**Câu 7: Input, Output của phương pháp**

1. **Input**

* **GDELT Text Data:**
* **train\_data['text']:** Dữ liệu văn bản tin tức GDELT để training
* **train\_data['date']:** Timestamp tương ứng
* **test\_data['text']:** Dữ liệu văn bản để test/forecast
* **test\_data['date']:** Timestamp test period

1. **Output:**

* **Ensemble Predictions:** Ma trận dự đoán predictions shape (n\_days, 3) cho top-3 hot topics
* Performance Metrics:
* MAE, MSE, RMSE cho từng topic và overall
* Feature importance từ XGBoost models
* Hot Topics Identification: List các topic hot nhất với hotness scores
* Visualizations: Biểu đồ trend comparison và individual topic forecasts

Câu 8: Hàm mất mất hoặc bài toán tối ưu của phương pháp

1. Prophet Model - Minimize trend + seasonality + holiday effects errors

Loss\_Prophet = Σ(y\_actual - (trend + seasonality + holidays))²

1. XGBoost Model - Gradient Boosting objective với regularization

Loss\_XGBoost = Σ(y\_actual - y\_pred)² + λ₁·Σ|w| + λ₂·Σw² - với L1, L2 regularization terms

1. LSTM Model - Mean Squared Error cho sequential prediction

Loss\_LSTM = (1/n)·Σ(y\_actual - y\_pred)²

1. Ensemble Optimization - Weighted combination optimization

Loss\_Ensemble = Σ(y\_actual - (w₁·Prophet + w₂·XGBoost + w₃·LSTM))² - Subject to: w₁ + w₂ + w₃ = 1, wᵢ ≥ 0

1. Hot Topics Selection Optimization - Multi-criteria hotness score maximization

Hotness\_Score = 0.3·avg\_prob + 0.3·recent\_trend + 0.2·variance +

0.1·peak\_intensity + 0.1·dominance\_freq

Câu 9: Các bước thực hiện của phương pháp

CÁC BƯỚC THỰC THI PIPELINE:

BƯỚC 1: Tải và tiền xử lý dữ liệu

# Tải các bộ dữ liệu GDELT

train\_data, test\_data = load\_datasets\_fast()

# Tiền xử lý văn bản nhanh theo lô (batch)

processed\_texts = batch\_preprocess\_fast(texts)

BƯỚC 2: Mô hình hóa chủ đề & Chọn chủ đề nóng

# Vector hóa TF-IDF

vectorizer = TfidfVectorizer(max\_features=1500, ngram\_range=(1,2))

tfidf\_matrix = vectorizer.fit\_transform(processed\_texts)

# Trích xuất chủ đề bằng LDA

lda\_model = LatentDirichletAllocation(n\_components=10)

topic\_distributions = lda\_model.fit\_transform(tfidf\_matrix)

# Xác định các chủ đề nóng dựa trên nhiều tiêu chí

identify\_hot\_topics(topic\_distributions, dates) # Chọn top-3

BƯỚC 3: Tạo đặc trưng dữ liệu thời gian

# Tổng hợp theo ngày

daily\_data = df.groupby('date')[topic\_cols].mean()

# Tạo các đặc trưng:

# - Đặc trưng thời gian: thứ trong tuần, tháng, quý, có phải cuối tuần không

# - Đặc trưng trễ (lag): topic\_i\_lag\_1, topic\_i\_lag\_7

# - Trung bình trượt: topic\_i\_ma\_3, topic\_i\_ma\_7

# - Tương tác chéo: topic\_i × topic\_j

BƯỚC 4: Huấn luyện đa mô hình

# A. Huấn luyện mô hình Prophet (cho mỗi chủ đề nóng)

for topic\_idx in hot\_topics:

prophet\_model = Prophet(weekly\_seasonality=True)

prophet\_model.fit(prophet\_data)

# B. Huấn luyện mô hình XGBoost (cho các đặc trưng tương tác)

for topic\_idx in hot\_topics:

xgb\_model = XGBRegressor(n\_estimators=100, max\_depth=6)

xgb\_model.fit(X\_features, y\_target)

# C. Huấn luyện mô hình LSTM (cho chuỗi dữ liệu)

lstm\_model = Sequential([LSTM(24), Dense(3)])

lstm\_model.fit(sequences, targets)

BƯỚC 5: Dự báo bằng tổ hợp mô hình (Ensemble)

# Tạo dự báo riêng lẻ từ từng mô hình

prophet\_preds = [model.predict(future) for model in prophet\_models]

xgb\_preds = [model.predict(X\_test) for model in xgb\_models]

lstm\_preds = lstm\_model.predict(sequences\_test)

# Kết hợp dự báo bằng tổ hợp trọng số

final\_predictions = (0.4·prophet\_preds + 0.4·xgb\_preds + 0.2·lstm\_preds)

BƯỚC 6: Đánh giá & Phân tích

# Tính các chỉ số đánh giá

mae = mean\_absolute\_error(actuals, predictions)

mse = mean\_squared\_error(actuals, predictions)

# Phân tích tầm quan trọng của đặc trưng

analyze\_feature\_importance()

# Trực quan hóa và tạo báo cáo

plot\_ensemble\_results()

generate\_hot\_topics\_report()