Báo cáo tóm tắt quá trình phân loại văn bản AG News

1. Giới thiệu

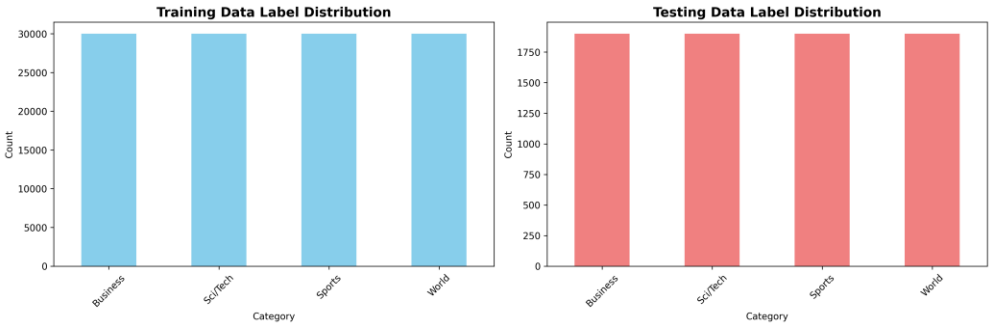
Báo cáo này trình bày quá trình xây dựng và triển khai một hệ thống phân loại văn bản cho tập dữ liệu AG News, bao gồm 4 danh mục: World, Sports, Business, và Sci/Tech. Mục tiêu là phát triển một mô hình có khả năng phân loại tin tức chính xác và đóng gói thành một API dễ sử dụng.

2. Chuẩn bị dữ liệu và tiền xử lý

- **Tải dữ liệu:** Dữ liệu AG News được tải từ thư viện datasets, bao gồm 120,000 mẫu huấn luyện và 7,600 mẫu kiểm tra. Dữ liệu được chuyển đổi sang định dạng Pandas DataFrame.
- **Khám phá dữ liệu:** Phân phối nhãn được phân tích, cho thấy sự cân bằng giữa các danh mục (30,000 mẫu mỗi danh mục). Độ dài văn bản và số lượng từ cũng được phân tích để hiểu rõ hơn về đặc điểm của dữ liệu.
- **Tiền xử lý:** Các bước tiền xử lý bao gồm chuyển đổi chữ thường, loại bỏ ký tự đặc biệt, số và khoảng trắng thừa, cũng như loại bỏ stop words tiếng Anh. Quá trình này giúp chuẩn hóa văn bản và giảm nhiễu.

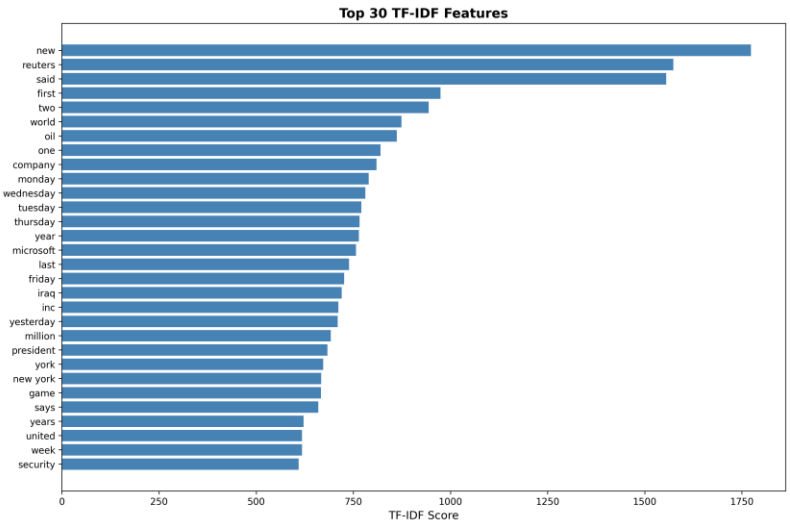
3. Trực quan hóa dữ liệu

- **Phân phối nhãn:** Biểu đồ cột cho thấy phân phối nhãn cân bằng trong cả tập huấn luyện và kiểm tra.
- **Thống kê văn bản:** Các biểu đồ phân phối độ dài văn bản và số lượng từ, cùng với biểu đồ boxplot theo danh mục, cung cấp cái nhìn về sự biến động của dữ liệu.
- **Word Cloud và Top Words:** Word Cloud và danh sách 20 từ hàng đầu được tạo cho từng danh mục để hình dung các từ khóa nổi bật, giúp xác nhận tính hiệu quả của bước tiền xử lý và hiểu biết sâu hơn về nội dung của mỗi nhãn.



4. Trích xuất đặc trưng

- **TF-IDF Vectorization:** Kỹ thuật TF-IDF (Term Frequency-Inverse Document Frequency) được sử dụng để chuyển đổi văn bản đã làm sạch thành các vector số. TfidfVectorizer được cấu hình với max_features=10000, ngram_range=(1, 2), min_df=5, và max_df=0.8 để tạo ra 10,000 đặc trưng, đại diện cho cả từ đơn và cặp từ.

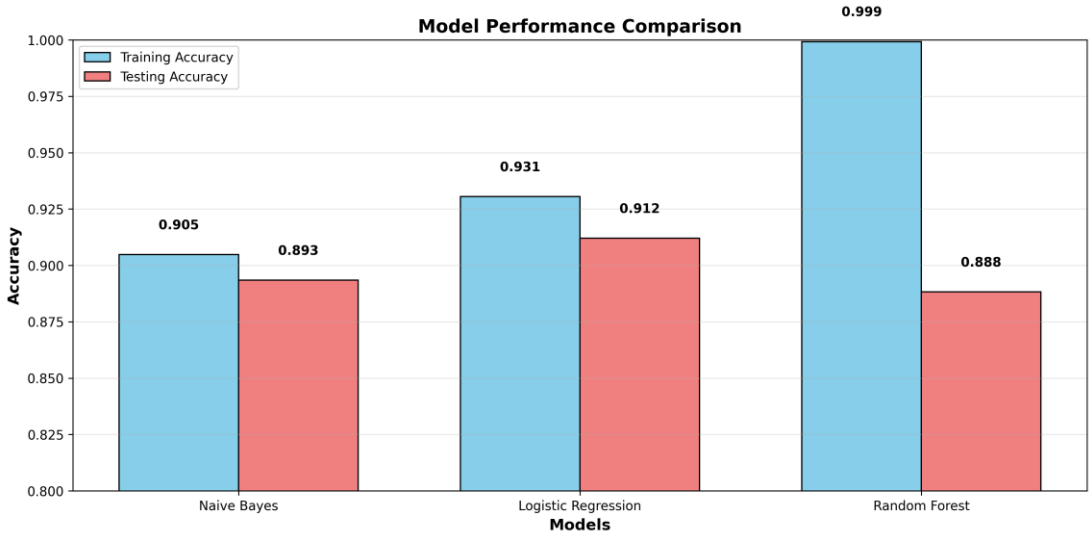


5. Xây dựng và huấn luyện mô hình

- **Mô hình so sánh:** Ba mô hình phân loại được huấn luyện và đánh giá: Naive Bayes (MultinomialNB), Logistic Regression, và Random Forest Classifier.
- **Đánh giá:** Các mô hình được đánh giá dựa trên độ chính xác (Accuracy) trên tập huấn luyện và tập kiểm tra, cùng với báo cáo phân loại chi tiết (precision, recall, f1-score).

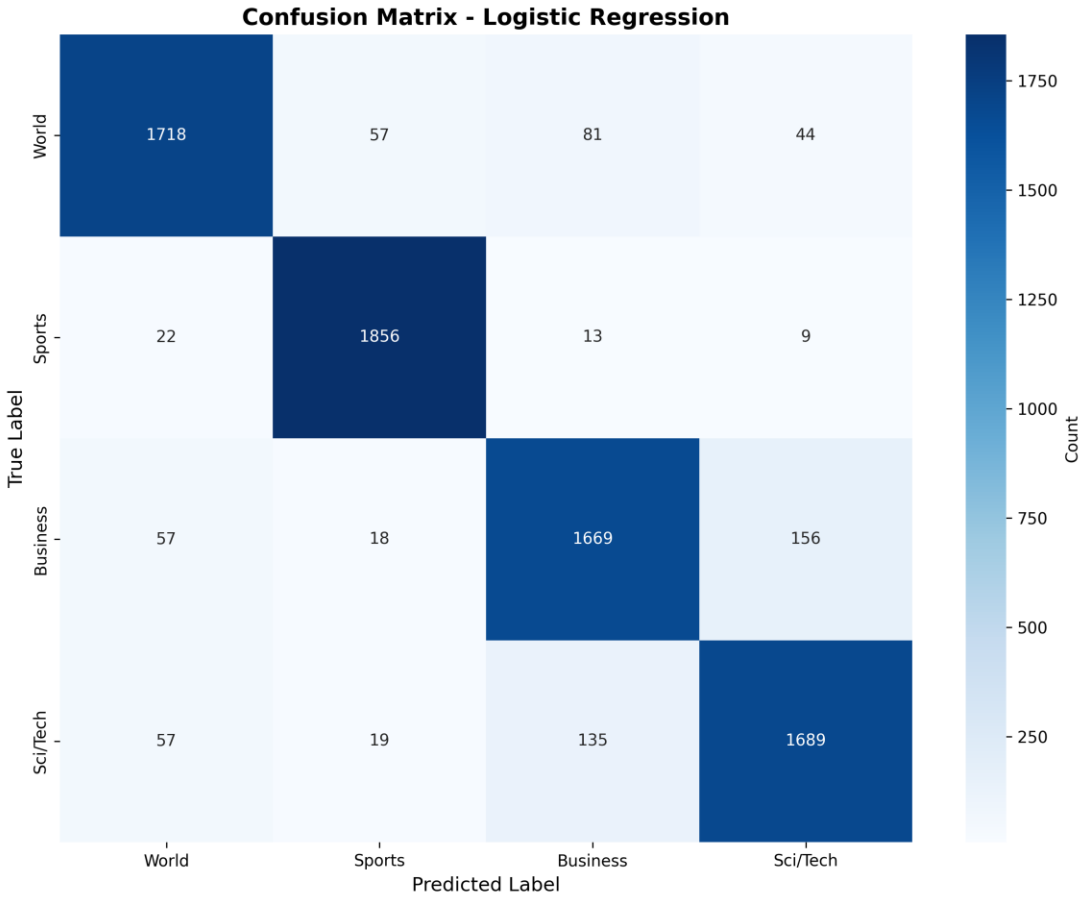
- **Naive Bayes:** Độ chính xác kiểm tra ~0.8934
- **Logistic Regression:** Độ chính xác kiểm tra ~0.9121
- **Random Forest:** Độ chính xác kiểm tra ~0.8883

- **Mô hình tốt nhất:** Logistic Regression cho thấy hiệu suất tốt nhất với độ chính xác kiểm tra là 0.9121.



6. Đánh giá chi tiết mô hình tốt nhất (Logistic Regression)

- **Ma trận nhầm lẫn (Confusion Matrix):** Ma trận nhầm lẫn được trực quan hóa để phân tích hiệu suất của mô hình đối với từng danh mục, cho thấy các lớp 'Sports' và 'World' được phân loại tốt hơn, trong khi 'Business' và 'Sci/Tech' có một số nhầm lẫn.
- **Các chỉ số theo lớp:** Precision, Recall, và F1-Score cho từng danh mục được trình bày, xác nhận rằng Logistic Regression là lựa chọn phù hợp nhất cho bài toán này.



7. Lưu trữ mô hình và triển khai API

- **Lưu trữ tài nguyên:** Mô hình Logistic Regression tốt nhất (best_model.pkl), vectorizer TF-IDF (vectorizer.pkl) và ánh xạ nhãn (label_names.pkl) được lưu lại bằng pickle.
- **FastAPI:** Một ứng dụng FastAPI (main.py) được tạo để triển khai mô hình. API cung cấp các điểm cuối (/, /health, /predict) để nhận yêu cầu phân loại văn bản và trả về kết quả dự đoán cùng với độ tin cậy và xác suất của từng lớp.
- **requirements.txt:** Các file requirements.txt cũng được tạo để dễ dàng đóng gói và triển khai ứng dụng FastAPI trong môi trường.
- **README.md:** File README.md cung cấp hướng dẫn chi tiết về cách cài đặt, chạy và sử dụng API.

8. Kết luận

Quá trình đã thành công trong việc xây dựng một hệ thống phân loại tin tức hiệu quả bằng cách sử dụng các kỹ thuật xử lý ngôn ngữ tự nhiên (NLP) cổ điển và mô hình học máy.

Logistic Regression với TF-IDF được chọn là mô hình tốt nhất với độ chính xác ~91.21%. Hệ thống này đã được đóng gói thành một API RESTful, sẵn sàng cho việc tích hợp và sử dụng.