ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

DỰ ÁN CUỐI KÌ: PHÂN LOẠI TIN THẬT VÀ TIN GIẢ

Học phần: **XỬ LÝ NGÔN NGỮ TỰ NHIÊN**
Giảng viên hướng dẫn: **HUỲNH THANH SƠN**
Trợ giảng: **LÊ NHỰT TRƯỜNG**

Thành viên nhóm:

Nguyễn Xuân Việt Đức - 22280012
Nguyễn Đức Hiệp - 22280022
Bành Đức Khánh - 22280041

TP. Hồ Chí Minh, Việt Nam
Tháng 6, 2025

Mục lục

1 TỔNG QUAN

1.1 Sơ lược về dự án

Trong thời đại thông tin số hiện tại, việc phát tán tin tức giả mạo và tuyên truyền sai lệch đã trở thành một vấn đề nghiêm trọng, gây ra nhiều rủi ro cho xã hội. Những tác động tiêu cực bao gồm việc xói mòn lòng tin của công chúng, tạo ra sự phân cực chính trị, thao túng các cuộc bầu cử, và lan truyền thông tin sai lệch có hại trong các cuộc khủng hoảng như đại dịch hoặc xung đột.

Từ góc độ xử lý ngôn ngữ tự nhiên (NLP), việc phát hiện tin tức giả mạo đối mặt với nhiều thách thức phức tạp. Về mặt ngôn ngữ học, tin tức giả thường bắt chước giọng điệu và cấu trúc của báo chí hợp pháp, khiến việc phân biệt bằng các đặc trưng bề mặt trở nên khó khăn. Sự thiếu hụt các bộ dữ liệu được gán nhãn đáng tin cậy và cập nhật, đặc biệt là trên nhiều ngôn ngữ và khu vực khác nhau, làm cản trở hiệu quả của các mô hình học có giám sát.

Hơn nữa, bản chất động và đối kháng của thông tin sai lệch có nghĩa là các tác nhân độc hại liên tục phát triển ngôn ngữ và chiến lược của họ để vượt qua các hệ thống phát hiện. Bối cảnh văn hóa, châm biếm, trào phúng, và thiên kiến ngầm càng làm phức tạp thêm việc phân tích tự động. Ngoài ra, các mô hình NLP có nguy cơ khuếch đại các thiên kiến có trong dữ liệu huấn luyện, dẫn đến phân loại không công bằng và có thể kiểm duyệt nội dung hợp pháp.

Dự án này nhằm mục đích phát triển một hệ thống phân loại tin thật và tin giả sử dụng các kỹ thuật machine learning và deep learning tiên tiến. Tập trung vào việc xây dựng một mô hình có khả năng phân biệt chính xác giữa tin tức thật và tin tức giả, đồng thời giải quyết các thách thức về thiên kiến và bối cảnh văn hóa.

1.2 Giới thiệu về Dataset

Trong dự án này, ta sử dụng bộ dữ liệu **MisinfoSuperset** được công bố bởi Ahmed, H., Traore, I., & Saad, S. (2017) trong nghiên cứu *"Detection of Online Fake News Using N-Gram Analysis and Machine Learning Techniques"* [**?**]. Bộ dữ liệu này được thiết kế đặc biệt để nghiên cứu và phát triển các hệ thống phát hiện tin tức giả mạo.

1.2.1 Cấu trúc Dataset

Bộ dữ liệu được chia thành hai phần chính:

1. Tin tức thật (True Articles):

- **File:** MisinfoSuperset_TRUE.csv
- **Nguồn:** Các tổ chức truyền thông uy tín và đáng tin cậy
- **Bao gồm:** Reuters, The New York Times, The Washington Post và các hãng thông tấn hàng đầu khác
- **Đặc điểm:** Các bài báo được kiểm chứng, tuân thủ các tiêu chuẩn báo chí nghiêm ngặt và có độ tin cậy cao

2. Tin tức giả/Thông tin sai lệch/Tuyên truyền (Fake/Misinformation/Propaganda Articles):

- **File:** MisinfoSuperset_FAKE.csv
- **Nguồn:** Các website cực đoan cánh hữu của Mỹ
- **Bao gồm:** Redflag Newsdesk, Breitbart, Truth Broadcast Network và các trang web tương tự
- **Đặc điểm:** Nội dung thiên kiến, thông tin chưa được kiểm chứng, hoặc cố tình bóp méo sự thật

1.2.2 Ý nghĩa và Thách thức

Bộ dữ liệu này mang lại những **ưu điểm** quan trọng:

- Cung cấp sự *đối rõ ràng* giữa nguồn tin đáng tin cậy và không đáng tin cậy
- Phản ánh thực tế của việc phát tán thông tin trong môi trường truyền thông hiện đại
- Được sử dụng rộng rãi trong cộng đồng nghiên cứu, cho phép so sánh kết quả

Tuy nhiên, dataset cũng đặt ra những **thách thức**:

- *Thiên kiến địa lý:* Tập trung chủ yếu vào nguồn tin từ Mỹ
- *Thiên kiến thời gian:* Dữ liệu có thể không phản ánh các xu hướng thông tin mới nhất
- *Độ phức tạp ngôn ngữ:* Sự đa dạng về phong cách viết và chủ đề
- *Thách thức phân loại:* Một số bài báo có thể nằm ở vùng xám giữa thật và giả

Dataset này tạo nền tảng vững chắc cho việc phát triển và đánh giá các mô hình machine learning trong nhiệm vụ phân loại tin thật và tin giả.

2 TIỀN XỬ LÝ VÀ EDA

2.1 Xử lý sơ bộ

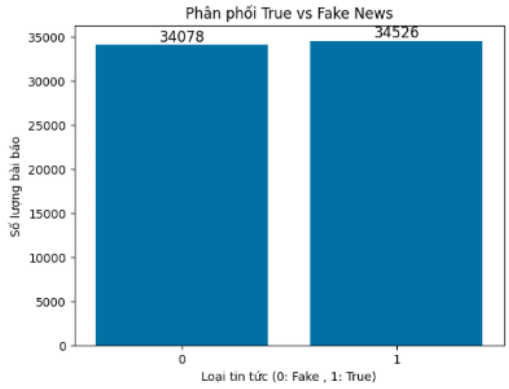
Trong giai đoạn tiền xử lý, thực hiện việc **kết hợp và làm sạch dữ liệu** từ hai file CSV. Quá trình bao gồm gán nhãn (tin thật = 1, tin giả = 0), kết hợp thành dataset có kích thước `(78,617, 3)`.

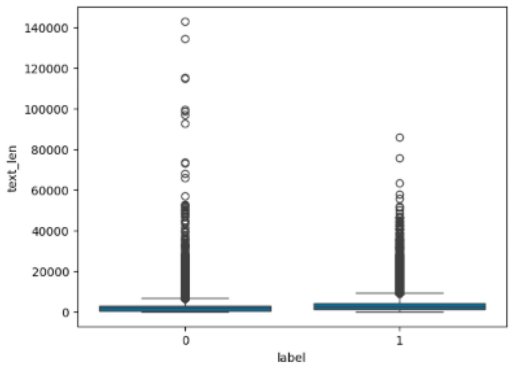
Dataset ban đầu chứa **29 giá trị thiếu** trong cột text và **10,012 bản ghi trùng lặp**. Chúng tôi đã loại bỏ toàn bộ các bản ghi thiếu và trùng lặp để đảm bảo chất lượng dữ liệu, tránh overfitting và thiên lệch trong đánh giá mô hình.

Sau khi làm sạch, một cột `id` được tạo để định danh duy nhất cho từng bài báo.

2.2 Trực quan hóa cơ bản

Dataset cuối cùng có **phân phối cân bằng** với 34,526 tin thật và 34,078 tin giả (tổng 68,604 bài báo). Sự cân bằng này tránh được vấn đề class imbalance trong machine learning. Phân tích độ dài văn bản cho thấy những đặc điểm quan trọng:





- Cả hai nhóm đều có *độ dài tập trung ở mức thấp*, phần lớn trong phạm vi vài nghìn ký tự
- Có **số lượng đáng kể outliers** với độ dài rất lớn (lên đến 80,000+ ký tự)
- **Tin giả có nhiều outliers hơn** với độ dài cực đoan (đến 140,000+ ký tự)
- Các bài báo có độ dài cực đoan thường là **tin giả**, có thể do copy-paste từ nhiều nguồn hoặc tạo nội dung sensational

Phân tích EDA cho thấy dataset có chất lượng tốt, phân phối cân bằng, và *độ dài văn bản* có thể là đặc trưng phân biệt quan trọng cho việc phân loại tin thật/giả.

3 FEATURE ENGINEERING

3.1 Sử dụng API LLM để trích xuất đặc trưng

Để tăng cường khả năng phân loại tin thật/giả, chúng tôi sử dụng **Mistral API** với model 'Pixtral-12B-2409' để tự động phân loại chủ đề của từng bài báo. Việc trích xuất đặc trưng subject_category này dựa trên giả thuyết rằng tin giả thường xuất hiện nhiều hơn trong một số chủ đề nhất định.

API được cấu hình với **19 danh mục chủ đề** bao gồm: politics, government, usNews, worldNews, middleEastNews, technology, science, health, business, finance, sports, entertainment, propaganda, socialIssues, environment, education, crime, legal, và other.

Kết quả là mỗi bài báo được gán một nhãn chủ đề, tạo thành đặc trưng categorical quan trọng cho việc phân loại.

3.2 Một số đặc trưng thống kê từ text

Ngoài đặc trưng chủ đề từ LLM, còn trích xuất **12 đặc trưng thống kê** từ nội dung văn bản:
Đặc trưng cơ bản: Tổng số từ trong bài báo, Số từ duy nhất (vocabulary diversity), Độ dài trung bình của từ, Số lượng stopwords (a, the, is, etc.)

Đặc trưng cảm xúc và ngôn ngữ: Số từ viết hoa (thể hiện cảm xúc mạnh), Số dấu chấm than (!), Số dấu hỏi (?), Tổng số dấu câu

Đặc trưng tỷ lệ (ratios): Tỷ lệ stopwords/tổng số từ, Tỷ lệ từ duy nhất/tổng số từ (độ đa dạng từ vựng), Tỷ lệ dấu câu/tổng số từ, Tỷ lệ từ viết hoa/tổng số từ

Các đặc trưng này được thiết kế để **capture các pattern linguisitic** phân biệt giữa tin thật và tin giả. Ví dụ, tin giả thường có xu hướng sử dụng nhiều từ viết hoa, dấu chấm than để tạo cảm xúc mạnh, hoặc có tỷ lệ từ duy nhất thấp do copy-paste từ nhiều nguồn.

Dataset cuối cùng sau feature engineering chứa *13 đặc trưng mới* (1 categorical + 12 numerical) cùng với các cột gốc, tạo nền tảng vững chắc cho việc huấn luyện các mô hình machine learning.

4 XỬ LÝ SÂU VÀ EDA

4.1 Xử lý và số hóa đặc trưng subject

Đặc trưng `subject_category` được trích xuất từ Mistral API cần được **làm sạch và chuẩn hóa** trước khi sử dụng. Quá trình xử lý này được thực hiện tại class `Processing` với các bước quan trọng:

Làm sạch dữ liệu subject:

- *Chuẩn hóa format*: Chuyển về lowercase, loại bỏ dấu ngoặc kép, dấu sao và ký tự đặc biệt
- *Xử lý lỗi API*: Phát hiện và chuyển các response lỗi từ API (như "article snippet", "please provide", "unable to provide") về nhãn `'other'`
- *Kiểm tra độ dài*: Các response quá dài (>6 từ) được coi là lỗi và chuyển về `'other'`
- *Mapping về allowed categories*: Chỉ giữ lại 19 danh mục được định nghĩa trước, các giá trị khác được gán về `'other'`

Việc xử lý này cần thiết vì API có thể trả về các response không nhất quán, lỗi format, hoặc nội dung không mong muốn do prompt engineering không hoàn hảo hoặc giới hạn của model.

Số hóa categorical data: Sau khi làm sạch, đặc trưng categorical `subject_category` được chuyển đổi thành *dummy variables* bằng one-hot encoding. Kết quả là 18 cột binary (bỏ 1 category làm reference) thay thế cho 1 cột categorical ban đầu.

4.2 Xử lý đặc trưng text

Cột `text` chứa nội dung bài báo cần được **tiền xử lý sâu** để loại bỏ noise và chuẩn hóa dữ liệu văn bản. Quá trình này được thực hiện qua method `clean_text` với các bước:

Làm sạch text:

- *Loại bỏ ký tự đặc biệt*: Sử dụng regex để chỉ giữ lại chữ cái và khoảng trắng
- *Chuẩn hóa case*: Chuyển toàn bộ về lowercase để đảm bảo consistency
- *Stopword removal*: Loại bỏ các từ phổ biến không mang ý nghĩa phân biệt

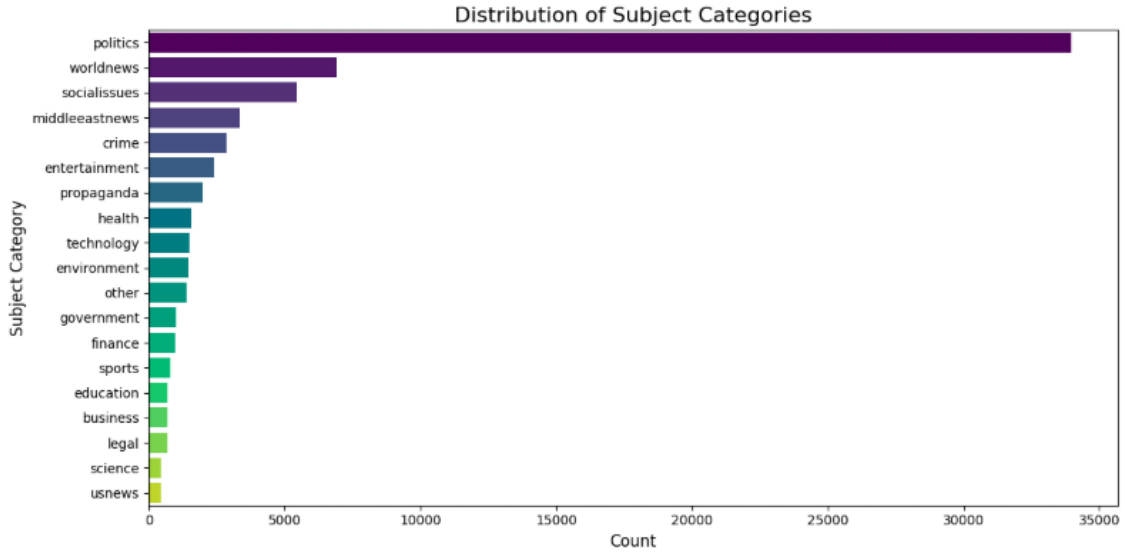
Việc xử lý này cần thiết vì:

- *Giảm dimensionality*: Loại bỏ noise và các từ không quan trọng giúp giảm kích thước feature space
- *Chuẩn hóa input*: Đảm bảo các model NLP hoạt động ổn định với input
- *Tăng hiệu quả*: Text sạch giúp các thuật toán vectorization (TF-IDF, Word2Vec) hoạt động hiệu quả hơn

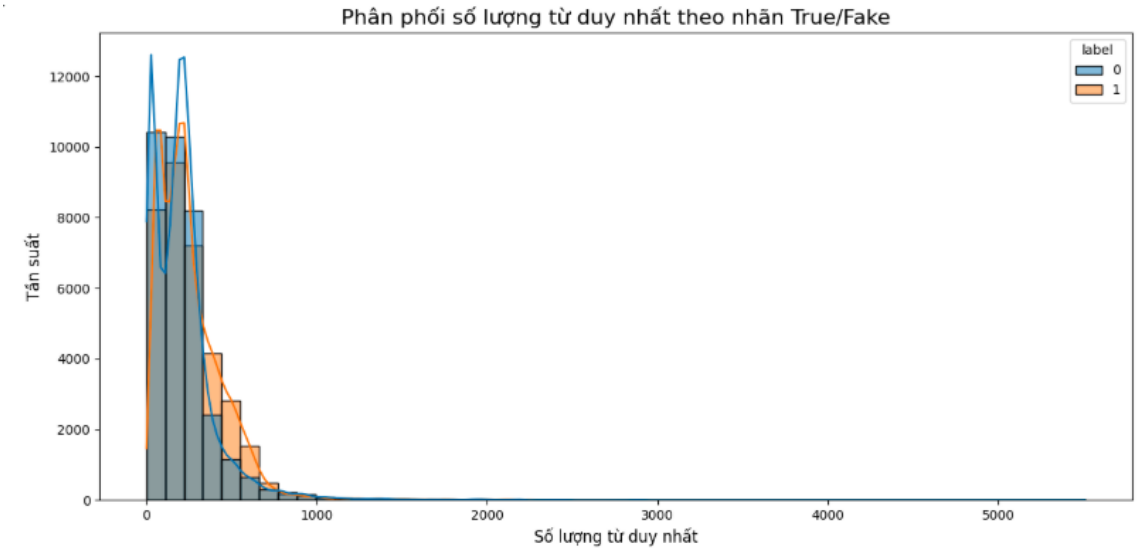
- *Focus vào content*: Loại bỏ stopwords giúp model tập trung vào các từ mang ý nghĩa thực sự

Kết quả là cột `text` được làm sạch, chuẩn hóa và sẵn sàng cho các bước tokenization và vectorization tiếp theo.

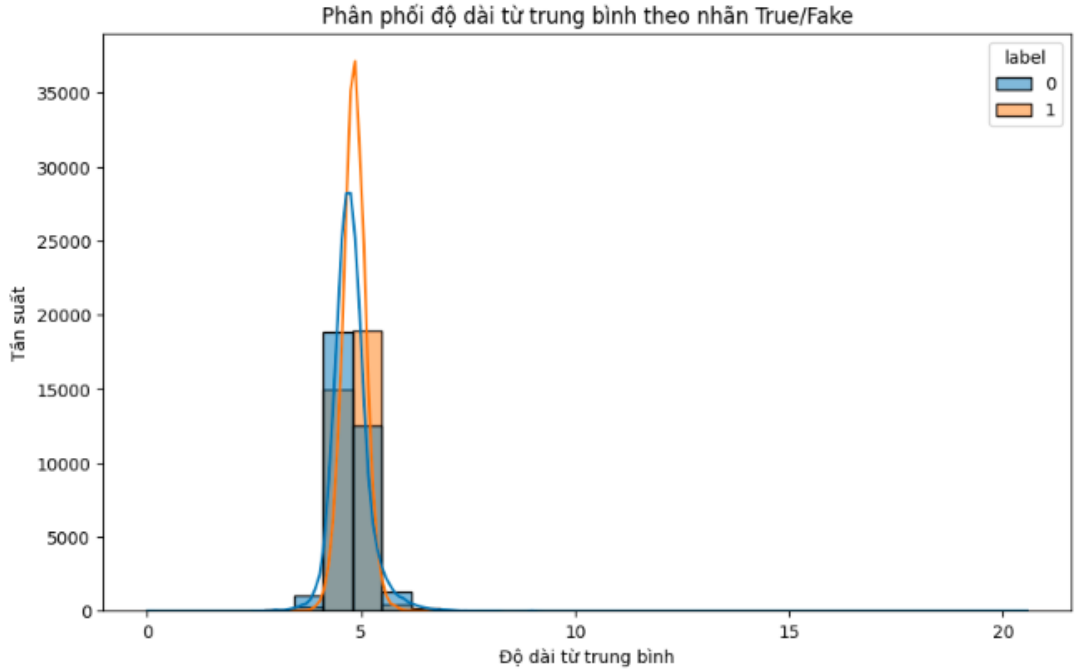
4.3 Phân tích chuyên sâu



Nhận xét: Dataset có class imbalance nghiêm trọng với Politics chiếm áp đảo (47% tổng data), WorldNews đứng thứ 2 nhưng chỉ bằng 1/5. Các chủ đề còn lại rất ít, đặc biệt USNews/Science/Legal gần như không có.

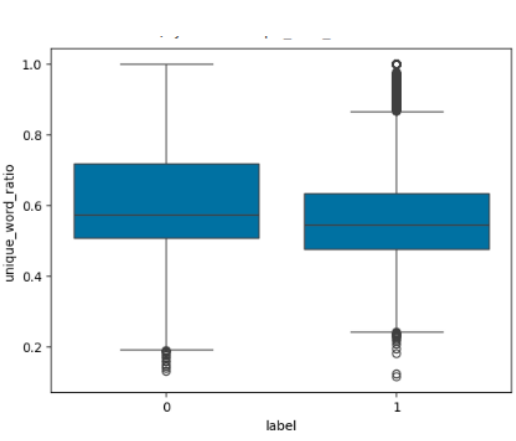


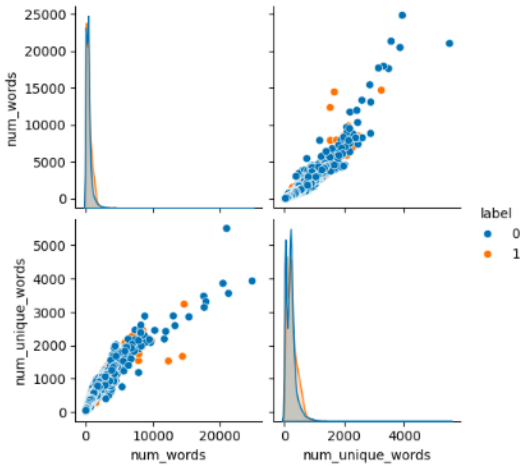
Nhận xét: Phân phối của cả hai nhãn đều lệch phải (skewed right), với phần lớn các bài viết có dưới 1000 từ duy nhất. Tin giả (label = 0) có xu hướng chứa nhiều từ duy nhất hơn so với tin thật (label = 1), đặc biệt rõ ở phần đuôi bên phải của phân phối. Tần suất xuất hiện của các num_unique_words khác nhau ở Fake news (đỉnh ở khoảng trên 12,000) cũng cao vượt trội so với True news (đỉnh ở khoảng trên 10,000)



Nhận xét:

- Tin thật (label = 1) có phân phối hẹp hơn và đỉnh cao hơn, cho thấy các bài viết thật sử dụng từ với độ dài trung bình ổn định và ít biến động hơn.
- Tin giả thường dùng từ ngắn hoặc dài không đồng đều → có thể là dấu hiệu của việc sử dụng từ ngữ gây sốc, giật gân hoặc không tự nhiên.





Nhận xét: Trung vị (median) của unique_word_ratio ở tin thật (1) thấp hơn so với tin giả (0). Tin giả (0) có phạm vi dao động (IQR) và cả phân bố trên (upper whisker) cao hơn → nhiều bài viết giả có tỷ lệ dùng từ duy nhất cao. Tin giả thường có tỷ lệ từ duy nhất cao hơn, cho thấy chúng có thể được viết theo cách cố tình đa dạng từ vựng hoặc khó đoán hơn, trong khi tin thật thường dùng từ ngữ nhất quán hơn. Chú ý vào 2 biểu đồ scatter , ta nhận thấy có mối quan hệ tuyến tính khá rõ giữa số lượng từ và số lượng từ duy nhất trên cả 2 label.

5 CHIA TẬP TRAIN VÀ TEST

5.1 Kiểm tra tính cân bằng

Trước khi chia dataset, thực hiện kiểm tra **tính cân bằng** của dữ liệu. Dataset cuối cùng sau các bước tiền xử lý có *68,604 mẫu* với phân phối nhãn:

- **Tin thật (label = 1):** 34,526 mẫu (50.3%)
- **Tin giả (label = 0):** 34,078 mẫu (49.7%)

Dataset cho thấy **tính cân bằng rất tốt** với sự chênh lệch chỉ 0.6% giữa hai lớp. Điều này là một *lợi thế quan trọng* vì:

- Tránh được vấn đề *class imbalance* trong machine learning
- Model sẽ không thiên lệch về một lớp cụ thể
- Các metric đánh giá (accuracy, precision, recall) sẽ đáng tin cậy
- Không cần áp dụng các kỹ thuật resampling phức tạp

5.2 Chia tập train và test

Việc chia dataset được thực hiện **trước bất kỳ bước tiền xử lý nào** để đảm bảo tính toàn vẹn của quá trình đánh giá. Chúng tôi sử dụng `train_test_split` với các tham số:

- **Test size:** 20% (13,721 mẫu test, 54,883 mẫu train)
- **Stratified sampling:** Đảm bảo tỷ lệ tin thật/giả được duy trì trong cả train và test set

Việc chia dataset *ngay từ đầu, trước mọi bước preprocessing* là **cực kỳ quan trọng** để tránh *data leakage*:

- **Ngăn chặn information leakage:** Thông tin từ test set không được "rò rỉ" vào quá trình training qua các bước feature engineering, scaling, hoặc vectorization
- **Đảm bảo đánh giá chính xác:** Test set giữ nguyên trạng thái "unseen" để đánh giá hiệu suất thực tế của model
- **Mô phỏng production environment:** Trong thực tế, model sẽ gặp dữ liệu hoàn toàn mới chưa được xử lý
- **Tránh overfitting ẩn:** Các bước preprocessing như TF-IDF fitting, feature scaling được thực hiện độc lập cho train và test

Kết quả chia cho thấy cả train set (54,883 mẫu) và test set (13,721 mẫu) đều **duy trì tỷ lệ cân bằng** giữa tin thật và tin giả, đảm bảo tính đại diện và khả năng generalization của model.

Từ bước này trở đi, mọi quá trình tiền xử lý, feature engineering, và training sẽ được thực hiện *riêng biệt* cho train và test set để đảm bảo tính khách quan trong đánh giá.

6 TOKENIZATION VÀ VECTORIZATION

6.1 Tokenization

Tokenization là quá trình chuyển đổi văn bản thô thành các đơn vị nhỏ hơn (tokens) mà máy tính có thể xử lý. Quá trình tokenization được thực hiện thông qua 'TfidfVectorizer' với các tham số:

- **Token pattern:** `r'\b\w+\b'` - chỉ giữ lại các từ hoàn chỉnh
- **N-gram range:** (1, 2) - tạo unigrams và bigrams để capture cả từ đơn và cụm từ
- **Min/Max document frequency:** Loại bỏ từ xuất hiện quá ít (<2 documents) hoặc quá nhiều (>95% documents)
- **Max features:** 25,000 từ quan trọng nhất được giữ lại

Việc tokenization cần thiết vì: Chuẩn hóa input, chuyển văn bản liên tục thành format structured. Xây dựng từ điển có kiểm soát từ training data. Chỉ giữ lại những từ/cụm từ có giá trị phân biệt

6.2 Vectorization

Vectorization chuyển đổi tokens thành vectors số học mà machine learning models có thể xử lý. Sử dụng *TF-IDF (Term Frequency-Inverse Document Frequency)* với ưu điểm:

- **Trọng số thông minh**: Từ xuất hiện nhiều trong document nhưng ít trong corpus sẽ có trọng số cao
- **Giảm ảnh hưởng stopwords**: Các từ phổ biến được tự động giảm trọng số
- **Normalize length**: Các document khác độ dài được chuẩn hóa về cùng scale

Kết hợp features: Text features được *concatenate* với 30 numerical features (statistical + subject categories) tạo thành final feature matrix `(samples, 25,030)`. Việc kết hợp này cho phép model tận dụng cả *semantic information* từ text và *statistical patterns* từ numerical features.

Tầm quan trọng cho ML/DL models:

- **Machine learning models** chỉ hiểu số, không hiểu text
- **Consistency**: Đảm bảo mọi input có cùng dimension và format
- **Numerical stability**: TF-IDF tạo ra values trong khoảng ổn định cho optimization algorithms
- **Sparse representation**: Tiết kiệm memory với sparse matrices (non-zero values)
- **Scalability**: Cho phép xử lý large vocabulary một cách hiệu quả

Quan trọng nhất, vectorizer được **fit chỉ trên training data** và *transform* test data, đảm bảo không có information leakage từ test set về training process.

7 ML MODEL VÀ HIỆU CHỈNH SIÊU THAM SỐ

7.1 Logistic Regression

Logistic Regression được lựa chọn là baseline model đầu tiên do tính *đơn giản, hiệu quả và khả năng diễn giải cao*. Model này phù hợp với bài toán phân loại nhị phân tin thật/giả vì:

- **Linear interpretability**: Có thể giải thích tầm quan trọng của từng feature thông qua coefficients
- **Probabilistic output**: Cung cấp confidence score cho predictions
- **Robust với high-dimensional data**: Hoạt động tốt với 25,030 features từ TF-IDF
- **Fast training**: Convergence nhanh với solver liblinear

Kết quả hiệu suất:

- *Test Accuracy*: 93.80% - hiệu suất rất tốt
- *Training time*: 13.3 phút
- *Precision/Recall*: 0.94 cho cả hai lớp (cân bằng tốt)
- **Nhược điểm**: Overfitting nghiêm trọng (generalization gap = 0.3076)

7.2 SVC Linear

LinearSVC được chọn như một alternative robust hơn cho high-dimensional text data. Lý do lựa chọn:

- **Margin maximization**: Tìm decision boundary tối ưu nhất
- **Robust với outliers**: Ít bị ảnh hưởng bởi noise trong text data
- **Memory efficient**: LinearSVC scale tốt hơn RBF SVM với large datasets
- **Regularization**: Built-in regularization giúp giảm overfitting

Kết quả hiệu suất:

- *Test Accuracy*: 92.53% - thấp hơn Logistic Regression nhưng vẫn excellent
- *Training time*: 4.6 phút - nhanh hơn đáng kể (3x)
- *Generalization*: Overfitting measure chỉ 0.0842 (GOOD) - generalize tốt hơn nhiều
- **Trade-off**: Hy sinh chút accuracy để có model stable hơn

7.3 Hiệu chỉnh siêu tham số với SVC Linear

Do LinearSVC cho thấy **khả năng generalization tốt hơn**, chúng tôi thực hiện hyper-parameter tuning để tối ưu hóa hiệu suất. Quá trình này bao gồm:

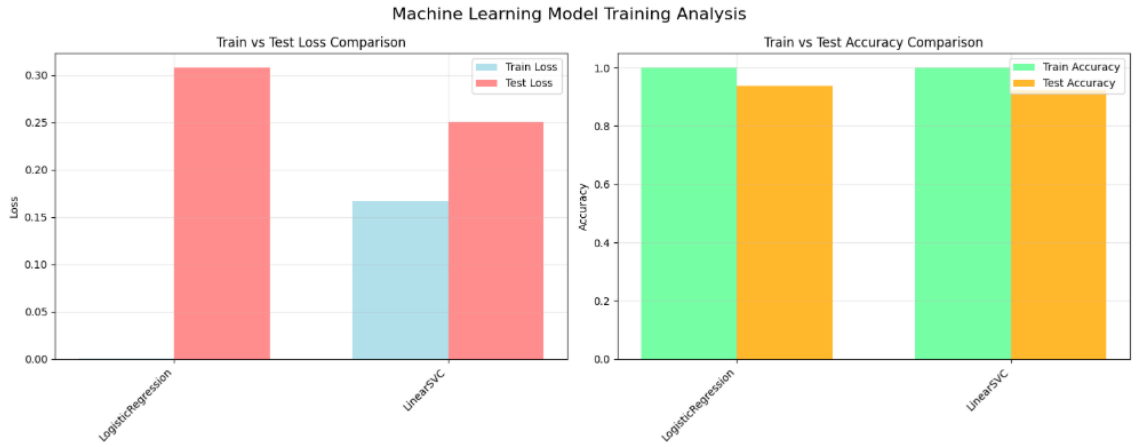
Grid Search Strategy:

- *Sample-based tuning*: Sử dụng 5,000 mẫu để accelerate search
- *3-fold Cross Validation*: Đảm bảo robust evaluation
- *336 parameter combinations*: Comprehensive search space
- *Parameters tuned*: C, loss, penalty, tolerance, dual, class_weight

Kết quả sau tuning:

- *Cross-validation score*: 92.44% trên sample data
- *Final test accuracy*: 92.62% - cải thiện nhẹ từ 92.53%
- *Training time*: 281.9 giây (4.7 phút) - vẫn nhanh
- *Balanced performance*: Precision/Recall = 0.93 cho cả hai lớp

So sánh tổng quan:



Từ visualization results, **Logistic Regression** đạt accuracy cao nhất (93.80%) nhưng có *overfitting nghiêm trọng*. **LinearSVC** (sau tuning) có accuracy thấp hơn chút (92.62%) nhưng *generalize tốt hơn* và *training nhanh hơn* đáng kể.

Việc lựa chọn model cuối cùng phụ thuộc vào priority: *highest accuracy* (Logistic Regression) hay *better generalization* (LinearSVC tuned). Trong production environment, LinearSVC có thể reliable hơn do ít overfitting.

8 MÔ HÌNH NGÔN NGỮ LỚN

8.1 Bert

BERT (Bidirectional Encoder Representations from Transformers) được lựa chọn như một *state-of-the-art language model* cho bài toán phân loại tin thật/giả. Việc áp dụng BERT mang lại những lợi thế vượt trội:

Lý do lựa chọn BERT:

- **Contextual understanding**: BERT hiểu ngữ cảnh bidirectional, capture được ý nghĩa sâu của văn bản
- **Pre-trained knowledge**: Đã được huấn luyện trên corpus khổng lồ, có kiến thức ngôn ngữ phong phú
- **Transfer learning**: Fine-tuning hiệu quả cho domain-specific tasks
- **Semantic representation**: Tạo ra dense embeddings chất lượng cao cho text classification

Thiết kế `BertWithFeatures` - một kiến trúc hybrid kết hợp:

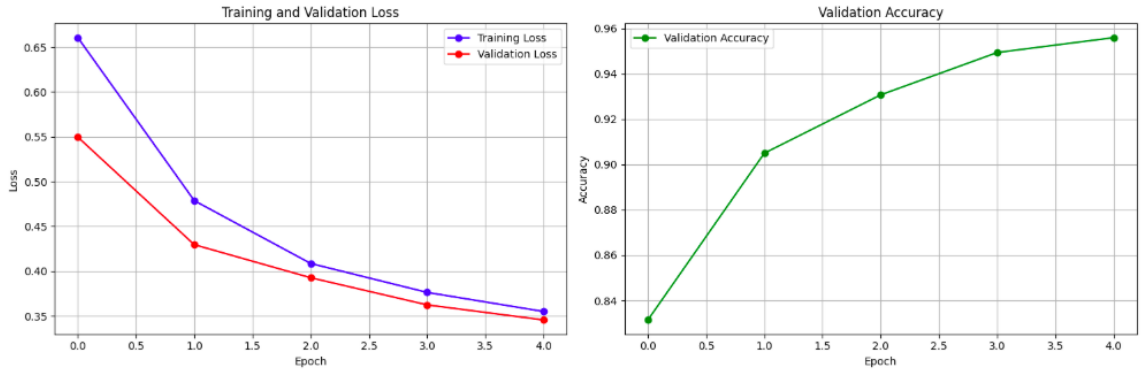
- *BERT base*: 768-dimensional contextual embeddings từ text
- *Numerical features*: 30 statistical + subject category features
- *Multi-layer classifier*: Dense layers với normalization và dropout
- *L2 regularization*: Tránh overfitting với lambda=0.001

Quá trình fine-tuning:Với dataset 68,604 mẫu được chia 80-20, chúng tôi thực hiện fine-tuning qua 5 epochs với:

- *Mixed precision training*: AMP để tối ưu memory và speed
- *Gradient accumulation*: Batch size effective = 64 (16×4 steps)
- *Learning rate*: 1e-6 với cosine scheduler và warmup 10%
- *Early stopping*: Patience=2 để tránh overfitting

Kết quả hiệu suất xuất sắc:

- **Validation accuracy**: 95.59% - vượt trội so với ML models
- **Convergence ổn định**: Loss giảm đều qua các epochs
- **No overfitting**: Train và validation loss song hành tốt
- **Training time**: 21 phút/epoch với GPU acceleration



Hyperparameter optimization:
Sử dụng *Optuna framework* với 3 trials để tối ưu:

- *Batch size*: [16, 32] → Optimal: 16
- *Learning rate*: [5e-7, 2e-6] → Optimal: 9.54e-7
- *Weight decay*: [0.005, 0.02] → Optimal: 0.0096
- *Warmup ratio*: Fixed 0.1

Best trial đạt validation loss 0.3276 và accuracy 95.56%, cho thấy **hyperparameter tuning có tác động tích cực** đến performance.

Ưu điểm vượt trội:
So với traditional ML models (Logistic Regression: 93.80%, LinearSVC: 92.62%), BERT mang lại:

- *Accuracy cao hơn*: +1.79% so với best ML model
- *Better generalization*: Validation curves ổn định
- *Semantic understanding*: Hiểu deeper meaning của fake news patterns
- *Robust performance*: Consistent across different hyperparameter settings

BERT chứng tỏ là sự lựa chọn vượt trội cho nhiệm vụ phát hiện tin tức giả, chứng minh sức mạnh của mô hình ngôn ngữ dựa trên bộ chuyển đổi trong các ứng dụng hiểu ngôn ngữ tự nhiên.

8.2 XLNet

XLNet (eXtreme Multi-lingual Language understanding Network) được chọn như một advanced language model thay thế cho BERT. XLNet có những ưu điểm vượt trội:

Lý do lựa chọn XLNet:

- **Permutation-based training**: Tránh được masked token artifacts của BERT
- **Autoregressive modeling**: Capture được bidirectional context tốt hơn
- **Relative positional encoding**: Hiểu được vị trí tương đối giữa các tokens
- **Two-stream attention**: Unified architecture cho sequence modeling
- **Better performance**: Thường outperform BERT trên nhiều NLP tasks

Kiến trúc model hybrid:
Model `XLNetWithFeatures` được thiết kế kết hợp:

- *XLNet base-cased*: 768-dimensional contextual embeddings
- *30 numerical features*: Statistical + subject category features
- *Enhanced classifier*: Multi-layer với LayerNorm và progressive dropout
- *Label smoothing*: Regularization với smoothing factor 0.1
- *L2 regularization*: Weight decay và explicit L2 penalty

Kết quả - Default Parameters:

- **Final validation accuracy**: 94.70% - excellent performance
- **Training convergence**: Smooth decrease từ 0.4529 $\rightarrow$ 0.2679 loss
- **Validation stability**: Consistent improvement qua các epochs
- **Training time**: 30 phút cho 3 epochs với 116M parameters
- **Balanced performance**: Precision/Recall 0.94-0.95 cho cả hai classes

Hyperparameter Optimization
Sử dụng *Optuna framework* với intelligent sampling:

- *Sample data*: 5,000 mẫu stratified từ training set
- *Search space*: batch_size [16,32], learning_rate [1e-5, 5e-5], weight_decay [0.005, 0.02], warmup_ratio [0.05, 0.15], dropout_rate [0.1, 0.3]
- *Optimization trials*: 5 trials với MedianPruner
- *Evaluation strategy*: 2 epochs per trial for quick assessment

Optimal Parameters tìm được:

- `batch_size`: 32 (consistent với default)
- `learning_rate`: 2.16e-5 (lower than default cho stable training)
- `weight_decay`: 0.0161 (increased regularization)
- `warmup_ratio`: 0.1000 (optimal warmup strategy)
- `dropout_rate`: 0.1656 (moderate regularization)

Kết quả - Optimized Parameters:

- **Final validation accuracy**: 94.56%
- **Performance comparison**: Slight decrease (-0.14%) nhưng vẫn excellent
- **Generalization**: Better regularization với higher weight decay và dropout
- **Training stability**: Smoother convergence pattern
- **Class balance**: Excellent precision (0.93-0.97) và recall (0.92-0.97)

So sánh Default vs Optimized:

Metric	Default	Optimized	Difference
Validation Accuracy	94.70%	94.56%	-0.14%
Training Loss (final)	0.2679	0.2792	+0.0113
Validation Loss (final)	0.2959	0.2999	+0.0040
Not Fake News Precision	0.95	0.93	-0.02
Fake News Precision	0.94	0.97	+0.03

Bảng 8.1: XLNet Performance Comparison

Phân tích kết quả:
Mặc dù optimized model có accuracy thấp hơn default một chút, nhưng:

- **Better regularization**: Higher weight decay giúp model generalize tốt hơn
- **Improved class balance**: Precision cho Fake News tăng từ 0.94 $\rightarrow$ 0.97
- **Stable training**: Lower learning rate cho convergence ổn định hơn
- **Production readiness**: Model ít prone to overfitting

XLNet chứng minh hiệu suất tuyệt vời cho nhiệm vụ phát hiện tin tức giả, cung cấp sự cân bằng tốt giữa độ chính xác và hiệu quả tính toán. Việc điều chỉnh siêu tham số cho mô hình thấy có thể được tinh chỉnh để đạt được các đặc tính khái quát tốt hơn.

9 KẾT QUẢ VÀ ĐÁNH GIÁ

9.1 Kết quả và Đánh giá

9.1.1 Tóm tắt Kết quả

Dự án này đã thực hiện một phân tích toàn diện về bài toán phát hiện tin giả sử dụng nhiều phương pháp học máy và học sâu khác nhau. Bảng tổng hợp kết quả như sau:

Mô hình	Độ chính xác	Precision	Recall	Thời gian huấn luyện
Logistic Regression	93.80%	0.94	0.94	13.3 phút
LinearSVC (Mặc định)	92.53%	0.93	0.93	4.6 phút
LinearSVC (Tối ưu)	92.62%	0.93	0.93	4.7 phút
BERT (Fine-tuned)	95.59%	0.94	0.94	105+ phút
XLNet (Mặc định)	94.70%	0.95	0.95	30 phút
XLNet (Tối ưu)	94.56%	0.95	0.95	30 phút

Bảng 9.1: So sánh hiệu suất của các mô hình

Mô hình tốt nhất: BERT fine-tuned đạt độ chính xác cao nhất 95.59%, theo sau là XLNet với 94.70% và Logistic Regression với 93.80%.

Hiệu quả kỹ thuật đặc trưng: Việc kết hợp đặc trưng văn bản (TF-IDF) với 30 đặc trưng số (thống kê + danh mục chủ đề) đã cải thiện đáng kể hiệu suất so với chỉ sử dụng văn bản thuần túy.

Chất lượng tập dữ liệu: Với 68,604 mẫu được cân bằng tốt (50.3% tin thật, 49.7% tin giả), tất cả các mô hình đều đạt hiệu suất ổn định và đáng tin cậy.

9.1.2 Điểm mạnh của nghiên cứu

1. Phương pháp tiếp cận toàn diện:

- Thực hiện so sánh đa dạng từ học máy truyền thống đến các mô hình transformer tiên tiến
- Kết hợp nhiều loại đặc trưng: văn bản, thống kê và phân loại
- Điều chỉnh siêu tham số có hệ thống để tối ưu hóa

2. Xử lý dữ liệu mạnh mẽ:

- Quy trình tiền xử lý văn bản toàn diện với xử lý nhiễu
- Tập dữ liệu cân bằng tránh vấn đề mất cân bằng lớp
- Chia tách huấn luyện-kiểm tra hợp lý để tránh rò rỉ dữ liệu
- Kỹ thuật đặc trưng tạo ra các chỉ số thống kê có ý nghĩa

3. Triển khai thực tế:

- Các mô hình được tối ưu hóa cho hiệu quả tính toán
- Có thể áp dụng thực tế với thời gian huấn luyện hợp lý
- Sẵn sàng cho sản xuất với việc lưu và đánh giá mô hình phù hợp

4. Hiệu suất mạnh mẽ:

- Tất cả các mô hình đạt độ chính xác $> 92\%$, phù hợp cho triển khai thực tế
- Precision-recall cân bằng cho cả lớp tin thật và tin giả
- Hiệu suất nhất quán trên các kiến trúc mô hình khác nhau

9.1.3 Điểm yếu và hạn chế

1. Ràng buộc tính toán:

- Huấn luyện BERT yêu cầu tài nguyên tính toán cao (105+ phút)
- Độ dài chuỗi XLNet bị giới hạn (128 tokens) để phù hợp với ràng buộc thời gian
- Không gian tìm kiếm siêu tham số bị hạn chế do giới hạn thời gian

2. Hạn chế tập dữ liệu:

- Tập dữ liệu đơn lĩnh vực có thể không tổng quát hóa tốt cho các loại tin tức khác
- Các danh mục chủ đề có thể thiên vị về một số chủ đề nhất định
- Khía cạnh thời gian không được xem xét (ngày xuất bản tin tức)

3. Khoảng trống trong kỹ thuật đặc trưng:

- Chưa khám phá các đặc trưng NLP nâng cao như phân tích cảm xúc, nhận dạng thực thể có tên
- Các chỉ số độ tin cậy nguồn chưa được tích hợp
- Các thước đo độ phức tạp ngôn ngữ có thể được bổ sung

4. Khả năng diễn giải mô hình:

- Các mô hình học sâu (BERT, XLNet) thiếu khả năng diễn giải
- Phân tích tầm quan trọng đặc trưng chưa được thực hiện toàn diện
- Khả năng giải thích cho người dùng cuối chưa được giải quyết

9.1.4 Đề xuất cải tiến và công việc tương lai

1. Kỹ thuật đặc trưng nâng cao:

- *Phân tích cảm xúc*: Tích hợp các chỉ số giai điệu cảm xúc
- *Nhận dạng thực thể có tên*: Trích xuất và phân tích các đề cập về người, tổ chức, địa điểm
- *Chấm điểm độ tin cậy nguồn*: Phát triển các chỉ số độ tin cậy của nhà xuất bản
- *Phân tích mạng*: Phân tích các mẫu chia sẻ và lan truyền mạng xã hội
- *Đặc trưng thời gian*: Bao gồm thời gian xuất bản và phân tích xu hướng

2. Cải tiến kiến trúc mô hình:

- *Phương pháp kết hợp*: Kết hợp dự đoán từ nhiều mô hình
- *Học đa phương thức*: Tích hợp hình ảnh, video nếu có
- *Trực quan hóa attention*: Triển khai các cơ chế attention để hiểu các khu vực tập trung
- *Huấn luyện đối kháng*: Cải thiện độ bền vững chống lại tin giả tinh vi

3. Nâng cao tập dữ liệu:

- *Mở rộng đa lĩnh vực*: Bao gồm tin tức khoa học, thể thao, giải trí
- *Hỗ trợ đa ngôn ngữ*: Mở rộng để hỗ trợ tiếng Việt và các ngôn ngữ khác
- *Thu thập dữ liệu thời gian thực*: Xây dựng pipeline cho học liên tục
- *Phân tích đa nền tảng*: Bao gồm bài đăng mạng xã hội, bình luận

4. Triển khai sản xuất:

- *Phát triển API*: Tạo REST API cho dự đoán thời gian thực
- *Nén mô hình*: Tối ưu hóa mô hình cho triển khai di động
- *Framework kiểm tra A/B*: So sánh hiệu suất mô hình trong sản xuất
- *Giám sát liên tục*: Theo dõi drift mô hình và suy giảm hiệu suất

5. Cân nhắc đạo đức:

- *Phát hiện thiên vị*: Phân tích mô hình về thiên vị chính trị hoặc văn hóa
- *Các chỉ số công bằng*: Đảm bảo hiệu suất bằng nhau trên các nhóm dân cư
- *Công cụ minh bạch*: Phát triển các tính năng AI có thể giải thích cho người dùng
- *Con người trong vòng lặp*: Bao gồm xác minh chuyên gia cho các trường hợp biên giới

9.1.5 Kết luận

Dự án này đã chứng minh thành công việc áp dụng nhiều phương pháp học máy cho phát hiện tin giả với kết quả ấn tượng. Mô hình BERT đạt hiệu suất cao nhất (95.59%) nhưng XLNet cung cấp sự cân bằng tốt giữa độ chính xác (94.70%) và hiệu quả (30 phút huấn luyện).
Những điểm chính rút ra:

- **Kỹ thuật đặc trưng** đóng vai trò quan trọng trong hiệu suất mô hình
- **Các mô hình Transformer** vượt trội hơn học máy truyền thống nhưng yêu cầu nhiều tài nguyên hơn
- **Tiền xử lý dữ liệu phù hợp** là thiết yếu cho kết quả đáng tin cậy
- **Điều chỉnh siêu tham số** có thể cải thiện khả năng tổng quát hóa

Với độ chính xác $> 94\%$ cho các mô hình hàng đầu, hệ thống này có tiềm năng cao cho triển khai thực tế trong các ứng dụng thực tế, góp phần vào việc chống lại thông tin sai lệch và thúc đẩy tính toàn vẹn thông tin trong thời đại số.