

# TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA TOÁN – TIN HỌC



## BÁO CÁO FINAL PROJECT

Đề tài: Phát Hiện Tin Giả Dựa Trên Học Máy và Học Sâu

### Thành viên nhóm

Họ và tên	Mã số sinh viên
Lương Thanh Nam	22280056
Lê Trọng Nghĩa	22280059
Phạm Tấn Phước	22280069

<b>Chi tiết đề tài.....</b>	<b>2</b>
<b>Phần 1: Tiền xử lý dữ liệu .....</b>	<b>3</b>
1. Đọc và Gộp dữ liệu .....	3
2. Làm sạch dữ liệu .....	3
3. Kết quả .....	3
<b>Phần 2: Khám phá dữ liệu (EDA).....</b>	<b>3</b>
1. Phân tích cơ bản .....	3
2. Trích xuất đặc trưng .....	3
<b>Phần 3: Xây dựng mô hình .....</b>	<b>4</b>
1. Mô hình học máy truyền thống .....	4
2. Mô hình học sâu .....	4
<b>Phần 4: Thiết lập siêu tham số .....</b>	<b>5</b>
1. Học máy truyền thống.....	5
2. Học sâu .....	5
<b>Phần 5: Huấn luyện mô hình.....</b>	<b>5</b>
1. Random Forest .....	5
2. Decision Tree .....	8
3. XGBoost Classifier .....	10
4. Logistic Regression.....	11
5. RNN với LSTM .....	12
6. BERT .....	13
7. XLNet .....	13
<b>Phần 6: Kết luận .....</b>	<b>13</b>
<b>Phần 7: Đánh giá điểm mạnh, yếu các mô hình.....</b>	<b>14</b>
<b>Phần 8: Đề xuất cải thiện .....</b>	<b>18</b>

## Chi tiết đề tài

Việc phát tán rộng rãi tin tức giả và tuyên truyền gây ra những rủi ro nghiêm trọng cho xã hội, bao gồm sự sôi mòn lòng tin của công chúng, phân cực chính trị, thao túng bầu cử và lan truyền thông tin sai lệch có hại trong các cuộc khủng hoảng như đại dịch và xung đột. Theo quan điểm của NLP, việc phát hiện tin tức giả mạo đầy rẫy những thách thức. Về mặt ngôn ngữ, tin tức giả mạo bắt chước giọng điệu và cấu trúc của báo chí chính thống, khiến việc phân biệt bằng các tính năng bề mặt trở nên khó khăn. Việc không có các tập dữ liệu được gắn nhãn đáng tin cậy và cập nhật, đặc biệt là trên nhiều ngôn ngữ và khu vực, cản trở hiệu quả của các mô hình học có giám sát. Ngoài ra, bản chất năng động và đổi đầu của thông tin sai lệch có nghĩa là những tác nhân độc hại liên tục phát triển ngôn ngữ và chiến lược của chúng để vượt qua các hệ thống phát hiện. Bối cảnh văn hóa, sự mĩa mai, châm biếm và thành kiến ngầm càng làm phức tạp thêm quá trình phân tích tự động. Hơn nữa, các mô hình NLP có nguy cơ khuếch đại các thành kiến có trong dữ liệu đào tạo, dẫn đến phân loại không công bằng và khả năng kiểm duyệt nội dung hợp pháp. Những thách thức này nhấn mạnh nhu cầu về các phương pháp tiếp cận thận trọng, có nhận thức về bối cảnh, vì việc không giải quyết chúng có thể vô tình góp phần gây ra thông tin sai lệch, thay vì giảm thiểu nó.

## Bộ dữ liệu

### 1. Bài viết có thật:

- **File:** MisinfoSuperset\_TRUE.csv
- **Nguồn:** các phương tiện truyền thông có uy tín như Reuters, The New York Times, The Washington Post, ...

### 2. Bài viết giả mạo/ thông tin sai lệch/ tuyên truyền:

- **File:** MisinfoSuperset\_FAKE.csv
- **Nguồn:** các trang web cực đoan cánh hữu của Mỹ (ví dụ: Redflag Newsdesk, Breitbart, Truth Broadcast Network).

*Ahmed, H., Traore, I., & Saad, S. (2017): "Detection of Online Fake News Using N-Gram Analysis and Machine Learning Techniques" (Springer LNCS 10618)*

## Phần 1: Tiền xử lý dữ liệu

### 1. Đọc và Gộp dữ liệu

- Hai tệp CSV được đọc bằng thư viện pandas và gộp thành một DataFrame duy nhất.
- Nhãn được gán: 1 cho bài báo thật, 0 cho bài báo giả.
- Cột không cần thiết (Unnamed: 0) được xóa nếu có.

### 2. Làm sạch dữ liệu

- Loại bỏ trùng lặp: Phát hiện và xóa 10,012 dòng trùng lặp, giảm kích thước dữ liệu từ 78,617 xuống 68,605 dòng.
- Xử lý giá trị thiếu: Xóa các dòng có cột text bị thiếu, giảm còn 68,604 dòng.
- Sao chép dữ liệu: Tạo các bản sao của DataFrame (df\_deeplearning, df\_rnn) để sử dụng cho các mô hình học sâu sau này.

### 3. Kết quả

- DataFrame cuối cùng có 68,604 dòng với hai cột: text (nội dung bài báo) và label (nhãn thật/giả).
- Không còn giá trị thiếu hoặc trùng lặp.

## Phần 2: Khám phá dữ liệu (EDA)

### 1. Phân tích cơ bản

- Kích thước tập dữ liệu: 68,604 dòng, 2 cột.
- Số bài báo thật: 34,975 (sau khi làm sạch: 34,526).
- Số bài báo giả: 43,642 (sau khi làm sạch: 34,078).
- Phân bố nhãn: Gần cân bằng (50.3% giả, 49.7% thật).

### 2. Trích xuất đặc trưng

Các đặc trưng bổ sung được trích xuất từ văn bản bao gồm:

- **num\_words:** Số từ trong bài báo.
- **num\_chars:** Số ký tự.

- **avg\_word\_len:** Độ dài trung bình của từ.
- **num\_upper:** Số ký tự in hoa.
- **num\_punct:** Số dấu câu.
- **num\_stopwords:** Số từ dừng (stopwords).

### 3. Phân tích từ phổ biến dựa trên thống kê

Để hiểu rõ hơn về nội dung văn bản, các từ phổ biến nhất trong mỗi lớp (thật và giả) được thống kê sau khi loại bỏ từ dừng và dấu câu, sử dụng CountVectorizer từ thư viện scikit-learn. Các bước thực hiện bao gồm:

- Tiền xử lý văn bản: Chuyển đổi thành chữ thường, loại bỏ từ dừng và dấu câu.
- Đếm tần suất từ: Lấy 10 từ phổ biến nhất trong bài báo thật và giả.

## Phần 3: Xây dựng mô hình

### 1. Mô hình học máy truyền thống

- **Random Forest:** Sử dụng tập hợp cây quyết định để phân loại.
- **Decision Tree:** Một cây quyết định đơn giản.
- **XGBoost Classifier:** Mô hình gradient boosting mạnh mẽ.
- **Logistic Regression:** Mô hình tuyến tính dựa trên xác suất.

### 2. Mô hình học sâu

- RNN với LSTM: Mạng nơ-ron hồi quy sử dụng các đơn vị LSTM để xử lý chuỗi văn bản.
- BERT (bert-base-uncased): Mô hình transformer được huấn luyện trước, tinh chỉnh cho bài toán phân loại.
- XLNet (xlnet-base-cased): Mô hình transformer cải tiến, tối ưu hóa cho các nhiệm vụ NLP.

## Phần 4: Thiết lập siêu tham số

### 1. Học máy truyền thống

- Sử dụng tìm kiếm lưới (Grid Search) hoặc tìm kiếm ngẫu nhiên (Random Search) để tối ưu hóa siêu tham số như độ sâu cây, số lượng cây (Random Forest), tỷ lệ học (XGBoost), và hệ số điều chuẩn (Logistic Regression).

### 2. Học sâu

- RNN với LSTM: Tinh chỉnh số lớp ẩn, kích thước embedding, và tỷ lệ dropout.
- BERT và XLNet: Sử dụng các siêu tham số như:
- Số epoch: 3.
- Kích thước batch: 24.
- Chế độ FP16 để tối ưu hóa hiệu suất GPU.
- Tỷ lệ học mặc định của Hugging Face Trainer.

## Phần 5: Huấn luyện mô hình

- Chia dữ liệu: Dữ liệu được chia thành tập huấn luyện (70%), tập xác thực (15%), và tập kiểm tra (15%) dùng cho các mô hình học sâu và huấn luyện (80%) và tập kiểm tra (20%).
- Các mô hình học máy truyền thống được huấn luyện trên các đặc trưng trích xuất từ văn bản (TF-IDF hoặc đặc trưng số).
- Các mô hình học sâu (RNN, BERT, XLNet) được huấn luyện trên văn bản đã mã hóa (tokenized) sử dụng GPU.
- BERT và XLNet được huấn luyện trong 3 epoch, với kết quả được lưu vào thư mục results\_bert và results\_xlnet.

### 1. Random Forest

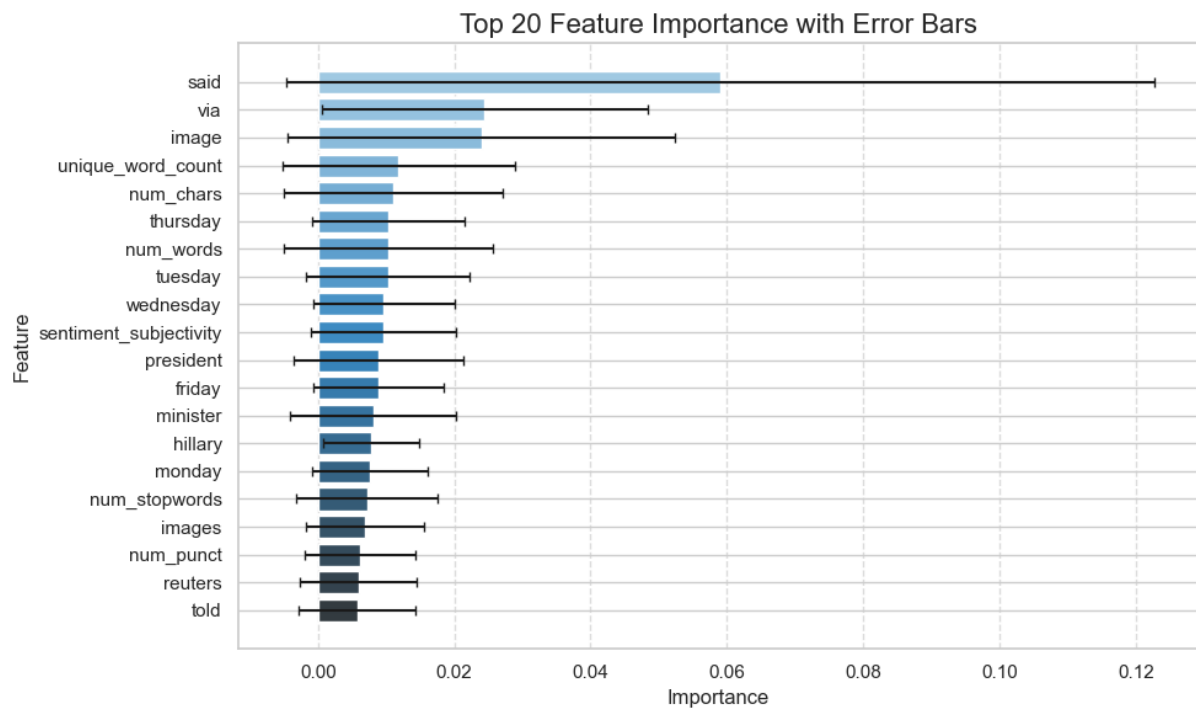
### Classification Report (Train):

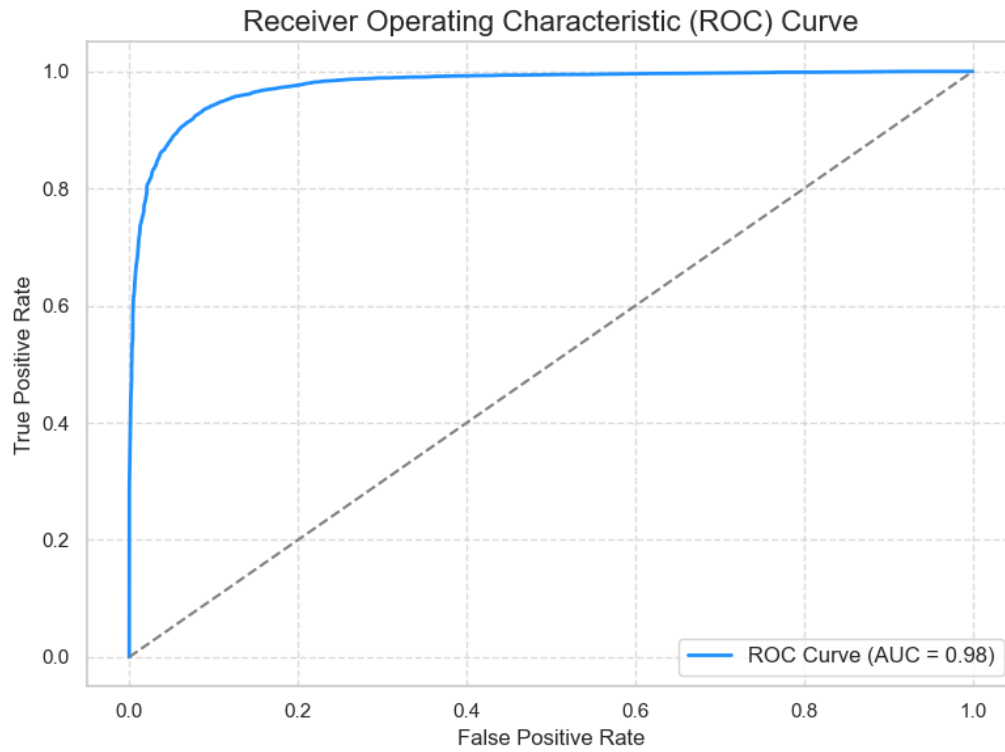
	precision	recall	f1-score	support
0	1.00	1.00	1.00	27262
1	1.00	1.00	1.00	27621
accuracy			1.00	54883
macro avg	1.00	1.00	1.00	54883
weighted avg	1.00	1.00	1.00	54883

Train report saved to reports\rf\_train\_report.txt

### Classification Report (Test):

	precision	recall	f1-score	support
0	0.93	0.91	0.92	6816
1	0.91	0.94	0.92	6905
accuracy			0.92	13721
macro avg	0.92	0.92	0.92	13721
weighted avg	0.92	0.92	0.92	13721







## 2. Decision Tree

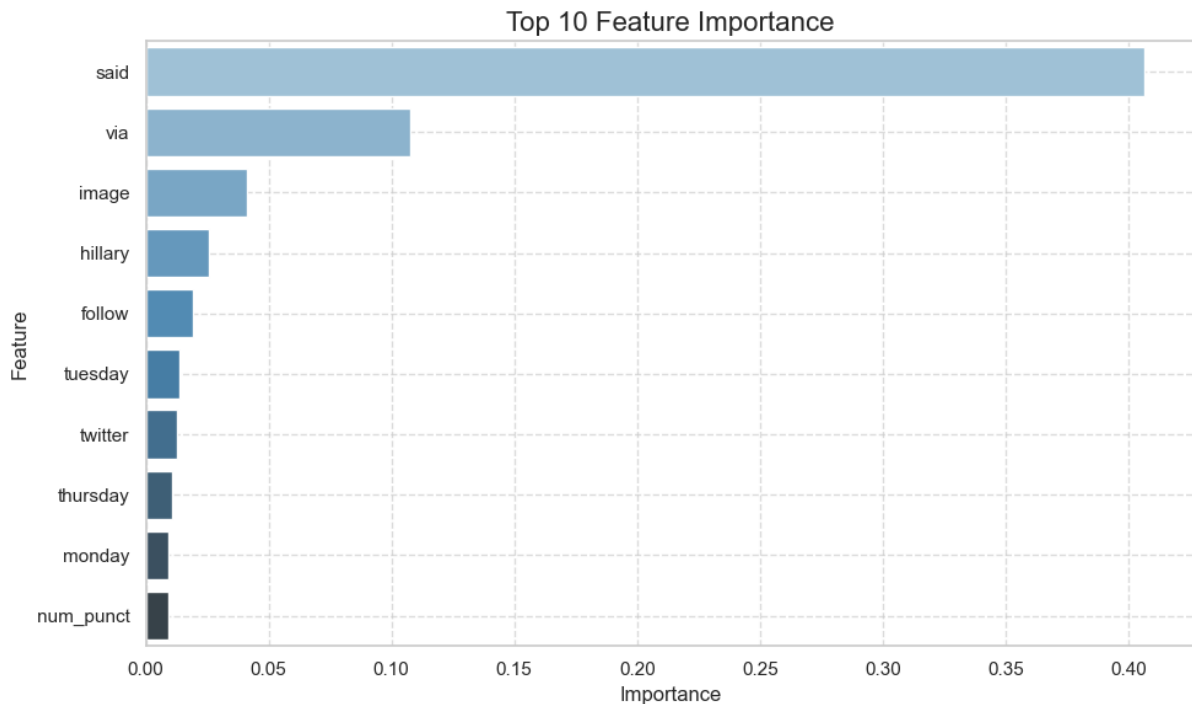
### Classification Report (Train):

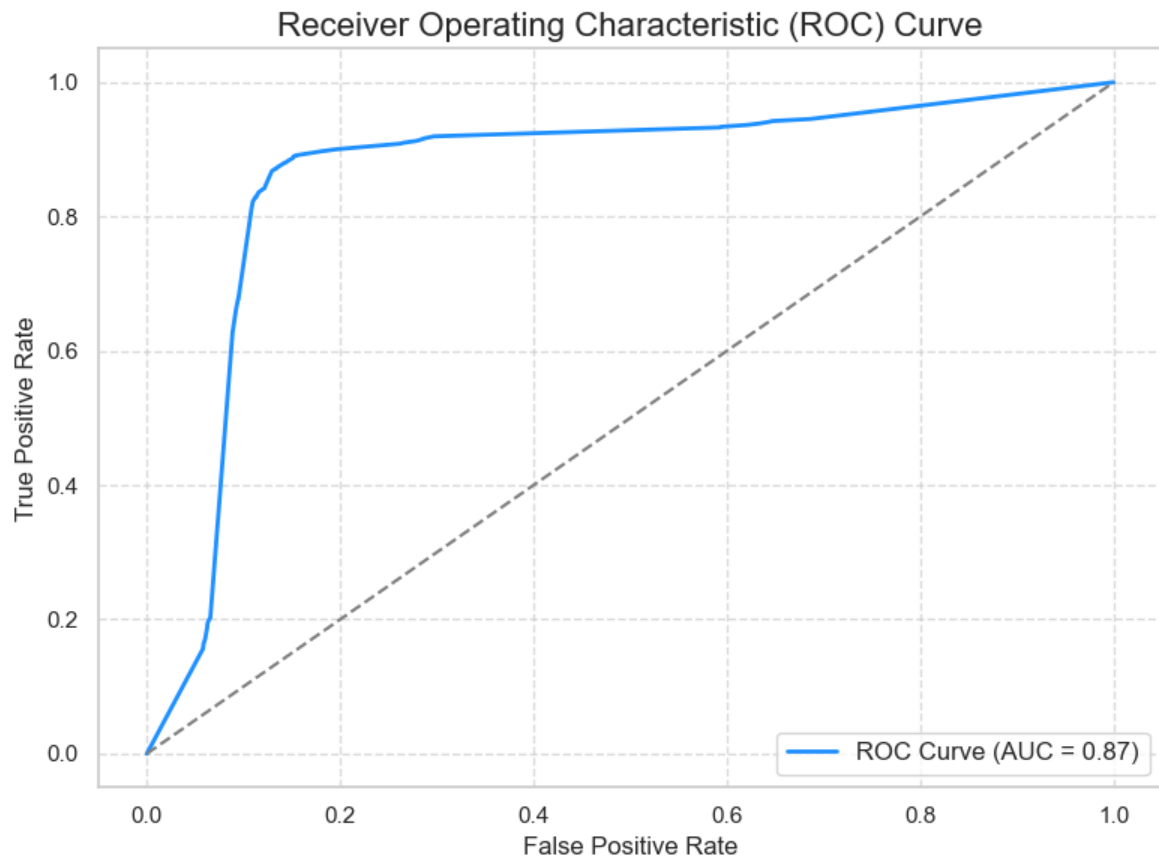
	precision	recall	f1-score	support
0	0.96	0.95	0.95	27262
1	0.95	0.96	0.95	27621
accuracy			0.95	54883
macro avg	0.95	0.95	0.95	54883
weighted avg	0.95	0.95	0.95	54883

Train report saved to reports\dt\_train\_report.txt

### Classification Report (Test):

	precision	recall	f1-score	support
0	0.87	0.86	0.87	6816
1	0.87	0.88	0.87	6905
accuracy			0.87	13721
macro avg	0.87	0.87	0.87	13721
weighted avg	0.87	0.87	0.87	13721





### 3. XGBoost Classifier

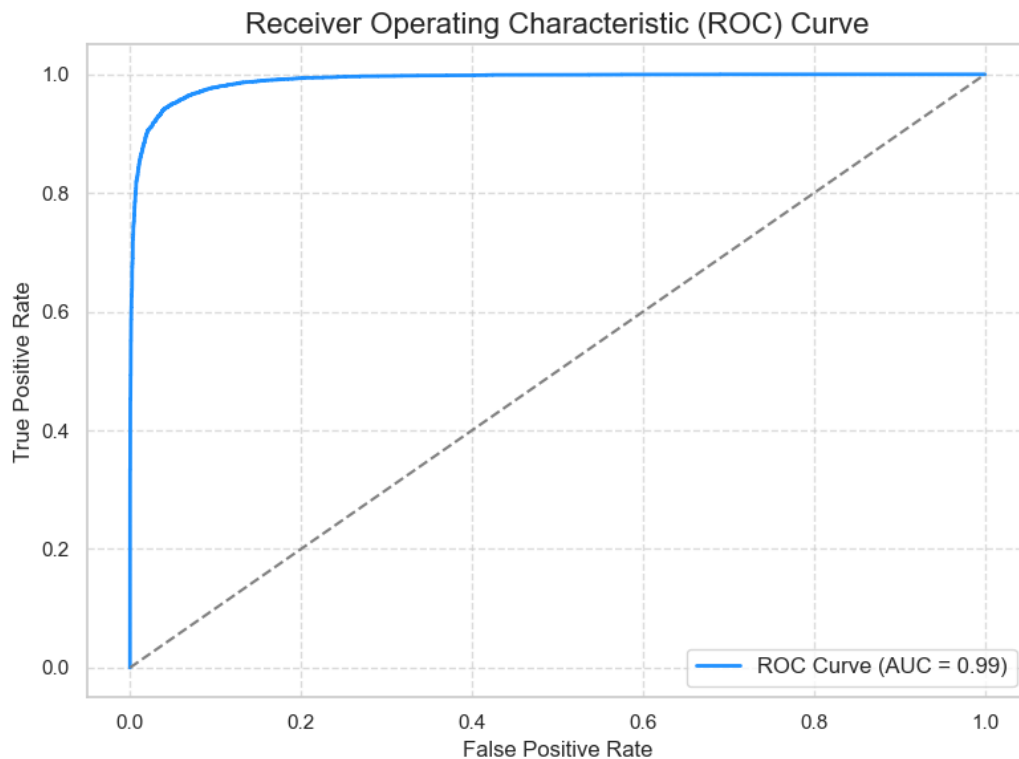
Classification Report (Train):

	precision	recall	f1-score	support
0	0.99	1.00	1.00	27262
1	1.00	0.99	1.00	27621
accuracy			1.00	54883
macro avg	1.00	1.00	1.00	54883
weighted avg	1.00	1.00	1.00	54883

Train report saved to 'reports/xgb\_train\_report.txt'

Classification Report (Test):

	precision	recall	f1-score	support
0	0.95	0.95	0.95	6816
1	0.95	0.95	0.95	6905
accuracy			0.95	13721
macro avg	0.95	0.95	0.95	13721
weighted avg	0.95	0.95	0.95	13721



#### 4. Logistic Regression

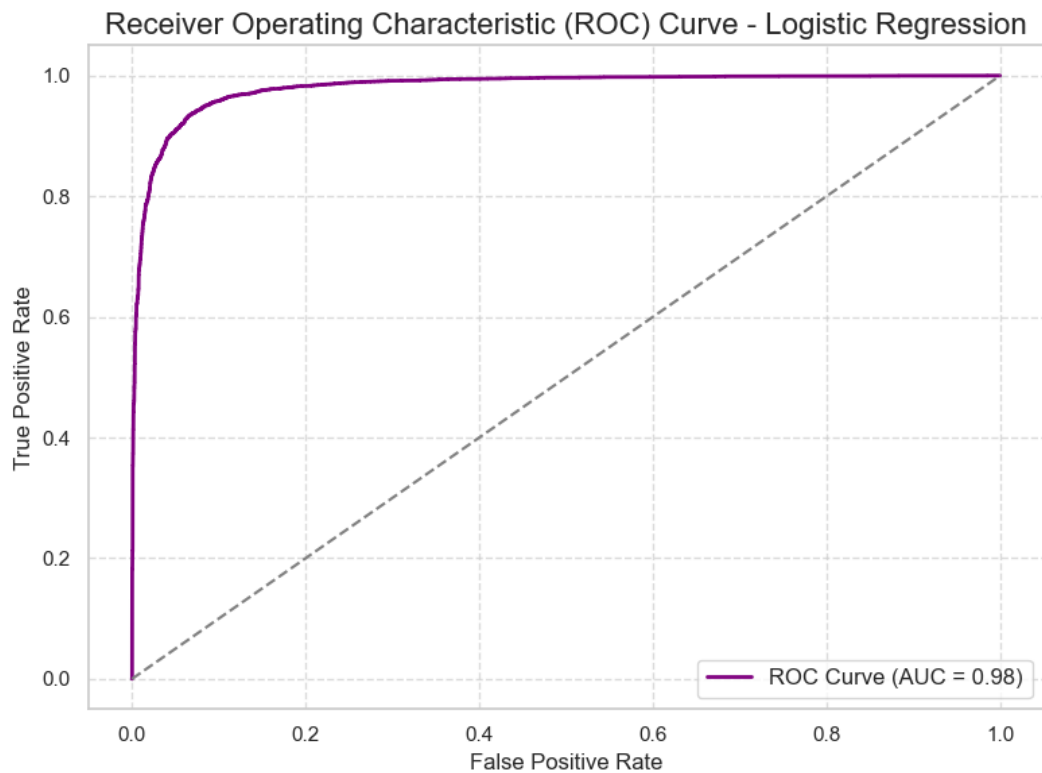
##### Classification Report (Train):

	precision	recall	f1-score	support
0	0.94	0.95	0.95	27262
1	0.95	0.94	0.95	27621
accuracy			0.95	54883
macro avg	0.95	0.95	0.95	54883
weighted avg	0.95	0.95	0.95	54883

Train report saved to 'reports\lr\_train\_report.txt'

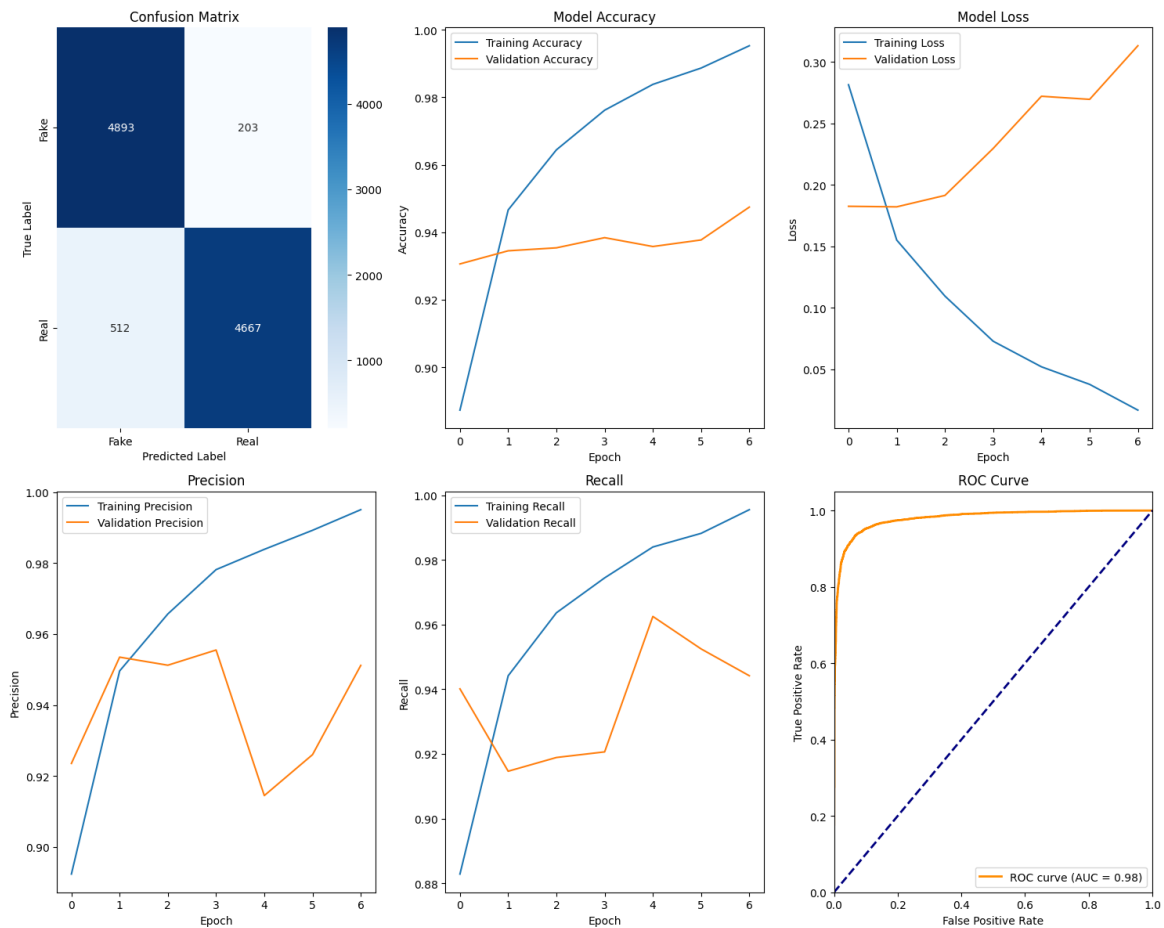
##### Classification Report (Test):

	precision	recall	f1-score	support
0	0.93	0.94	0.93	6816
1	0.94	0.93	0.93	6905
accuracy			0.93	13721
macro avg	0.93	0.93	0.93	13721
weighted avg	0.93	0.93	0.93	13721



# 5. RNN với LSTM

MODEL PERFORMANCE METRICS				
Accuracy:	0.9304			
Precision:	0.9583			
Recall:	0.9011			
F1-Score:	0.9288			
DETAILED CLASSIFICATION REPORT				
	precision	recall	f1-score	support
Fake News	0.91	0.96	0.93	5096
Real News	0.96	0.90	0.93	5179
accuracy			0.93	10275
macro avg	0.93	0.93	0.93	10275
weighted avg	0.93	0.93	0.93	10275



## 6. BERT

Epoch	Training Loss	Validation Loss	Accuracy	F1	Precision	Recall
1	0.101800	0.111401	0.972209	0.972190	0.973213	0.972209
2	0.029200	0.070092	0.982606	0.982605	0.982664	0.982606
3	0.006700	0.082757	0.984744	0.984744	0.984745	0.984744

## 7. XLNet

Epoch	Training Loss	Validation Loss	Accuracy	F1	Precision	Recall
1	0.087900	0.068309	0.983578	0.983578	0.983584	0.983578
2	0.041000	0.042746	0.986202	0.986200	0.986280	0.986202
3	0.018200	0.040245	0.989408	0.989408	0.989411	0.989408

## Phần 6: Kết luận

- XLNet đạt hiệu suất cao nhất với độ chính xác 99.04% trên tập kiểm tra, vượt trội hơn BERT (98.51%).
- Các mô hình học máy truyền thống và RNN với LSTM có khả năng thấp hơn trong việc xử lý các bài báo phức tạp.
- Tiền xử lý dữ liệu: Dữ liệu được làm sạch hiệu quả, loại bỏ trùng lặp và giá trị thiếu, đảm bảo chất lượng đầu vào cho các mô hình.
- EDA cho thấy tập dữ liệu với 68,604 bài báo có phân bố nhãn gần cân bằng (50.3% giả, 49.7% thật), hỗ trợ lựa chọn mô hình và chia dữ liệu hợp lý. Các đặc trưng văn bản như số từ, ký tự, và từ dừng được trích xuất, làm sáng tỏ sự khác biệt ngôn ngữ giữa tin thật và giả, định hướng trích xuất đặc trưng hiệu quả. Những phát hiện này giúp tối ưu hóa tiền xử lý và huấn luyện, đặc biệt cho các mô hình học sâu như BERT và XLNet.
- Huấn luyện và suy diễn: Các mô hình transformer (BERT, XLNet) hoạt động tốt, đặc biệt là XLNet với khả năng học ngữ cảnh hai chiều.

## Phần 7: Đánh giá điểm mạnh, yếu các mô hình

### 1. Random Forest

Điểm mạnh	Điểm yếu
<ul style="list-style-type: none"><li>• <b>Khả năng chống overfitting:</b> Sử dụng tập hợp nhiều cây quyết định, giảm thiểu nguy cơ quá khớp so với một cây quyết định đơn lẻ.</li><li>• <b>Xử lý dữ liệu không cân bằng:</b> Có thể hoạt động tốt trên tập dữ liệu với nhãn không cân bằng nhờ cơ chế lấy mẫu ngẫu nhiên.</li><li>• <b>Dễ sử dụng:</b> Yêu cầu ít tiền xử lý dữ liệu (không cần chuẩn hóa) và dễ tinh chỉnh siêu tham số.</li><li>• <b>Khả năng giải thích:</b> Cung cấp mức độ quan trọng của các đặc trưng, hỗ trợ phân tích các yếu tố ảnh hưởng đến dự đoán.</li></ul>	<ul style="list-style-type: none"><li>• <b>Hiệu suất hạn chế với văn bản:</b> Không tận dụng được ngữ cảnh chuỗi dài trong dữ liệu văn bản, phụ thuộc nhiều vào đặc trưng trích xuất (như TF-IDF).</li><li>• <b>Tốn tài nguyên:</b> Với số lượng cây lớn và dữ liệu lớn (68,604 dòng), thời gian huấn luyện và suy diễn có thể chậm.</li><li>• <b>Khó xử lý ngữ nghĩa phức tạp:</b> Không hiệu quả trong việc phát hiện các mẫu tin giả sử dụng châm biếm, giễu nhại hoặc ngữ cảnh văn hóa.</li></ul>

### 2. Decision Tree

Điểm mạnh	Điểm yếu
<ul style="list-style-type: none"><li>• <b>Dễ hiểu và trực quan:</b> Cấu trúc cây quyết định dễ hình dung, giúp giải thích các quyết định của mô hình.</li></ul>	<ul style="list-style-type: none"><li>• <b>Dễ overfitting:</b> Một cây quyết định đơn lẻ dễ bị quá khớp, đặc biệt với dữ liệu văn bản phức tạp.</li></ul>

<ul style="list-style-type: none"> <li>• <b>Nhanh chóng:</b> Thời gian huấn luyện và suy diễn nhanh trên tập dữ liệu vừa và nhỏ.</li> <li>• <b>Không cần chuẩn hóa dữ liệu:</b> Có thể xử lý các đặc trưng không đồng nhất mà không cần tiền xử lý phức tạp.</li> </ul>	<ul style="list-style-type: none"> <li>• <b>Hiệu suất thấp:</b> So với các mô hình phức tạp hơn, Decision Tree thường có độ chính xác thấp hơn trên bài toán NLP.</li> <li>• <b>Không tận dụng ngữ cảnh:</b> Không thể nắm bắt được mối quan hệ ngữ nghĩa giữa các từ trong bài báo.</li> </ul>
---	---

### 3. XGBoost Classifier

Điểm mạnh	Điểm yếu
<ul style="list-style-type: none"> <li>• <b>Hiệu suất cao:</b> Là một trong những mô hình học máy mạnh mẽ nhất nhờ sử dụng gradient boosting.</li> <li>• <b>Tối ưu hóa tốc độ:</b> Được tối ưu cho hiệu suất tính toán, phù hợp với tập dữ liệu lớn.</li> <li>• <b>Khả năng tinh chỉnh:</b> Hỗ trợ nhiều siêu tham số (tỷ lệ học, độ sâu cây, điều chuẩn) để cải thiện hiệu suất.</li> <li>• <b>Xử lý dữ liệu không cân bằng:</b> Có các tham số như <code>scale_pos_weight</code> để xử lý tốt dữ liệu không cân bằng.</li> </ul>	<ul style="list-style-type: none"> <li>• <b>Phụ thuộc vào đặc trưng:</b> Hiệu quả phụ thuộc nhiều vào chất lượng đặc trưng trích xuất (TF-IDF, đặc trưng số).</li> <li>• <b>Khó giải thích:</b> Mô hình phức tạp hơn Random Forest, khó cung cấp giải thích trực quan cho dự đoán.</li> <li>• <b>Tốn thời gian tinh chỉnh:</b> Yêu cầu tìm kiếm siêu tham số kỹ lưỡng để đạt hiệu suất tối ưu.</li> </ul>

### 4. Logistic Regression



Điểm mạnh	Điểm yếu
<ul style="list-style-type: none"> <li>• <b>Đơn giản và hiệu quả:</b> Là mô hình tuyến tính, dễ triển khai và huấn luyện nhanh trên dữ liệu lớn.</li> <li>• <b>Khả năng giải thích:</b> Hệ số của các đặc trưng có thể được phân tích để hiểu mức độ ảnh hưởng.</li> <li>• <b>Tốt cho bài toán nhị phân:</b> Phù hợp với bài toán phân loại thật/giả nhờ đầu ra xác suất.</li> <li>• <b>Ít nhạy cảm với nhiễu:</b> Hoạt động tốt khi đặc trưng được tiền xử lý tốt.</li> </ul>	<ul style="list-style-type: none"> <li>• <b>Hạn chế với dữ liệu phi tuyến:</b> Không thể nắm bắt các mối quan hệ phức tạp trong văn bản.</li> <li>• <b>Phụ thuộc vào đặc trưng:</b> Cần trích xuất đặc trưng chất lượng cao (như TF-IDF) để đạt hiệu suất tốt.</li> <li>• <b>Không xử lý ngữ cảnh:</b> Không thể hiểu ngữ nghĩa hoặc mối quan hệ chuỗi trong văn bản.</li> </ul>

## 5. RNN với LSTM

Điểm mạnh	Điểm yếu
<ul style="list-style-type: none"> <li>• <b>Xử lý chuỗi:</b> LSTM có khả năng nắm bắt mối quan hệ tuần tự trong văn bản, phù hợp với bài toán NLP.</li> <li>• <b>Học ngữ cảnh dài:</b> Có thể ghi nhớ thông tin từ các từ trước đó, cải thiện việc hiểu ngữ nghĩa.</li> <li>• <b>Linh hoạt:</b> Có thể tùy chỉnh kiến trúc (số lớp, kích thước embedding, dropout) để tối ưu hóa.</li> </ul>	<ul style="list-style-type: none"> <li>• <b>Hiệu suất thấp hơn transformer:</b> So với BERT và XLNet, RNN với LSTM kém hiệu quả trong việc xử lý ngữ cảnh hai chiều.</li> <li>• <b>Thời gian huấn luyện lâu:</b> Yêu cầu nhiều thời gian và tài nguyên tính toán, đặc biệt với tập dữ liệu lớn.</li> </ul>

	<ul style="list-style-type: none"> <li>• <b>Khó tinh chỉnh:</b> Siêu tham số như số lớp ẩn, tỷ lệ dropout cần được điều chỉnh cẩn thận để tránh overfitting.</li> <li>• <b>Hạn chế với văn bản dài:</b> LSTM gặp khó khăn khi xử lý các bài báo dài do vấn đề vanishing gradient.</li> </ul>
--	--

## 6. BERT (bert-base-uncased)

Điểm mạnh	Điểm yếu
<ul style="list-style-type: none"> <li>• <b>Hiểu ngữ cảnh hai chiều:</b> BERT được huấn luyện trước trên dữ liệu lớn, nắm bắt ngữ nghĩa và mối quan hệ ngữ cảnh tốt.</li> <li>• <b>Hiệu suất cao:</b> Đạt độ chính xác 98.51% trên tập kiểm tra, phù hợp cho bài toán phân loại tin giả.</li> <li>• <b>Khả năng tổng quát:</b> Có thể tinh chỉnh cho nhiều bài toán NLP khác nhau.</li> <li>• <b>Hỗ trợ cộng đồng mạnh:</b> Thư viện Hugging Face cung cấp công cụ dễ sử dụng để huấn luyện và suy diễn.</li> </ul>	<ul style="list-style-type: none"> <li>• <b>Tốn tài nguyên:</b> Yêu cầu GPU mạnh và RAM lớn để huấn luyện và suy diễn trên tập dữ liệu lớn.</li> <li>• <b>Thời gian huấn luyện dài:</b> Mặc dù chỉ huấn luyện 3 epoch, thời gian vẫn đáng kể (19 phút).</li> <li>• <b>Khó giải thích:</b> Là mô hình black-box, khó phân tích lý do đưa ra dự đoán cụ thể.</li> <li>• <b>Hạn chế với châm biếm:</b> Có thể gặp khó khăn khi phân biệt tin giả sử dụng ngôn ngữ phức tạp hoặc châm biếm.</li> </ul>

## 7. XLNet (xlnet-base-cased)

Điểm mạnh	Điểm yếu
<ul style="list-style-type: none"> <li>• <b>Hiệu suất vượt trội:</b> Đạt độ chính xác cao nhất (99.04%) nhờ cơ chế học ngữ cảnh hai chiều cải tiến.</li> <li>• <b>Xử lý chuỗi dài tốt hơn:</b> So với BERT, XLNet tối ưu hơn trong việc nắm bắt mối quan hệ giữa các từ xa nhau.</li> <li>• <b>Tổng quát hóa tốt:</b> Được huấn luyện trước trên dữ liệu lớn, phù hợp cho nhiều bài toán NLP.</li> <li>• <b>Khả năng phân biệt ngữ nghĩa phức tạp:</b> Hiệu quả hơn BERT trong việc xử lý các bài báo có ngữ cảnh tinh vi.</li> </ul>	<ul style="list-style-type: none"> <li>• <b>Tốn tài nguyên hơn BERT:</b> Thời gian huấn luyện lâu hơn (29 phút) và yêu cầu bộ nhớ lớn.</li> <li>• <b>Khó giải thích:</b> Tương tự BERT, là mô hình black-box, khó cung cấp giải thích trực quan.</li> <li>• <b>Phức tạp trong triển khai:</b> Yêu cầu hiểu biết sâu về transformer và thư viện Hugging Face để tinh chỉnh.</li> <li>• <b>Hạn chế với dữ liệu đa ngôn ngữ:</b> Chỉ hiệu quả trên dữ liệu tiếng Anh, cần tinh chỉnh thêm cho các ngôn ngữ khác.</li> </ul>

## Phần 8: Đề xuất cải thiện

- **Mở rộng dữ liệu:** Thu thập thêm dữ liệu từ nhiều ngôn ngữ, khu vực, và nguồn để tăng tính tổng quát.
- **Tăng cường đặc trưng:** Kết hợp các đặc trưng bên ngoài như nguồn bài báo, ngày xuất bản, hoặc tín hiệu xã hội (lượt chia sẻ, bình luận).
- **Tối ưu hóa mô hình truyền thống:** Thực hiện tìm kiếm siêu tham số kỹ lưỡng hơn cho Random Forest, XGBoost, và Logistic Regression.
- **Xử lý ngữ cảnh phức tạp:** Áp dụng các kỹ thuật như phân tích cảm xúc hoặc phát hiện châm biếm để cải thiện khả năng phân biệt.
- **Triển khai thực tế:** Xây dựng API hoặc giao diện người dùng để tích hợp mô hình vào các nền tảng kiểm duyệt nội dung.