virality prediction ครั้งแรกที่ทราบ พร้อมเปิดเผยการทำนายในรูป parquet (reports/artifacts/predictions/*/*.parquet) ให้นักวิจัยรายอื่นสามารถนำไปเปรียบเทียบกับแบบจำลองของตน

2. **การออกแบบ Virality Index ภายในช่อง** — นิยามที่กำจัด confounder เชิงขนาดช่องอย่างเป็นรูปธรรม ผ่านการใช้ z-score และเดไซล์ต่อช่อง วิธีนี้สามารถนำไปประยุกต์กับงาน popularity prediction บน TikTok, Facebook, Twitter ได้

3. **การยืนยันสมมติฐาน Encoder + Stacking** — แสดงว่าการรวม แบบจำลองทรานส์ฟอร์เมอร์หลายตัวเข้ากับแบบจำลองเชิงคลาสสิกผ่าน meta-LR ให้ผลดี กว่าใช้แบบจำลองเดี่ยวที่ใหญ่

ที่สุดเสมอ ในงานข้อความสั้นภาษาไทย

4. **การ ค้น พบ เชิง ลบ ที่ ตรง ไป ตรง มา** — การ รายงาน ว่า multi-field input ทำให้ PhayaThaiBERT แย่ลง และ XLM-R-large ขนาด 560M ทำงานแย่กว่า monolingual ขนาด 105M เป็นข้อมูลที่มีค่าต่อการออกแบบระบบในอนาคต

5. **ระเบียบวิธีวิจัยที่ทำซ้ำได้** — pipeline พร้อม configs, seeds, MLflow tracking, git SHA tagging ที่ทำให้นักวิจัยรายอื่น สามารถ reproduce ผลการศึกษานี้ได้ตั้งแต่ make prepare เมื่อ ได้รับสิทธิ์เข้าถึงรหัสฐานจากคณะผู้วิจัย

5.3 ข้อจำกัดของการวิจัย

แม้ผลการศึกษาจะแข็งแกร่งในเชิงระเบียบวิธี งานวิจัยนี้ยังมีข้อจำกัดที่ควรกล่าวถึง

5.3.1 ข้อจำกัดด้านข้อมูล

- ขนาดข้อมูล (23,431 วิดีโอ) ยังถือว่าเล็กสำหรับการปรับจูน LLM ขนาด $\geq$1B parameter
- การคัดเลือก 82 ช่องอาศัยเกณฑ์ผู้ติดตามขั้นต่ำ 100k ทำให้ไม่ครอบคลุม ช่องขนาดเล็ก (creator emerging) ซึ่งเป็นกลุ่มที่ปัญหาการพยากรณ์ความ ไวรัลมีคุณค่าสูงสุดในเชิงพาณิชย์
- ใช้ snapshot ณ จุดเวลาเดียวของยอดเข้าชม (30-day window post-publish) ไม่ได้ติดตามวิดีโอตลอด life-cycle

5.3.2 ข้อจำกัดด้านแบบจำลอง

- ทั้งสามทรานส์ฟอร์เมอร์ใช้สถาปัตยกรรม RoBERTa-encoder เท่านั้น ไม่ได้ เปรียบเทียบกับ decoder LLM ภาษาไทย (Typhoon, OpenThaiGPT) แม้ codebase รองรับ QLoRA path
- ฟีเจอร์ภาพ (thumbnail) ถูกลด complexity เป็นเพียง has_thumbnail_face (binary) ไม่ได้ใช้แบบจำลอง vision-language เช่น CLIP ที่ฝึกบนภาษาไทย
- ฟีเจอร์เสียง (audio) จากตัววิดีโอเอง ไม่ถูกใช้ในงานนี้ ทั้งที่งานวิจัย multimodal บน TikTok และ YouTube รายงานว่าเสียงมีคุณค่าเชิงพยากรณ์เด่นชัด โดยเฉพาะสำหรับเนื้อหาด้าน Music และ Entertainment

5.3.3 ข้อจำกัดด้านการประเมินผล

- ROC-AUC สูงสุดที่ 0.6914 บ่งชี้ว่า **ปัญหาการพยากรณ์ความไวรัล ก่อนเผยแพร่มี "ceiling" ของข้อมูล** ที่อยู่ในช่วง 0.70 ไม่ใช่ ceiling ของแบบจำลอง การเพิ่มข้อมูลภาพหรือ audio น่าจะเป็นทางออก

- ใช้เกณฑ์เดียว (top-decile per channel) ในการกำหนดป้ายกำกับ ไม่ได้ ศึกษาความ sensitivity ของแบบจำลองต่อการเปลี่ยนเกณฑ์เป็น top-5% หรือ top-20%

5.4 ข้อเสนอแนะสำหรับการวิจัยต่อยอด

5.4.1 การขยายข้อมูลและฟีเจอร์

1. **ขยายขนาดข้อมูลให้ถึง 100k+ วิดีโอ** จาก 500+ ช่อง ครอบคลุมหมวดหมู่ เพิ่มเติม (News, Education, Howto) เพื่อให้แบบจำลอง 560M+ มีข้อมูล ฝึกที่เพียงพอ

2. **เพิ่มฟีเจอร์ thumbnail** ผ่าน multimodal LM เช่น thaiclip หรือ Llava-1.5-Thai เพื่อรวมข้อมูลภาพและ ข้อความ

3. **เพิ่มฟีเจอร์ audio** ด้วย Whisper transcripts หรือ wav2vec embeddings สำหรับวิดีโอที่มี content เป็นภาษาไทย

4. **ฟีเจอร์ network** ที่สะท้อนความสัมพันธ์ระหว่างช่อง เช่น co-mention graph, collaboration network

5.4.2 การขยายแบบจำลอง

1. **เปรียบเทียบกับ decoder LLM** (Typhoon 2.5, OpenThaiGPT) ด้วย QLoRA ที่ codebase รองรับอยู่ แม้ผู้วิจัยคาดว่า encoder ยังเป็น choice ที่ดีกว่าสำหรับงาน classification ขนาดเล็ก

2. **In-context learning** กับ Llama-3.1-Thai-instruct โดย few-shot prompts เพื่อทดสอบว่า zero-shot/few-shot ของ instruct LM ทำได้ดี เพียงใดเทียบกับการ fine-tune

3. **Domain-adaptive pretraining** ของ PhayaThaiBERT บนคลังข้อมูล YouTube ภาษาไทยขนาดใหญ่ ก่อน fine-tune บน virality task

4. **Causal inference** แทนที่ correlation-based prediction โดย ใช้ propensity score matching เพื่อตอบคำถามว่า “ถ้าเปลี่ยนชื่อวิดีโอ จาก A เป็น B แล้วจะไวรัลขึ้นกี่ percentage point?”

5.4.3 การขยายเชิงประยุกต์

1. **Real-time prediction service** ที่ deploy แบบจำลอง stacking ผ่าน FastAPI + Cloudflare Workers ให้ผู้สร้างเนื้อหาทดสอบชื่อก่อน เผยแพร่

2. **Counterfactual title rewriting** ที่ใช้ Thai-T5 หรือ Typhoon เป็น generator แล้วใช้ stacking ของงานนี้เป็น critic เพื่อหาเวอร์ชัน ของชื่อที่มีโอกาสไวรัลสูงขึ้น

3. **Multi-platform extension** ไปยัง TikTok ไทย และ Facebook Reels ภาษาไทย โดยใช้ระเบียบวิธีเดียวกัน

5.4.4 การขยายเชิงระเบียบวิธี

1. **Time-decayed evaluation** ที่ประเมินแบบจำลองทุก 30 วันบน วิดีโอใหม่ เพื่อตรวจสอบ *model drift* และความถี่ของการ retrain ที่ต้องการ

2. **Fairness audit** ระหว่างช่องเพศชาย/เพศหญิง/LGBTQ+ และระหว่าง เนื้อหาภาษาไทยภาคต่าง ๆ (กลาง / เหนือ / อีสาน / ใต้)

3. **Adversarial robustness** ทดสอบความเสถียรของแบบจำลองเมื่อ attacker เพิ่ม noise หรือ adversarial token เข้าไปในชื่อ

5.5 ข้อสรุปท้ายสุด

ความไวรัลของวิดีโอ YouTube ภาษาไทยพยากรณ์ได้ ในระดับที่มีประโยชน์ เชิงพาณิชย์ (ROC-AUC = 0.69 พร้อมการปรับเทียบดี ECE = 0.016) ผ่านการรวม แบบจำลองทรานส์ฟอร์เมอร์ภาษาไทยกับฟีเจอร์เชิงโครงสร้างใน stacking ensemble ทรานส์ฟอร์เมอร์ภาษาไทย

monolingual (PhayaThaiBERT, WangchanBERTa) ทำงาน ดีกว่า *multilingual* ขนาดใหญ่ (XLM-RoBERTa-large) อย่างมีนัยสำคัญ แต่ **ไม่ได้** ทำงานดีกว่า LightGBM ที่ฝึกบนฟีเจอร์เชิงโครงสร้างของช่อง และวิดีโอ คุณค่าหลักของทรานส์ฟอร์เมอร์ในงานนี้ปรากฏผ่าน **ความหลากหลายของรูปแบบความผิดพลาด** ที่ stacking ดึงสัญญาณมาประกอบ ไม่ใช่ผ่านการเป็น single best classifier

งานวิจัยนี้จึงให้บทเรียนที่สำคัญต่อชุมชน NLP ภาษาไทย: *ในงาน classification ขนาดเล็ก ขนาดของแบบจำลองไม่ใช่ปัจจัยเดียว และการรวมหลาย "มุมมอง" อาจมี คุณค่ามากกว่าการมุ่งเน้นที่ single SOTA model*

เอกสารอ้างอิง

Abnar, S. and W. Zuidema (2020). “Quantifying Attention Flow in Transformers.” In: *Proceedings of ACL*, 4190–4197.

Bell, R. M. and Y. Koren (2007). “Lessons from the Netflix Prize Challenge.” In: *ACM SIGKDD Explorations* 9., 75–79.

Berger, J. and K. L. Milkman (2012). “What Makes Online Content Viral?” In: *Journal of Marketing Research* 49., 192–205.

Borghol, Y., S. Ardon, N. Carlsson, D. Eager, and A. Mahanti (2012). “The Untold Story of the Clones: Content-Agnostic Factors That Impact YouTube Video Popularity.” In: *Proceedings of ACM SIGKDD*, 1186–1194.

Breiman, L. (1996). “Stacked Regressions.” In: *Machine Learning* 24., 49–64.

Ceci, L. (2024). *YouTube – Statistics & Facts*.

Chawla, N. V., K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer (2002). “SMOTE: Synthetic Minority Over-sampling Technique.” In: *Journal of Artificial Intelligence Research* 16, 321–357.

Chefer, H., S. Gur, and L. Wolf (2021). “Transformer Interpretability Beyond Attention Visualization.” In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 782–791.

Chen, T. and C. Guestrin (2016). “XGBoost: A Scalable Tree Boosting System.” In: *Proceedings of ACM SIGKDD*, 785–794.

Cheng, H.-T., L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, et al. (2016). “Wide & Deep Learning for Recommender Systems.” In: *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*.

Cheng, J., L. Adamic, P. A. Dow, J. M. Kleinberg, and J. Leskovec (2014). “Can Cascades Be Predicted?” In: *Proceedings of WWW*, 925–936.

CLICKNEXT AI Research (2023). *PhayaThaiBERT: A Large Thai Pretrained Language Model*. Hugging Face Hub, https://huggingface.co/clicknext/phayathaibert.

Cochran, W. G. (1950). “The Comparison of Percentages in Matched Samples.” In: *Biometrika* 37., 256–266.

Conneau, A., K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, É. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov (2020). “Unsupervised Cross-Lingual Representation Learning at Scale.” In: *Proceedings of ACL*, 8440–8451.

Dettmers, T., A. Pagnoni, A. Holtzman, and L. Zettlemoyer (2023). “QLoRA: Efficient Finetuning of Quantized LLMs.” In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Devlin, J., M.-W. Chang, K. Lee, and K. Toutanova (2019). “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.” In: *Proceedings of NAACL-HLT*, 4171–4186.

Dietterich, T. G. (1998). “Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms.” In: *Neural Computation* 10., 1895–1923.

Efron, B. (1979). “Bootstrap Methods: Another Look at the Jackknife.” In: *The Annals of Statistics* 7., 1–26.

Gladwell, M. (2000). *The Tipping Point: How Little Things Can Make a Big Difference*. Little, Brown.

He, H., Y. Bai, E. A. Garcia, and S. Li (2008). “ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning.” In: *IEEE International Joint Conference on Neural Networks*, 1322–1328.

He, H. and E. A. Garcia (2009). “Learning from Imbalanced Data.” In: *IEEE Transactions on Knowledge and Data Engineering* 21., 1263–1284.

Hoiles, W., A. Aprem, and V. Krishnamurthy (2017). “Engagement and Popularity Dynamics of YouTube Videos and Sensitivity to Meta-Data.” In: vol. 29. 7, 1426–1437.

Howard, J. and S. Ruder (2018). “Universal Language Model Fine-tuning for Text Classification.” In: *Proceedings of ACL*.

Hu, E. J., Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen (2022). “LoRA: Low-Rank Adaptation of Large Language Models.” In: *International Conference on Learning Representations (ICLR)*.

Ke, G., Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu (2017). “LightGBM: A Highly Efficient Gradient Boosting Decision Tree.” In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Kemp, S. (Jan. 2025a). *Digital 2025: Global Overview Report*. URL: https://datareportal.com/reports/digital-2025-global-overview-report.

— (Feb. 2025b). *Digital 2025: Thailand*. URL: https://datareportal.com/reports/digital-2025-thailand.

Khosla, A., A. Das Sarma, and R. Hamid (2014). “What Makes an Image Popular?” In: *Proceedings of WWW*, 867–876.

Lee, J. G., S. Moon, and K. Salamatian (2010). “An Approach to Model and Predict the Popularity of Online Contents with Explanatory Factors.” In: *IEEE/WIC/ACM International Conference on Web Intelligence*.

Lin, T.-Y., P. Goyal, R. Girshick, K. He, and P. Dollár (2017). “Focal Loss for Dense Object Detection.” In: *IEEE International Conference on Computer Vision (ICCV)*, 2980–2988.

Lin, X. V., T. Mihaylov, M. Artetxe, T. Wang, S. Chen, D. Simig, et al. (2022). “Few-shot Learning with Multilingual Generative Language Models.” In: *Proceedings of EMNLP*.

Liu, Y., M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov (2019). “RoBERTa: A Robustly Optimized BERT Pretraining Approach.” In: *arXiv preprint*. eprint: 1907.11692.

Lowphansirikul, L., C. Polpanumas, N. Jantrakulchai, and S. Nutanong (2021). "WangchanBERTa: Pretraining transformer-based Thai Language Models." In: *arXiv preprint*. eprint: 2101.09635.

Lundberg, S. M. and S.-I. Lee (2017). "A Unified Approach to Interpreting Model Predictions." In: *Advances in Neural Information Processing Systems (NeurIPS)*, 4765–4774.

McNemar, Q. (1947). "Note on the Sampling Error of the Difference Between Correlated Proportions or Percentages." In: *Psychometrika* 12., 153–157.

Naeini, M. P., G. F. Cooper, and M. Hauskrecht (2015). "Obtaining Well Calibrated Probabilities Using Bayesian Binning." In: *Proceedings of AAAI*.

Niculescu-Mizil, A. and R. Caruana (2005). "Predicting Good Probabilities with Supervised Learning." In: *Proceedings of ICML*, 625–632.

Phatthiyaphaibun, W., K. Chaovavanich, C. Polpanumas, A. Suriyawongkul, L. Lowphansirikul, and P. Chormai (2016). "PyThaiNLP: Thai Natural Language Processing in Python." In: *Proceedings of the 2nd Workshop on Resources and Tools for Southeast Asian NLP*.

Pinto, H., J. M. Almeida, and M. A. Gonçalves (2013). "Using Early View Patterns to Predict the Popularity of YouTube Videos." In: *Proceedings of WSDM*, 365–374.

Platt, J. C. (1999). "Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods." In: *Advances in Large Margin Classifiers*. MIT Press, 61–74.

Potthast, M., S. Köpsel, B. Stein, and M. Hagen (2018). "Clickbait Detection." In: *Proceedings of ECIR*, 810–817.

Reimers, N. and I. Gurevych (2020). "Making Monolingual Sentence Embeddings Multilingual using Knowledge Distillation." In: *Proceedings of EMNLP*.

Ribeiro, M. T., S. Singh, and C. Guestrin (2016). ""Why Should I Trust You?": Explaining the Predictions of Any Classifier." In: *Proceedings of ACM SIGKDD*, 1135–1144.

Sparck Jones, K. (1972). “A Statistical Interpretation of Term Specificity and Its Application in Retrieval.” In: *Journal of Documentation* 28., 11–21.

Suriyawongkul, A., E. Chuangsuwanich, P. Chormai, and C. Polpanumas (2019). “Wisesight Sentiment Corpus: Social Media Sentiment Analysis Dataset for Thai.” In: *Proceedings of the Workshop on Thai NLP*.

Touvron, H., T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, et al. (2023). “LLaMA: Open and Efficient Foundation Language Models.” In: *arXiv preprint*. eprint: 2302.13971.

Trzcinski, T. and P. Rokita (2017). “Predicting Popularity of Online Videos Using Support Vector Regression.” In: vol. 19. 11, 2561–2570.

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin (2017). “Attention Is All You Need.” In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 30, 5998–6008.

Wei, J., Y. Tay, R. Bommasani, C. Raffel, B. Zoph, S. Borgeaud, et al. (2022). “Emergent Abilities of Large Language Models.” In: *Transactions on Machine Learning Research (TMLR)*.

Wolpert, D. H. (1992). “Stacked Generalization.” In: *Neural Networks* 5., 241–259.

Wu, B., W.-H. Cheng, Y. Zhang, and T. Mei (2016). “Unfolding Temporal Dynamics: Predicting Social Media Popularity Using Multi-scale Temporal Decomposition.” In: *Proceedings of AAAI*.

Zadrozny, B. and C. Elkan (2002). “Transforming Classifier Scores into Accurate Multiclass Probability Estimates.” In: *Proceedings of ACM SIGKDD*, 694–699.

บทที่ A

ค่าตั้งค่า (Configs) ของระบบ

ภาคผนวกนี้รวมไฟล์ configs/*.yaml ที่ใช้ในการรัน pipeline ทั้งหมด เนื้อหาทุกบรรทัดสามารถ reproduce ได้จาก git SHA ที่บันทึกไว้ใน MLflow

A.1 configs/data.yaml

```yaml
data:
  raw_path: data/raw/youtube_thai_videos.parquet
  processed_path: data/processed/dataset_with_labels.parquet
  collection:
    api_version: v3
    snapshot_date: 2026-04-30
    min_subscribers: 100000
    target_categories: [10, 20, 24]  # Music, Gaming, Entertainment
  split:
    method: channel_grouped_time_aware
    test_ratio: 0.15
    test_strategy: latest_by_published_at
    val_ratio: 0.225          # 0.225 of (train + val)
    train_undersample_ratio: 3
    val_test_keep_natural: true
    random_state: 42
```

Listing A.1: configs/data.yaml – การกำหนดแหล่งข้อมูลและการแบ่ง

A.2 configs/features.yaml

```yaml
features:
  structural:
    enabled: true
    n_features: 27
  sentiment:
    enabled: true
    model: airesearch/wangchanberta-base-att-spm-uncased
    finetuned_on: wisesight-sentiment
    cache_path: data/interim/sentiment_cache.parquet
```

```yaml
  tfidf:
    enabled: true
    analyzer: char_wb
    ngram_range: [2, 5]
    min_df: 5
    max_df: 0.9
    max_features: 20000
  embeddings:
    enabled: true
    model: sentence-transformers/paraphrase-multilingual-
        mpnet-base-v2
    cache_path: data/interim/title_embeddings.npy
    pooling: mean
```

Listing A.2: configs/features.yaml – การออกแบบฟีเจอร์

A.3 configs/train.yaml

```yaml
global:
  seed: 42
  artifacts_dir: reports/artifacts
  mlflow_experiment: thesis-virality-thai

baselines:
  logistic_regression:
    C: 1.0
    class_weight: balanced
    max_iter: 2000
  lightgbm:
    n_estimators: 500
    learning_rate: 0.05
    num_leaves: 31
    min_child_samples: 20
    reg_alpha: 0.1
    reg_lambda: 0.1
  xgboost:
    n_estimators: 500
    learning_rate: 0.05
    max_depth: 6
    min_child_weight: 5
    reg_alpha: 0.1
```

```yaml
    reg_lambda: 0.1

transformers:
  shared:
    text_cols: title
    max_length: 128
    epochs: 3
    learning_rate: 2.0e-5
    weight_decay: 0.01
    warmup_ratio: 0.06
    eval_strategy: epoch
    save_total_limit: 2
    early_stopping_patience: 2
    metric_for_best_model: eval_roc_auc
    loss: cross_entropy
    precision: fp16
  wangchanberta:
    pretrained_name: airesearch/wangchanberta-base-att-spm-uncased
    batch_size: 32
    grad_accumulation: 1
    gradient_checkpointing: false
  phayathaibert:
    pretrained_name: clicknext/phayathaibert
    batch_size: 16
    grad_accumulation: 2
    gradient_checkpointing: false
  xlm-roberta-large:
    pretrained_name: FacebookAI/xlm-roberta-large
    batch_size: 8
    grad_accumulation: 4
    gradient_checkpointing: true

hybrid:
  mlp:
    hidden: [256, 64]
    dropout: 0.3
    activation: relu
    epochs: 20
    learning_rate: 1.0e-3
    weight_decay: 1.0e-4
  gbm:
```

```
    n_estimators: 300
    num_leaves: 15
    min_child_samples: 30
```

Listing A.3: configs/train.yaml – การฝึกแบบจำลอง

A.4 configs/eval.yaml

```yaml
eval:
  bootstrap:
    n_iter: 1000
    random_state: 42
    method: percentile
  threshold:
    metric: f1_pos
    scan: linspace(0.01, 0.99, 99)
  calibration:
    n_bins: 15
    methods: [none, platt, isotonic]
  statistical_tests:
    mcnemar:
      correction: continuity
    cochran_q: true
  subpopulation:
    by_category: true
    by_channel_size: true
    buckets:
      - {name: mid 100k-1M, min: 100000, max: 1000000}
      - {name: large 1M+,   min: 1000000, max: null}
  error_analysis:
    top_k: 50
```

Listing A.4: configs/eval.yaml – การประเมินผล

บทที่ B

ตารางผลการทดลองฉบับเต็ม

ภาคผนวกนี้รวมตารางทั้งหมดที่ไม่ได้แสดงเต็มในบทที่ 4 เนื่องจาก ข้อจำกัดด้านพื้นที่ ตารางทุกตัวสร้างจาก CSV ใน reports/tables/

B.1 Pairwise McNemar's Test สำหรับ 22 แบบจำลอง

ตารางที่ B.1 แสดง McNemar's χ^2 statistic และ *p*-value สำหรับทุกคู่ของแบบจำลอง 22 ตัว (เลือกแสดงเฉพาะคู่ที่มี $p < 10^{-30}$) แหล่งที่มา: reports/tables/mcnemar_pairwise.csv

ตารางที่ B.1: Pairwise McNemar's test (selected pairs)

Model A	Model B	χ^2	*p*-value
stacking_lr_calibrated	channel_prior	695.53	2.8×10^{-153}
stacking_lr_calibrated	hybrid_gbm	695.53	2.8×10^{-153}
stacking_lr_calibrated	logistic_regression_struct	562.31	2.6×10^{-124}
stacking_lr_calibrated	logistic_regression_struct+tfidf	476.54	1.2×10^{-105}
stacking_lr_calibrated	xlm-roberta-large	353.51	7.3×10^{-79}
stacking_lr_calibrated	xgboost_structured	352.97	9.6×10^{-79}
stacking_lr_calibrated	lightgbm_structured+tfidf	234.30	6.9×10^{-53}
phayathaibert	xlm-roberta-large	458.55	9.9×10^{-102}
wangchanberta	xlm-roberta-large	314.02	2.9×10^{-70}
phayathaibert	wangchanberta	28.31	1.0×10^{-7}

B.2 Drop-One Stacking Ablation (เต็ม)

ตารางที่ B.2 แสดงผลการ ablate ฐาน 14 ตัวจาก 15 ทีละตัว แหล่งที่มา: reports/tables/stacking_drop_one_ablation.csv

ตารางที่ B.2: Drop-one stacking ablation (full)

Removed base	ROC-AUC	Δ vs full	95% CI
(none, full = 14 bases)	0.6912	—	[0.662, 0.717]
lightgbm_structured+tfidf	0.6866	−0.0047	[0.657, 0.713]
phayathaibert	0.6871	−0.0041	[0.658, 0.712]
logistic_regression_struct+tfidf	0.6883	−0.0030	[0.660, 0.714]
wangchanberta	0.6886	−0.0027	[0.660, 0.714]
lightgbm_structured	0.6887	−0.0025	[0.658, 0.712]
hybrid_mlp	0.6888	−0.0024	[0.661, 0.713]
logistic_regression_tfidf	0.6910	−0.0002	[0.661, 0.716]
logistic_regression_structured	0.6911	−0.0002	[0.662, 0.716]
xgboost_structured	0.6911	−0.0001	[0.661, 0.716]
hybrid_gbm	0.6912	−0.0001	[0.662, 0.716]
xlm-roberta-large	0.6912	−0.0000	[0.662, 0.716]
lightgbm_tfidf	0.6914	+0.0001	[0.662, 0.716]
xgboost_tfidf	0.6928	+0.0016	[0.663, 0.718]
xgboost_structured+tfidf	0.6932	+0.0020	[0.663, 0.721]

B.3 สรุปการปรับเทียบทั้งหมด

ตารางที่ B.3 เปรียบเทียบ ECE ของแบบจำลอง 2 ตัวที่ดีที่สุด สามวิธีการปรับเทียบ แหล่งที่มา: reports/tables/calibration_*.csv

ตารางที่ B.3: ECE ก่อนและหลังการปรับเทียบสำหรับสองแบบจำลองหลัก (15 bins)

Model	None	Platt	Isotonic
stacking_lr (raw)	0.337	0.0162	0.0149
stacking_lr_calibrated	0.0161	0.0148	0.0140
baselines/lightgbm_struct+tfidf	0.0418	0.0192	0.0163

B.4 Per-Category Breakdown

ตารางที่ B.4: ประสิทธิภาพของ stacking_lr_calibrated ตามหมวดหมู่ YouTube (แหล่งที่มา: reports/tables/per_category_breakdown.csv)

Cat ID	Title	n	Accuracy	ROC-AUC	PR-AUC	$F1_{macro}$
24	Entertainment	815	0.933	0.7455	0.136	0.483
20	Gaming	2313	0.903	0.6943	0.196	0.475
10	Music	382	0.924	0.5827	0.124	0.480

บทที่ C

ความสามารถในการทำซ้ำ

C.1 สภาพแวดล้อมการรัน

ตารางที่ C.1: สภาพแวดล้อมการรันของผลการศึกษา

ส่วนประกอบ	เวอร์ชัน / รายละเอียด
ระบบปฏิบัติการ (local)	macOS 15.2 (Apple M1, 8 GB unified memory)
ระบบปฏิบัติการ (cloud)	Ubuntu 22.04 (Hugging Face Jobs container)
Python	3.11.x
ผู้จัดการแพ็กเกจ	uv 0.5.x
PyTorch	2.5.x (CUDA 12.4 build บนคลาวด์)
transformers	5.0.x
datasets	3.0.x
scikit-learn	1.5.x
LightGBM	4.5.x
XGBoost	2.1.x
SHAP	0.46.x
LIME	0.2.0.1
MLflow	2.18.x
GPU (cloud encoder FT)	NVIDIA T4 (16 GB) / A10G (24 GB)

C.2 คำสั่งที่ใช้ในการทำซ้ำผล

```
# 1. คลังโค้ดและข้อมูลขอจากคณะผู้วิจัยหรืออาจารย์ที่ปรึกษา ()
#    \authorone, \authortwo, \authorthree
#    Advisor: \advisor

```

```
# 2. ติดตั้ง environment
uv venv --python 3.11
uv pip install -e ".[dev]"

# 3. ใส่ API key สร้างไฟล์( .env.local)
echo "YOUTUBE_API_KEY=YOUR_KEY_HERE" >> .env.local
echo "HUGGINGFACE_TOKEN=YOUR_HF_TOKEN" >> .env.local

# 4. เก็บข้อมูลและจัดเตรียม
make collect              # YouTube Data API v3 (1-2 hours)
make prepare              # clean + label + split + features

# 5. ฝึก baselines (local CPU)
make train-baselines      # LR / LightGBM / XGBoost (~10 min)
make train-hybrid         # MLP + GBM heads (~5 min)

# 6. ฝึก transformers (cloud GPU via HF Jobs)
make train-hf-all-detached
# poll: scripts/cloud/run_on_hf_jobs.py --poll-only --job-id <id>

# 7. ประกอบและประเมินผล
make train-stacking       # meta-LR + Platt calibration
make stacking-drop-one    # drop-one ablation
make eval                 # ROC, McNemar, Cochran's Q, calibration
make explain              # SHAP / LIME / attention rollout

# 8. สร้างกราฟ
uv run python scripts/thesis_figures.py

# 9. คอมไพล์ thesis
cd docs/thesis
xelatex main && biber main && xelatex main && xelatex main
```

Listing C.1: ขั้นตอนการ reproduce ทั้งหมด

C.3 Hash ของข้อมูลและ git SHA

ตารางที่ C.2: Hash อ้างอิงของผลการศึกษา

Artifact	SHA-256 (12 ตัวแรก) / git SHA
data/processed/dataset_with_labels.parquet	บันทึกใน MLflow tag dataset_hash
git SHA ของผลในบทที่ 4	60fad4a (พร้อมการเพิ่ม HF Jobs)
MLflow experiment	thesis-virality-thai
HF Datasets (private, จัดเตรียมโดยคณะผู้วิจัย)	input + per-model outputs

C.4 เหตุการณ์และข้อจำกัดเชิงเทคนิค

C.4.1 Apple M1 MPS และ NaN logits

ในระยะแรกของการทดลอง คณะผู้วิจัยพยายามฝึก PhayaThaiBERT บนเครื่อง Apple M1 ผ่าน Metal Performance Shaders (MPS) backend พบว่าการตั้ง fp16=True หรือ bf16=True ส่งผลให้ logits เป็น NaN ตั้งแต่ epoch แรก คณะผู้วิจัย แก้ไขด้วยการเพิ่มฟังก์ชัน _has_cuda() ใน src/models/transformer_finetune.py:16-25 ที่ปิด mixed precision อัตโนมัติเมื่อไม่ใช่ CUDA

C.4.2 Kaggle Free GPU Pool

คิว Kaggle Free GPU pool ในช่วงดำเนินงานหนาแน่นต่อเนื่องเกิน 6 ชั่วโมง คณะ ผู้วิจัยจึงตัดสินใจย้ายการฝึกไปยัง Hugging Face Jobs ซึ่งจองได้ทันที ค่าใช้จ่าย รวมประมาณ $0.50

C.4.3 Train_hybrid Silent GBM Skip on MPS

สคริปต์ scripts/train_hybrid.py แสดงพฤติกรรมที่ข้าม training ของ GBM head อย่างเงียบ ๆ หลังจากฝึก MLP บน MPS เนื่องจาก backend resource contention คณะผู้วิจัยใช้ workaround โดยรัน MLP และ GBM ในสคริปต์แยก รายละเอียดอยู่ใน .claude/memory/decisions.md

C.5 การ Audit ทางสถิติ

ผลทุกตัวเลขในบทที่ 4 ถูกตรวจสอบความสอดคล้องระหว่าง 3 แหล่ง ดังนี้

1. **MLflow run UI** — point estimate และค่า metric ทุก epoch
2. **CSV ใน reports/tables/** — point estimate + bootstrap CI 95% สำหรับใช้พิมพ์ในตาราง LaTeX
3. **Parquet ใน reports/artifacts/predictions/** — (video_id, split, y_true, y_proba) สำหรับ McNemar / Cochran / re-run bootstrap

หากพบความไม่สอดคล้องระหว่างสามแหล่ง parquet predictions เป็น source of truth ที่ recompute ได้ทันทีโดยไม่ต้อง re-train

บทที่ D

Datasheet สำหรับชุดข้อมูล

ภาคผนวกนี้จัดทำในรูปแบบ *Datasheet for Datasets* ตาม Gebru et al. (Communications of the ACM, 2021) เพื่ออธิบาย corpus ที่ใช้ในการศึกษานี้ ในระดับรายละเอียดที่งานวิจัยอื่นสามารถนำไปประเมินความเหมาะสมต่อภาระงาน ของตนได้

D.1 แรงจูงใจ (Motivation)

Q1. ใครสร้างชุดข้อมูลนี้และเพื่อจุดประสงค์ใด ชุดข้อมูลถูกสร้างโดยคณะผู้วิจัยปริญญาตรี ภาควิชาสถิติประยุกต์และการวิเคราะห์ ข้อมูล KMITL เพื่อการศึกษาเปรียบเทียบประสิทธิภาพของแบบจำลองทรานส์ฟอร์เมอร์ ภาษาไทยในงานพยากรณ์ความไวรัล

Q2. ใครเป็นผู้สนับสนุนการสร้างชุดข้อมูล งานวิจัยใช้ public API ของ YouTube (free tier 10,000 units/day) ไม่มี แหล่งทุนภายนอก ค่าใช้จ่าย cloud GPU ($0.49 USD) คณะผู้วิจัยรับผิดชอบเอง

Q3. ตอบช่องว่างใดในวรรณกรรม (1) ไม่มีงานก่อนหน้าที่เปรียบเทียบ Thai transformer encoders บนงาน virality prediction บนชุดข้อมูลเดียวกัน (2) ไม่มี Thai YouTube corpus สาธารณะที่ ออกแบบเพื่อการศึกษาความไวรัลแบบ *intra-channel*

D.2 องค์ประกอบ (Composition)

Q4. ตัวอย่างเป็นอะไร หนึ่งตัวอย่าง = หนึ่งวิดีโอ YouTube ภาษาไทย ระดับ public visibility ประกอบด้วยข้อความ (title, description, tags), meta-data ของช่อง, ตัวชี้วัด engagement (views, likes, comments) ณ snapshot date

Q5. มีกี่ตัวอย่าง 23,431 วิดีโอ จาก 82 ช่อง เผยแพร่ระหว่าง พ.ศ. 2562–2569 (10 ปี) ขนาดทางสถิติ: ตัวอย่างต่อช่องเฉลี่ย = 285 (ต่ำสุด 12, สูงสุด 1,847)

Q6. แต่ละตัวอย่างประกอบด้วยข้อมูลใด ดูตารางที่ 3.2 ในบทที่ 3 (15 ฟิลด์หลัก) หลังการแปลงเป็นฟีเจอร์ ผลคือ 27 structural + 6 sentiment + 768 embedding + 20,000 TF-IDF dimensions (ใช้แยกตามชนิดของแบบจำลอง)

Q7. มี label หรือ target หรือไม่ มี label เป็นไบนารี $y_i \in \{0, 1\}$ คำนวณจากดัชนี *VI* ในบทที่ 3 §3.3.2 เกณฑ์ = top-decile per channel

Q8. มีข้อมูลที่หายหรือไม่ มี description = "" ใน ∼8% ของวิดีโอ tags = [] ใน ∼15% มีการ impute เป็นสตริงว่างก่อนสร้างฟีเจอร์ ส่วน view_count = NULL ถูกตัดออกในระยะการ clean

Q9. ความสัมพันธ์ระหว่างตัวอย่างมีหรือไม่ มี วิดีโอที่อยู่ในช่องเดียวกันมีความสัมพันธ์เชิงโครงสร้าง (รูปแบบภาษา ของผู้สร้าง audience overlap, recommendation algorithm bias) คณะผู้วิจัย จัดการความสัมพันธ์นี้ผ่าน channel-grouped split (ดู §3.4)

Q10. มี split ที่แนะนำหรือไม่ ใช่ — train (6,964) / val (3,143) / test (3,510) แบบ channel-grouped + time-aware (§3.4) split files อยู่ใน data/processed/dataset_with_labels.parquet โดยฟิลด์ split

Q11. มี noise / errors ที่ทราบหรือไม่ (1) Unicode mojibake ในชื่อวิดีโอบางส่วนของช่องเก่า (พบ <1%) (2) จำนวน like_count อาจไม่ตรงกับช่วงเวลาของ view_count เพราะ API ส่ง snapshot ณ เวลาที่ query แต่ไม่มีการ versioning

Q12. ชุดข้อมูล self-contained หรือพึ่ง resource ภายนอก ครบสมบูรณ์ในไฟล์ parquet เดียว ไม่ต้องการ resource ภายนอก ยกเว้น ฟีเจอร์ที่สกัดหลังจากนั้น (sentiment cache + title embeddings) ซึ่งสามารถ recompute ได้จาก raw text

Q13. มีข้อมูลที่อาจถือว่าเป็น secret หรือไม่ ไม่มี. ทุกฟิลด์ถูกเปิดเผยต่อสาธารณะผ่าน YouTube web UI

Q14. มีข้อมูลที่อาจสร้างความรู้สึกไม่สบายใจให้ผู้อ่านหรือไม่ ในชื่อวิดีโอบางส่วนของหมวด Entertainment มีคำหยาบหรือคำที่กระตุ้นอารมณ์ ในเชิงลบ คณะผู้วิจัยไม่ได้กรองออก เพราะคำเหล่านี้เป็นส่วนหนึ่งของ phenomenon ที่ศึกษา

D.3 การเก็บรวบรวม (Collection)

Q15. กระบวนการเก็บข้อมูล ใช้ src/data_collection/youtube_collector.py เรียก YouTube Data API v3 endpoint videos.list, channels.list, search.list โดย maxResults=50 ต่อ request

Q16. ใครเก็บข้อมูล คณะผู้วิจัยเอง ระหว่าง พ.ค. 2569

Q17. การจ่ายเงินตอบแทนผู้เข้าร่วม ไม่เกี่ยวข้อง (ข้อมูลสาธารณะ ไม่มี human subject)

Q18. ระยะเวลาการเก็บ 24 ชั่วโมง (single snapshot) ในวันที่ 30 เมษายน 2569

Q19. การพิจารณาทางจริยธรรม / IRB ไม่เข้าข่าย IRB เพราะ (1) ไม่มี human subject (2) ใช้ข้อมูลสาธารณะ (3) ไม่มี PII คณะผู้วิจัยปฏิบัติตาม YouTube API Services Terms of Service ทุกประการ

D.4 การ Preprocessing / Labeling / Cleaning

Q20. การ preprocess ที่ทำ ดู §3.3 (clean) และ §3.5 (feature extraction) ทุกขั้นตอนใน src/data_processing/ และ src/features/

Q21. ข้อมูลดิบยังถูกเก็บไว้หรือไม่ **ใช่** data/raw/youtube_thai_videos.parquet ถูกเก็บไว้ในเครื่องของคณะผู้วิจัย และในที่เก็บข้อมูลส่วนตัว (private dataset repository) ของคณะผู้วิจัยและอาจารย์ที่ปรึกษา

Q22. software ที่ใช้ preprocess Python 3.11, pandas 2.x, pyarrow 15.x, PyThaiNLP 5.x ทั้งหมดเป็น open-source

D.5 การใช้งาน (Uses)

Q23. มีงานวิจัยอื่นที่ใช้ชุดข้อมูลนี้แล้วหรือยัง ยังไม่มี ปัญหาพิเศษนี้เป็นการใช้งานครั้งแรก

Q24. ชุดข้อมูลถูกใช้ในงานประเภทใดได้บ้าง นอกจาก virality prediction แล้ว สามารถใช้ในงาน (1) Thai title generation/rewriting (2) channel-level engagement modelling (3) Thai sentiment-engagement correlation studies

Q25. มีการใช้งานที่ *ไม่ควร* ทำหรือไม่ ไม่ควร ใช้เพื่อ (1) เปิดเผยข้อมูลรายบุคคลของผู้สร้างเนื้อหา (channel_id เปิดเผย แต่ไม่ควร นำไปจับคู่กับข้อมูลส่วนตัว) (2) สร้างเครื่องมือ *predict-and-attack* ช่อง competitor (3) optimisation ที่ละเมิด YouTube TOS (เช่น view-count manipulation)

D.6 การจัดเก็บและการแจกจ่าย (Distribution)

Q26. ชุดข้อมูลจะแจกจ่ายให้บุคคลที่สามหรือไม่ เฉพาะคำขอเชิงวิชาการ เพื่อปฏิบัติตาม YouTube TOS Section II.B.3 ที่ห้าม redistributable ของ Content เชิงพาณิชย์ ผู้ขอใช้ต้องลงนามใน data use agreement และมี institutional affiliation

Q27. แจกจ่ายผ่านช่องทางใด ผ่านที่เก็บข้อมูลส่วนตัวของคณะผู้วิจัยและอาจารย์ที่ปรึกษา โดยให้สิทธิ์ เป็นรายบุคคลตามเงื่อนไขข้อ Q26

Q28. มี IP หรือสัญญาเงื่อนไขการใช้หรือไม่ ผู้ขอใช้ต้องอ่านและยอมรับ *YouTube API Services Terms of Service* ก่อน และตกลงไม่ redistribute ภายใต้เงื่อนไข non-commercial / academic-only

D.7 การบำรุงรักษา (Maintenance)

Q29. ใครรับผิดชอบดูแล / อัปเดต ระหว่างที่ปัญหาพิเศษอยู่ในการประเมิน — คณะผู้วิจัย หลังจากนั้น — อาจารย์ ที่ปรึกษา (ผศ. ดร. พรพิมล ชัยวุฒิศักดิ์)

Q30. มีกำหนดอัปเดตเวอร์ชันใหม่หรือไม่ ไม่มี ชุดข้อมูลถูก freeze ไว้ตามวันที่ snapshot สำหรับการ reproduce สถานะปัจจุบันอย่างถาวร

Q31. ติดต่อผู้ดูแลได้ที่ใด ผ่านอาจารย์ที่ปรึกษา (ผศ. ดร. พรพิมล ชัยวุฒิศักดิ์) หรือคณะผู้วิจัยทั้งสาม (นายตรัยภูรินท์ สืบสุวรรณสาร, นายธีรนนท์ ไชยลังกา, นายภาณุเดช ภู่โทสนธิ์) ผ่านช่องทางอีเมลของ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง