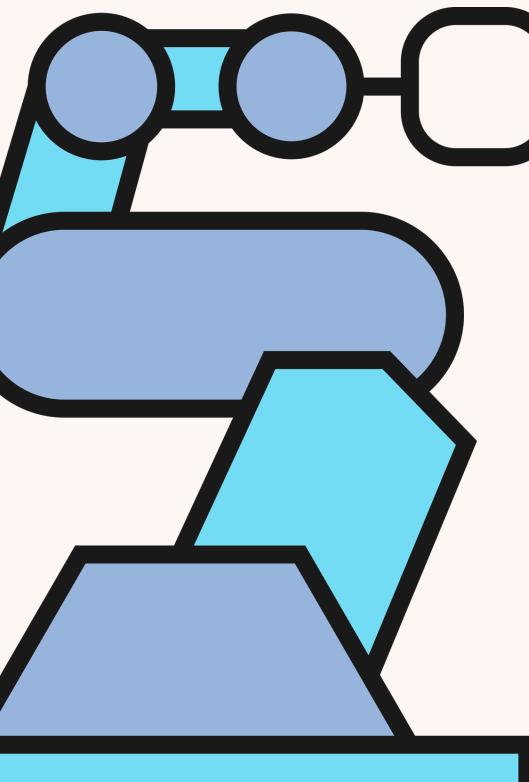


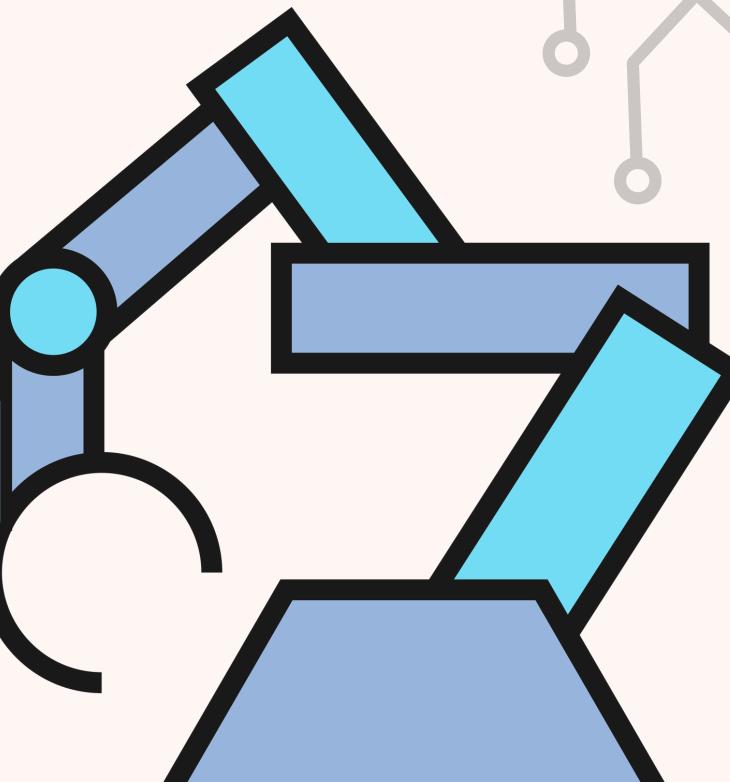
Bài tập lớn
Đề tài:

Dự đoán đánh giá sản phẩm

Ứng dụng Machine Learning & NLP trong phân loại phản hồi khách hàng



Sinh viên thực hiện: Lê Đức Thắng



Giảng viên hướng dẫn: PGS. TS. Nguyễn Văn Hậu

- 
- 1** Tổng quan & đặt vấn đề
 - 2** Khám phá dữ liệu - EDA
 - 3** Tiền xử lý dữ liệu
 - 4** Mô hình hóa & Thuật toán
 - 5** Kết quả thực nghiệm
 - 6** Tổng kết

Phần 1: Tổng quan & đặt vấn đề

Bối cảnh Thương mại điện tử (Context)

- Sự bùng nổ của E-commerce (Amazon, Shopee,...)
- Vai trò của “User Generated Content” (Nội dung người dùng tạo): Review là yếu tố #1 quyết định hành vi mua hàng
- Vấn đề Big Data: Hàng nghìn review/ngày → Không thể kiểm soát thủ công.

Bài toán nghiên cứu(Problem Statement)

- Doanh nghiệp cần phân loại cảm xúc khách hàng (Sentiment Analysis)
- **Input:** Văn bản review (Text) + Thông tin nhân khẩu học (Age) + Hành vi (Rating)
- **Output:** Nhãn nhị phân
 - 0: Không khuyên dùng (Negative/Neutral)
 - 1: Khuyên dùng (Positive)

Thách thức của bài toán

- **Dữ liệu phi cấu trúc (Unstructured Data):** Máy tính không hiểu ngôn ngữ tự nhiên, tiếng long, sai chính tả
- **Mất cân bằng dữ liệu (Imbalanced Class):** Tỷ lệ khen/chê chênh lệnh quá lớn (82% vs 18%)
- **Đa chiều (High Dimensionality):** Số lượng từ vựng lên tới hàng nghìn từ

Mục tiêu đề tài (Objectives)

- Xây dựng Pipeline xử lý dữ liệu tự động (End-to-end)
- Ứng dụng và so sánh hiệu quả của 4 thuật toán Machine Learning cổ điển
- Tìm ra mô hình tối ưu (Champion Model) về độ chính xác và độ ổn định

Phạm vi nghiên cứu(Scope)

- Dữ liệu: Women's Clothing E-Commerce Reviews
- Ngôn ngữ: Tiếng Anh
- Công cụ: Python, Scikit-learn, NLTK, Seaborn

Phần 2: Khám phá dữ liệu - EDA

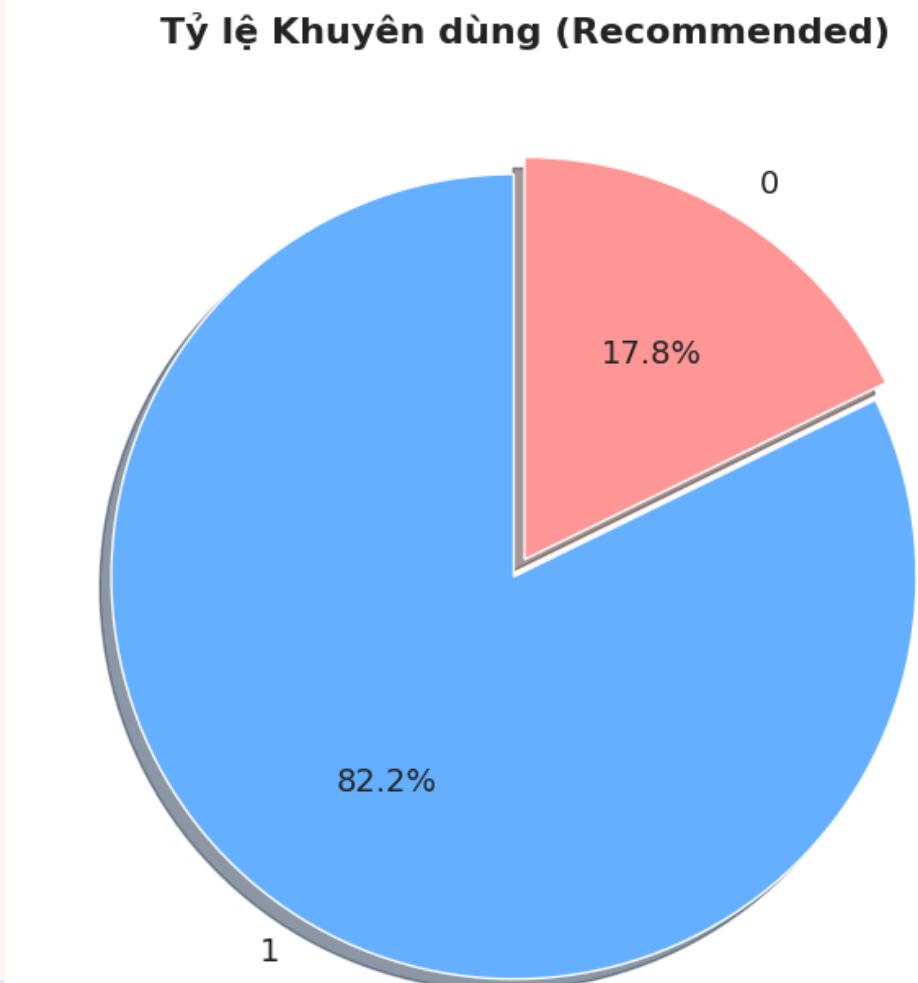
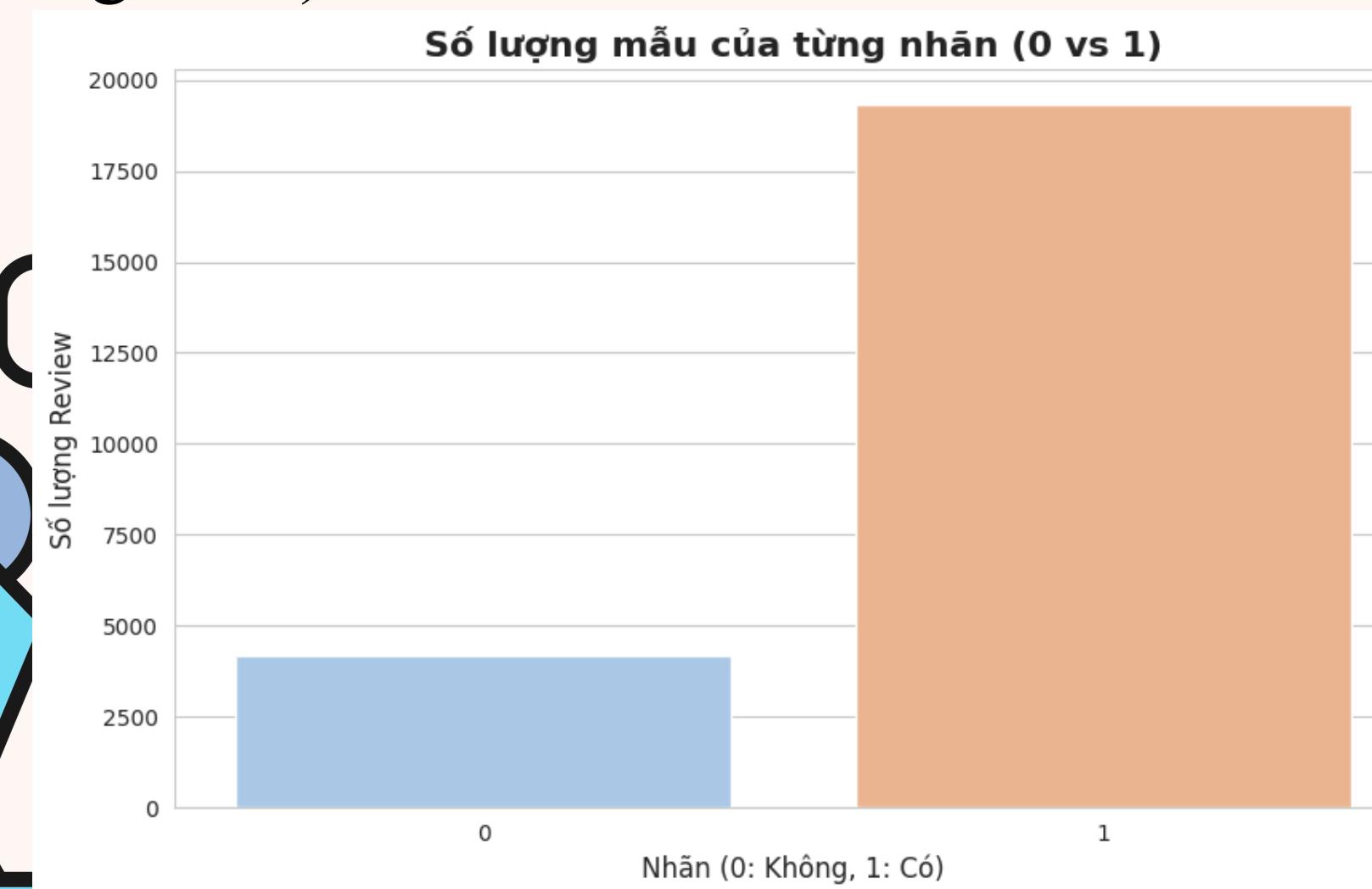
Tổng quan bộ dữ liệu (Dataset Statistics)

- Số lượng mẫu: 23,486 dòng
- Số lượng thuộc tính: 10 cột
- Các thuộc tính quan trọng:
Review Text, Rating (1-5), Age (18-99), Recommended IND (Target)
- Dữ liệu thiếu (Missing values):
Tập trung ở cột Title (Không quan trọng) và Review Text (Cần xử lý)

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 23486 entries, 0 to 23485
Data columns (total 11 columns):
 #   Column           Non-Null Count   Dtype  
--- 
 0   Unnamed: 0        23486 non-null    int64  
 1   Clothing ID      23486 non-null    int64  
 2   Age              23486 non-null    int64  
 3   Title            19676 non-null    object  
 4   Review Text       22641 non-null    object  
 5   Rating           23486 non-null    int64  
 6   Recommended IND  23486 non-null    int64  
 7   Positive Feedback Count  23486 non-null    int64  
 8   Division Name    23472 non-null    object  
 9   Department Name  23472 non-null    object  
 10  Class Name       23472 non-null    object  
dtypes: int64(6), object(5)
memory usage: 2.0+ MB
```

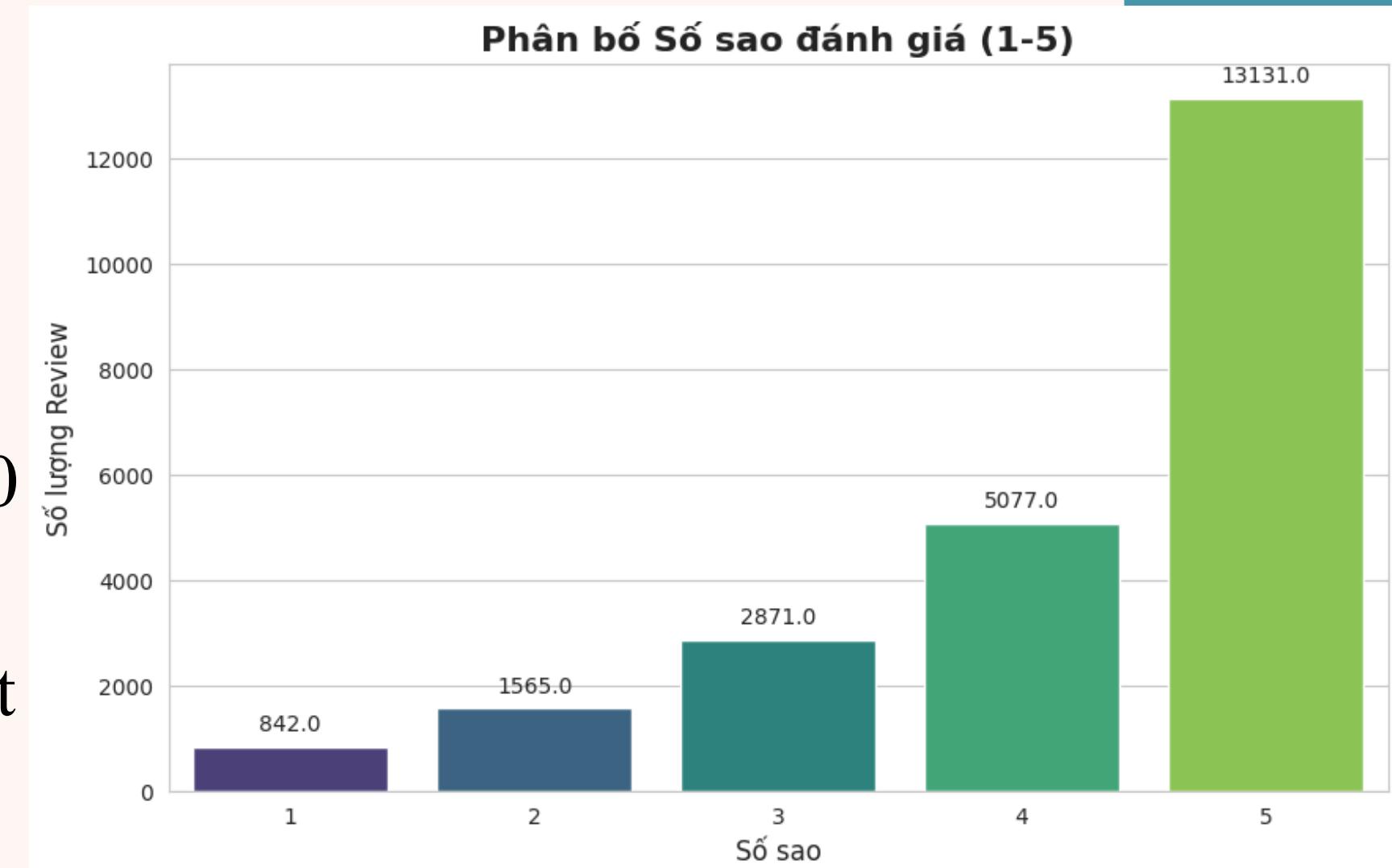
Phân tích biến mục tiêu (Target Analysis)

- Biểu đồ tròn tần số nhãn 0 và 1
- **Nhận định:** Dữ liệu bị lệch nghiêm trọng (Imbalanced)
- **Hệ quả:** Nếu dùng Accuracy làm thước đo duy nhất, mô hình sẽ bị đánh lừa (Ví dụ: Dự đoán tất cả là 1 thì vẫn đúng 82%)



Phân tích phân bố Rating (Rating Distribution)

- Biểu đồ cột phân bố Rating 1-5 sao
- Mối tương quan:
 - Rating 5 sao → 99% là Nhãn 1
 - Rating 1-2 sao → 90% là Nhãn 0
 - Rating 3 sao → Vùng lưỡng lự (Ambiguous), khó phân loại nhất



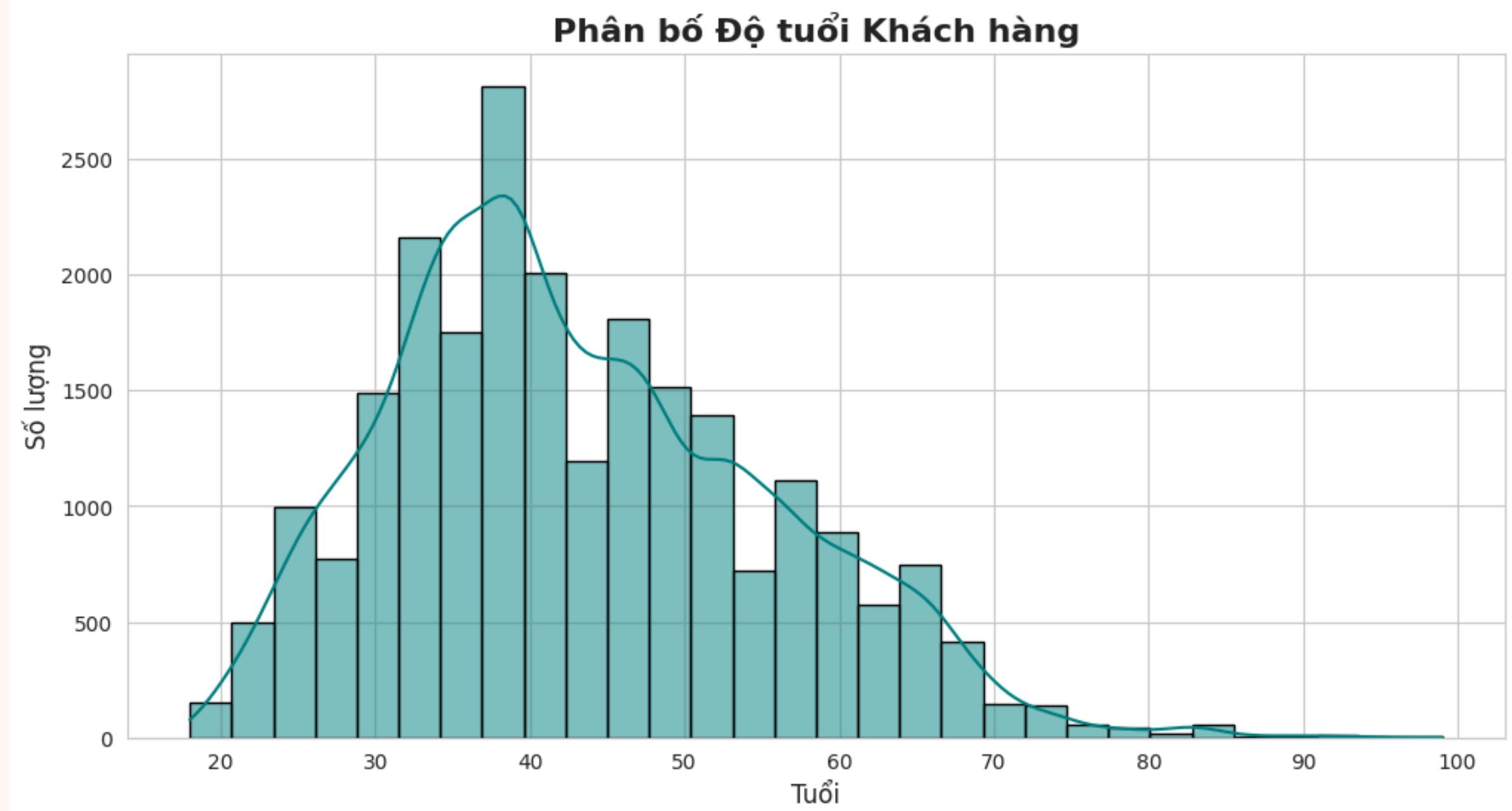
Phân tích đám mây từ (WordCloud)



- Hình ảnh WordCloud cho nhóm Khen và Chê
 - Khen: Love, Great, Perfect, Soft, Comfortable
 - Chê: Return, Small, Cheap, Disappointed, Tight
=> Khẳng định: từ khóa là đặc trưng quan trọng nhất để
hân loại

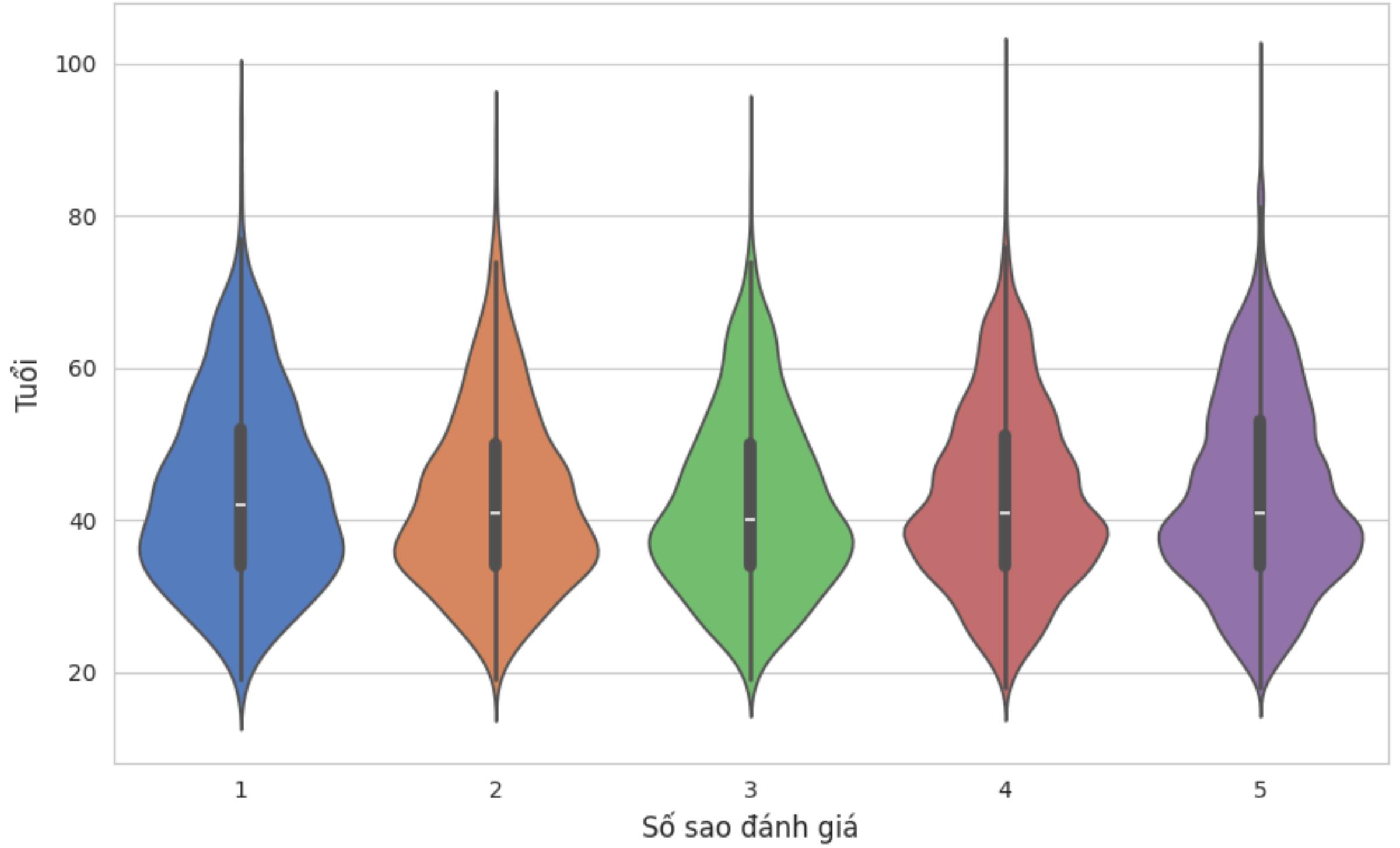
Phân tích nhân khẩu học (Age Analysis)

- Biểu đồ phân bố độ tuổi
- Khách hàng chủ yếu từ 30-50 tuổi
- Mối quan hệ Age vs Rating:
Không có tương quan rõ rệt
(Tuổi tác không ảnh hưởng
nhiều đến việc khó tính hay
dễ tính)

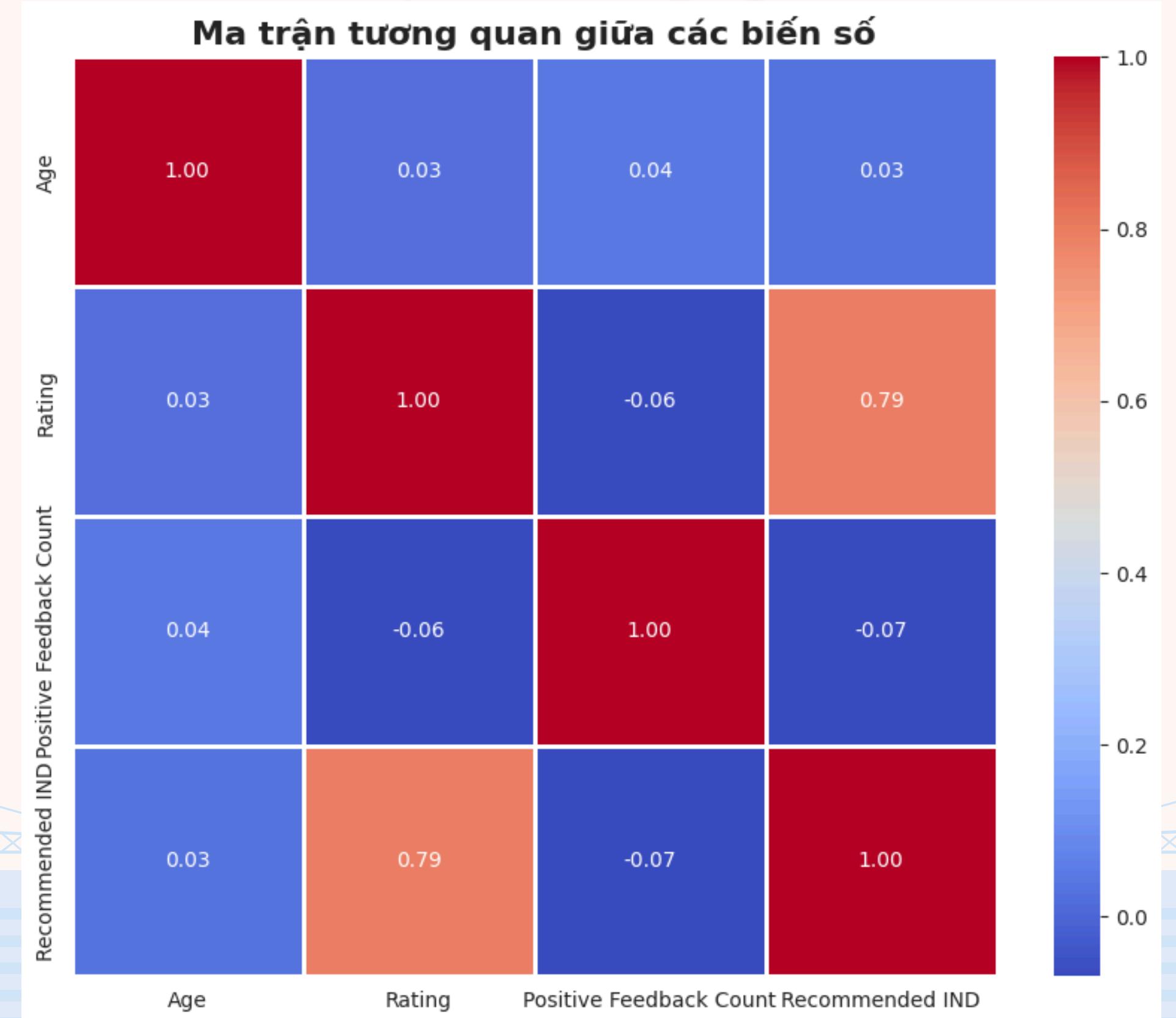


Yếu tố nhân khẩu học

Mật độ Tuổi của khách hàng theo mức Rating

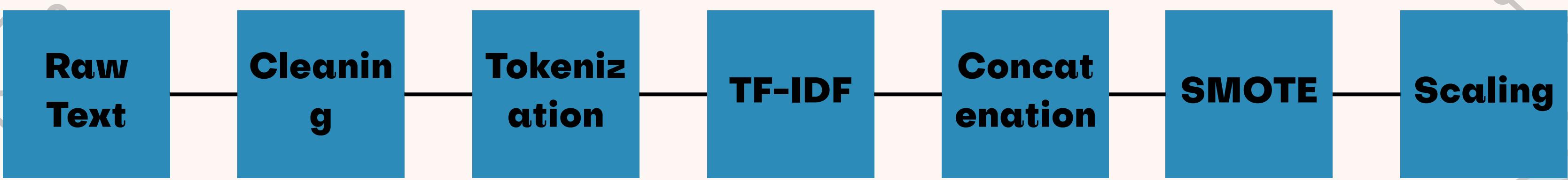


Ma trận tương quan



Phần 3: Tiền xử lý dữ liệu

Sơ đồ quy trình xử lý (Preprocessing Flowchart)



Làm sạch văn bản (Text Cleaning Details)

- **Lowercasing:** Đóng nhất định dạng (Python = python)
- **Regex Cleaning:** Loại bỏ ký tự đặc biệt, số, YRL, HTML tags
- **Stopwords Removal:** Loại bỏ từ nối (the, a, an, in, on, ...) sử dụng thư viện NLTK
- **Stemming/Lemmatization:** Đưa từ về dạng hốc (running → run)

Vector hóa văn bản - TF-IDF

- Khái niệm: Chuyển đổi văn bản sang không gian vector (Vector Space Model)
- Tại sao không dùng Bag-of-Words (Đếm từ)? Vì nó không đánh giá được mức độ quan trọng của từ
- TF-IDF giúp giảm trọng số của những từ xuất hiện quá nhiều nhưng ít thông tin

Công thức toán học TF_IDF

- TF (Term Frequency): Tần suất xuất hiện của từ t trong văn bản d

$$TF(t, d) = \frac{count(t, d)}{TotalWords(d)}$$

- IDF (Inverse Document Frequency): Nghịch đảo tần suất văn bản

$$IDF(t) = \log\left(\frac{N}{df(t)}\right)$$

- TF-IDF:

$$W_{t,d} = TF(t, d) \times IDF(t)$$

Kỹ thuật Feature Engineering (Hybrid Approach)

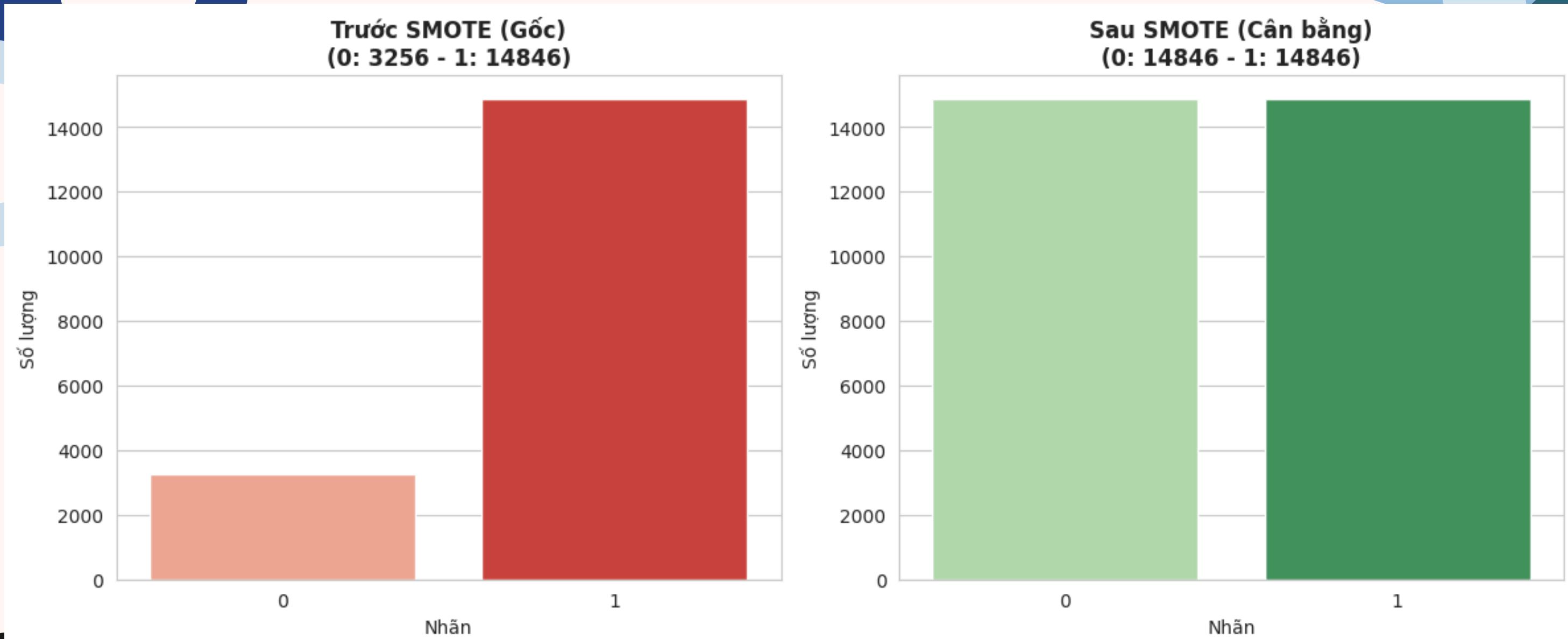
- Chỉ dùng Text là chưa đủ
- Nhóm đề xuất phương án lai (Hubrid): Kết hợp Vector TF-IDF (2000 chiều) + Vector số học (Age, Rating, Feedback Count)
- Mục đích: Tận dụng tối đa thông tin để hỗ trợ mô hình (Ví dụ: Text chưa rõ ràng nhưng Rating thấp → Suy ra là Chê)

Cân bằng dữ liệu - SMOTE

- **Nguyên lý:** Chọn 1 điểm dữ liệu chê (A), tìm hàng xóm (B), và tạo ra điểm mới (M) nằm giwuax đường nối A và B
- **Công thức vector:**

$$X_{new} = X_A + rand(0, 1) \times (X_B - X_A)$$

Giải quyết mất cân bằng - SMOTE



- **Vấn đề:** Tập Train bị lệch (Khen >> Chê)
- **Giải pháp:** SMOTE (Synthetic Minority Over-sampling Technique)
- **Cơ chế:** Không nhân bản (duplicate) đơn thuần. Nó nội suy tuyến tính giữa các điểm dữ liệu thiểu số để tạo ra dữ liệu mới đa dạng hơn

Chuẩn hóa dữ liệu số (Frature Scaling)

- Sử dụng StandardScale cho cột Age và Rating
- Công thức Z-score:

$$z = \frac{x - \mu}{\sigma}$$

- Mục đích: Giúp các thuật toán dựa trên khoảng cách (KNN, SVM) hội tụ nhanh hơn và không bị chi phối bởi đơn vị đo lường(Tuổi 50 > Rating 5)

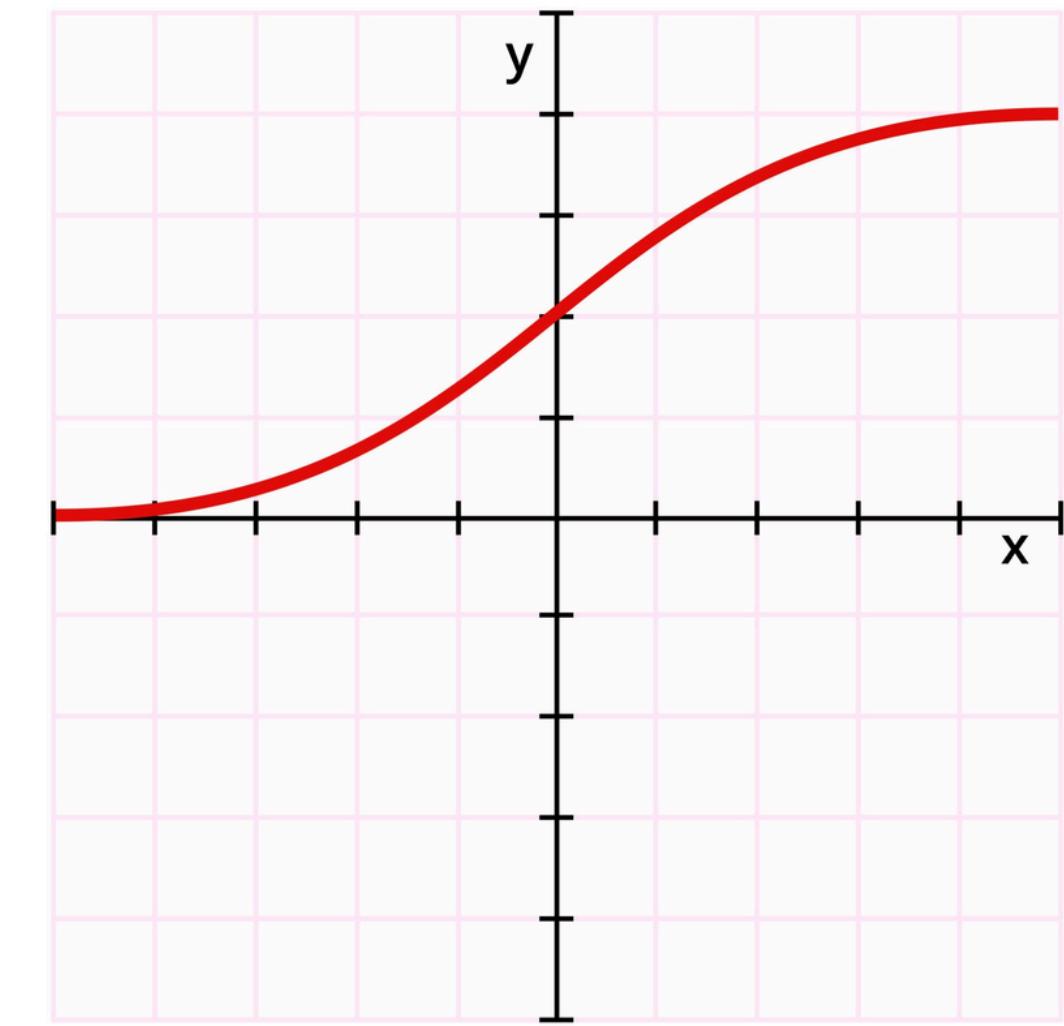
Phần 4: Mô hình hóa & thuật toán

Logistic Regression

- Là mô hình phân loại dựa trên xác suất
- Sử dụng hàm Sigmoid để ép giá trị đầu ra vào khoảng [0,1]
- Công thức

$$P = \frac{1}{1 + e^{-z}}$$

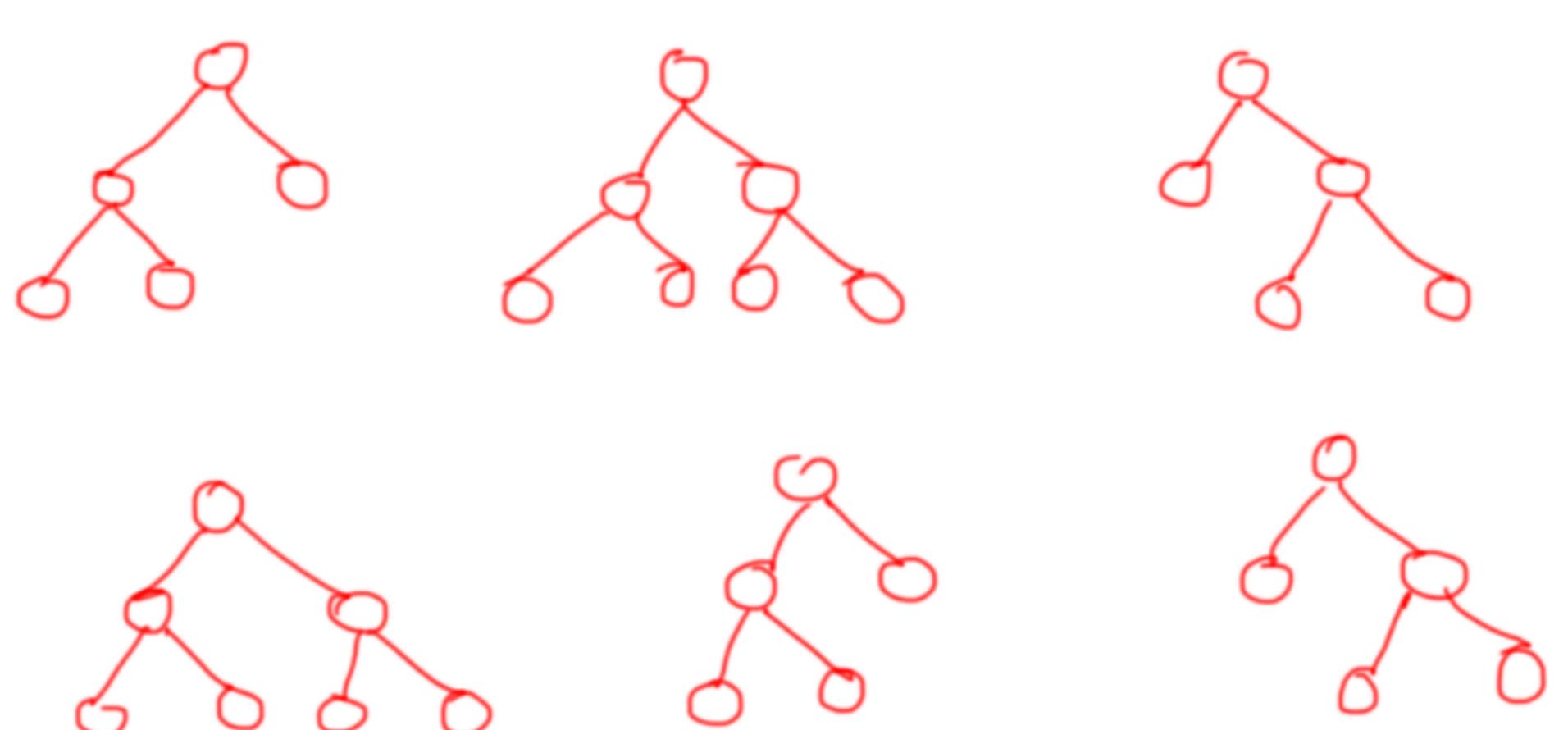
Logistic Function



Ưu/Nhược điểm

- **Ưu:** Đơn giản, dễ giải thích (Interpretability), huấn luyện cực nhanh
- **Nhược:** Giả định ranh giới tuyến tính, khó học được các mối quan hệ phức tạp

Random Forest



- Kỹ thuật **Bagging (Bootstrap Aggregating)**
- Tạo ra N cây quyết định (Decision Trees) từ các tập con ngẫu nhiên của dữ liệu
- Kết quả cuối cùng là cơ chế Bỏ phiếu đa số (Majority Voting)

Ưu/Nhược điểm

- **Ưu:** Độ chính xác cao, chống Overfitting tốt, xử lý tốt dữ liệu bị thiếu hoặc nhiễu
- **Nhược:** Mô hình "Hộp đen" (Black-box), khó giải thích, tốn tài nguyên tính toán

Support Vector Machine (SVM)

- Mục tiêu: Tìm siêu phẳng (Hyperplane) phân cách 2 lớp sao cho lề (Margin) là lớn nhất
- Các điểm dữ liệu nằm trên biên gọi là Support Vectors
- Trong bài toán này dùng **Linear Kernel** vì số chiều đặc trưng (Features) lớn hơn số mẫu (Samples)

SVM

- Hàm quyết định:

$$f(x) = \text{sign}(wx + b)$$

- Bài toán tối ưu hóa (Lagrange Multipliers) để cực đại hóa Margin $\frac{2}{\|w\|}$

Ưu/Nhược điểm

- **Ưu:** Hiệu quả nhất trong không gian nhiều chiều (High-dimensional space) → Phù hợp nhất cho phân loại văn bản.
- **Nhược:** Thời gian huấn luyện lâu với tập dữ liệu lớn, nhạy cảm với nhiễu.

K-Nearest Neighbors (KNN)

- Thuật toán lười (Lazy Learning): Không học tham số, chỉ lưu dữ liệu.
- Khi dự đoán: Tính khoảng cách từ điểm mới tới k điểm gần nhất.
- Công thức khoảng cách Euclid:

$$d(p, q) = \sqrt{\sum (p_i - q_i)^2}$$

Ưu/Nhược điểm & Tại sao thất bại?

- **Ưu:** Đơn giản, không cần huấn luyện.
- **Nhược:** Chi phí dự đoán đắt (tính toán với toàn bộ dữ liệu).
- **Vấn đề trong bài toán này:** Lời nguyền số chiều (Curse of Dimensionality). Trong không gian TF-IDF (2000 chiều), khái niệm "gần nhau" trở nên vô nghĩa, làm giảm độ chính xác.

Phần 5: Kết quả

Thiết lập

- Chia tập dữ liệu: 80% Train (có SMOTE) - 20% Test (Gốc).
- Random State: 42.
- Môi trường: Google Colab.
- Thư viện: Scikit-learn.

Các chỉ số đánh giá

- **Accuracy:** Tổng số đoán đúng / Tổng số mẫu.

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

- **Precision:** Đoán là khen thì đúng bao nhiêu?

$$Precision = \frac{TP}{TP + FP}$$

- **Recall:** Có bao nhiêu lời khen thực tế được tìm thấy?

$$Recall = \frac{TP}{TP + FN}$$

- **F1-Score:**
dữ liệu lệch).

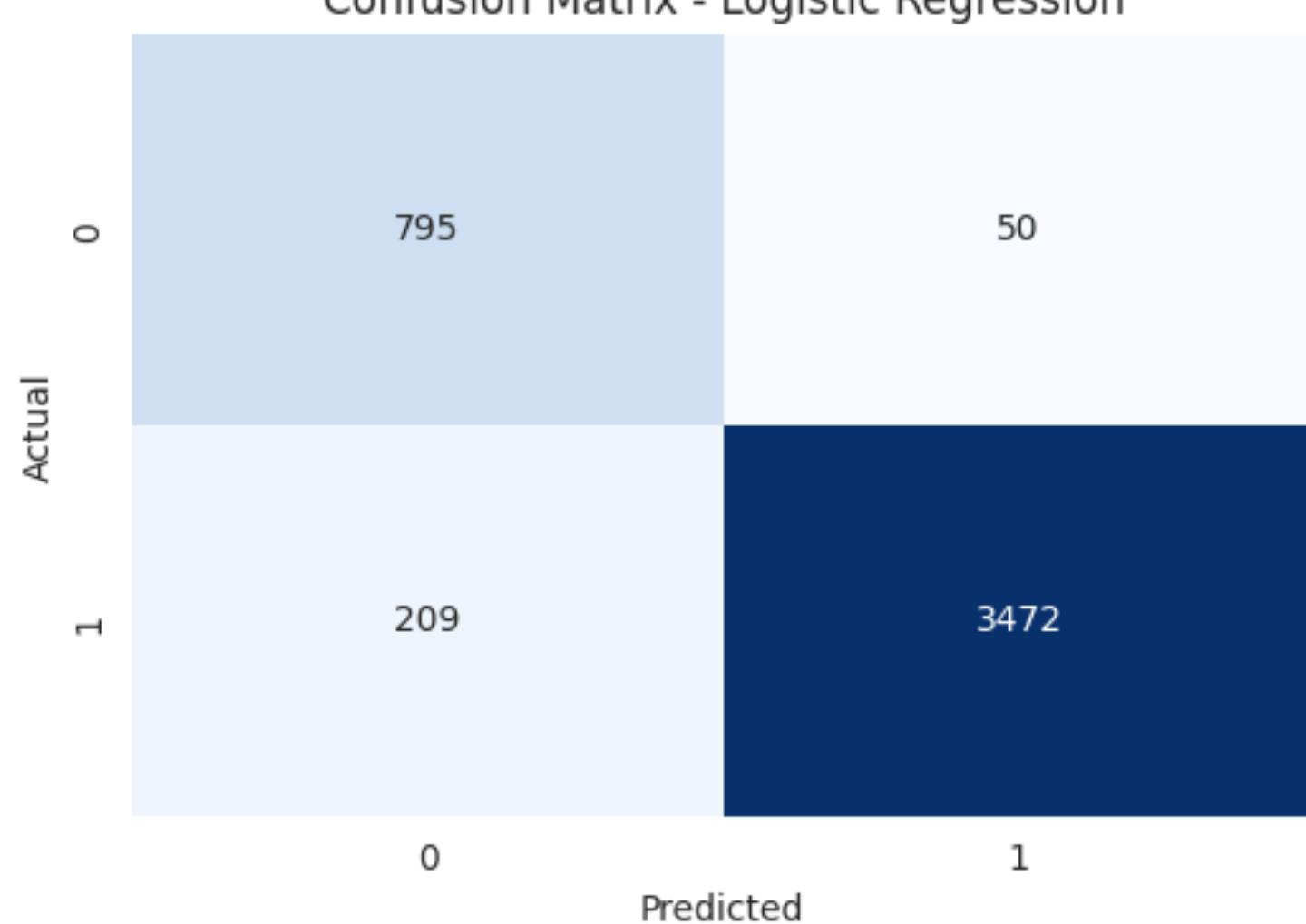
$$2 \times \frac{Precision \times Recall}{Precision + Recall}$$

(Quan trọng nhất vì

Kết quả- Logistic Regression

MODEL: Logistic Regression					
Accuracy: 0.9428					
	precision	recall	f1-score	support	
0	0.79	0.94	0.86	845	
1	0.99	0.94	0.96	3681	
accuracy			0.94	4526	
macro avg	0.89	0.94	0.91	4526	
weighted avg	0.95	0.94	0.94	4526	

Confusion Matrix - Logistic Regression

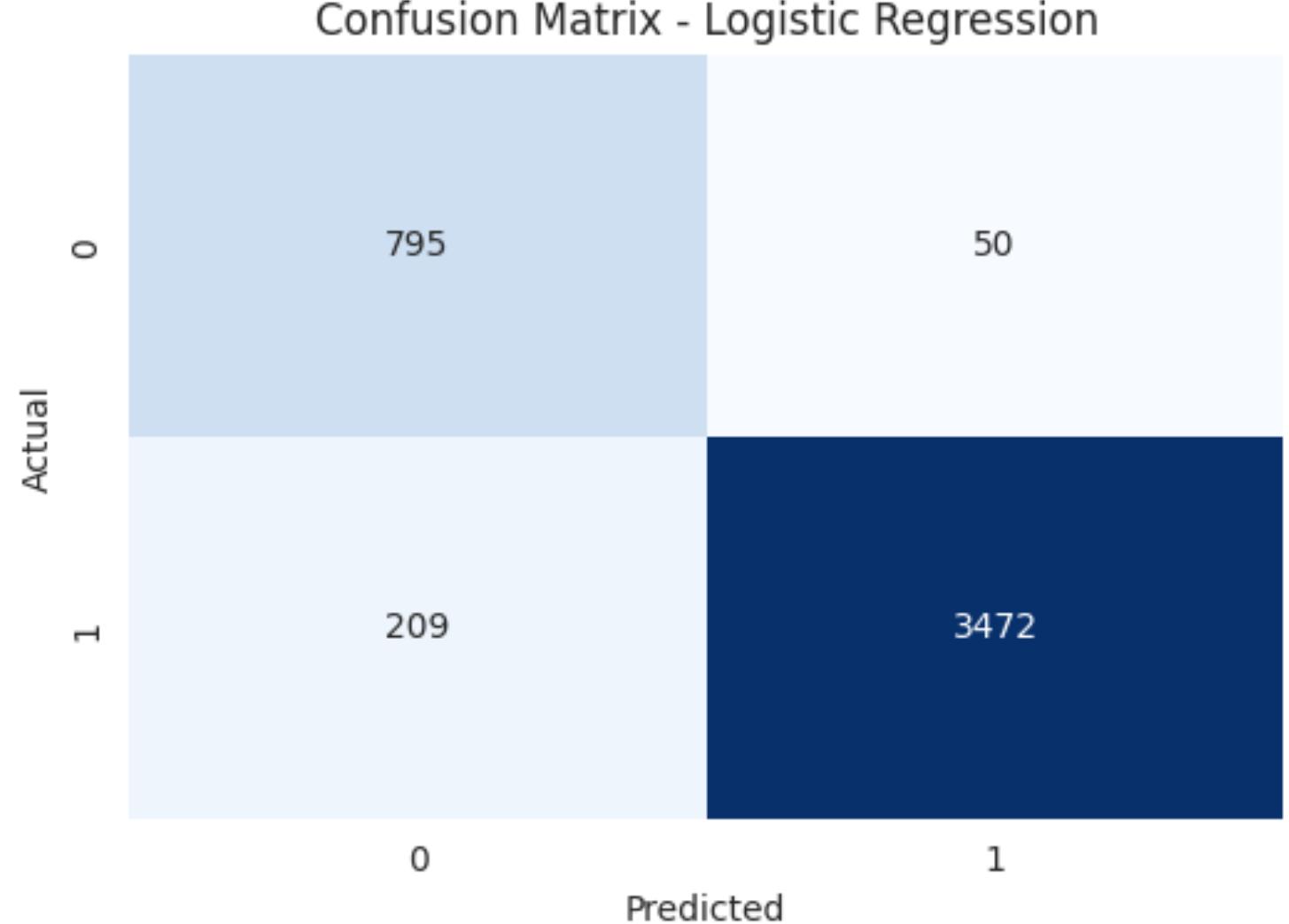


- Độ chính xác (Accuracy): 94.28% (Cao nhất).
- Điểm mạnh: Cân bằng tốt nhất giữa Precision và Recall. F1-score của cả 2 lớp đều rất cao (0.86 và 0.96).
- Đánh giá: Đây là mô hình "Baseline" nhưng lại chiến thắng tất cả các mô hình phức tạp khác.

Kết quả- Logistic Regression

MODEL: Logistic Regression					
Accuracy: 0.9428					
	precision	recall	f1-score	support	
0	0.79	0.94	0.86	845	
1	0.99	0.94	0.96	3681	
accuracy			0.94	4526	
macro avg	0.89	0.94	0.91	4526	
weighted avg	0.95	0.94	0.94	4526	

Confusion Matrix - Logistic Regression



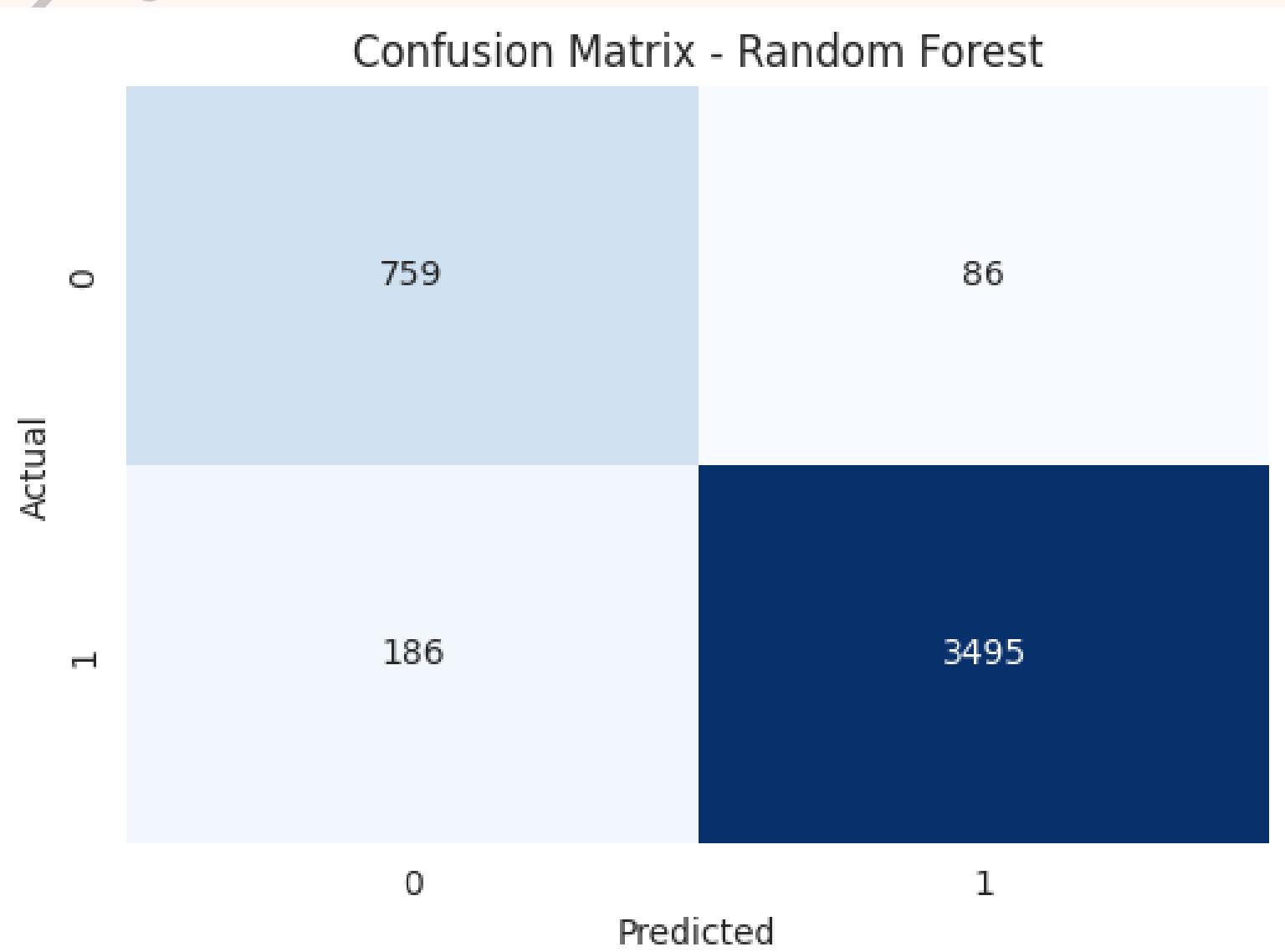
Tại sao nó thắng??

- Phù hợp với dữ liệu thưa (Sparsity): Dữ liệu văn bản (TF-IDF) chứa rất nhiều số 0. Logistic Regression xử lý cực tốt dạng này vì nó bỏ qua các trọng số bằng 0, tính toán nhanh và chính xác.
- Tính phân tách tuyến tính: Trong không gian cao chiều (hàng nghìn từ), các điểm dữ liệu văn bản thường dễ dàng được phân tách bằng một đường thẳng (hoặc mặt phẳng). Logistic Regression sinh ra là để tìm đường thẳng này.

Kết quả - Random Forest

MODEL: Random Forest					
Accuracy: 0.9399					
	precision	recall	f1-score	support	
0	0.80	0.90	0.85	845	
1	0.98	0.95	0.96	3681	
accuracy			0.94	4526	
macro avg	0.89	0.92	0.91	4526	
weighted avg	0.94	0.94	0.94	4526	

Confusion Matrix - Random Forest

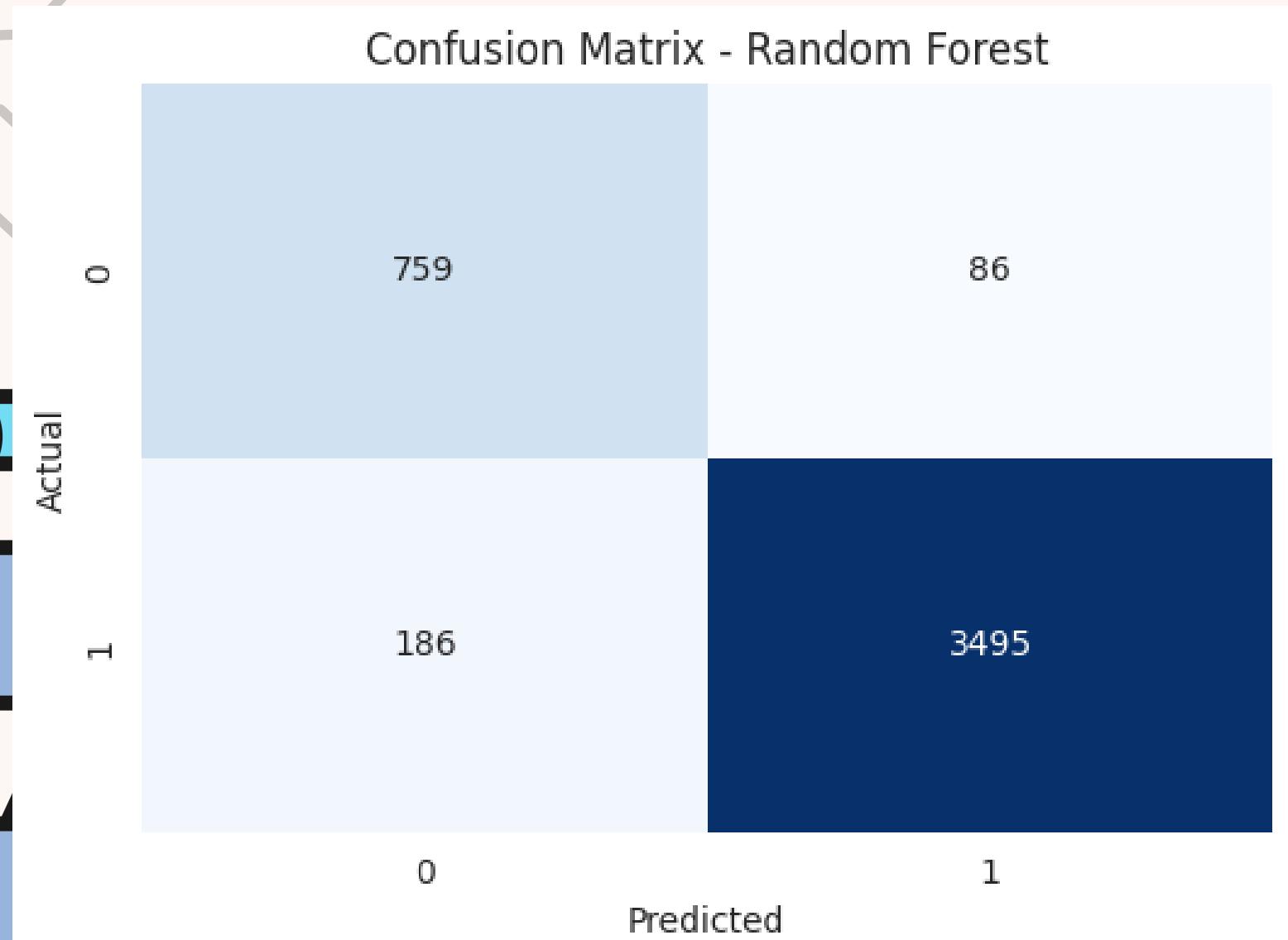


- Độ chính xác (Accuracy): 93.99% (Thua Logistic Regression một chút xíu ~0.3%).
- Điểm mạnh: Precision của lớp 0 (0.80) nhỉnh hơn Logistic một chút, cho thấy khi nó đã phán là "Tê" thì rất đáng tin.
- Điểm yếu: Bỏ sót nhiều trường hợp lớp 0 hơn (Recall thấp hơn Logistic: 0.90 so với 0.94).

Kết quả - Random Forest

MODEL: Random Forest					
Accuracy: 0.9399					
	precision	recall	f1-score	support	
0	0.80	0.90	0.85	845	
1	0.98	0.95	0.96	3681	
accuracy			0.94	4526	
macro avg	0.89	0.92	0.91	4526	
weighted avg	0.94	0.94	0.94	4526	

Confusion Matrix - Random Forest



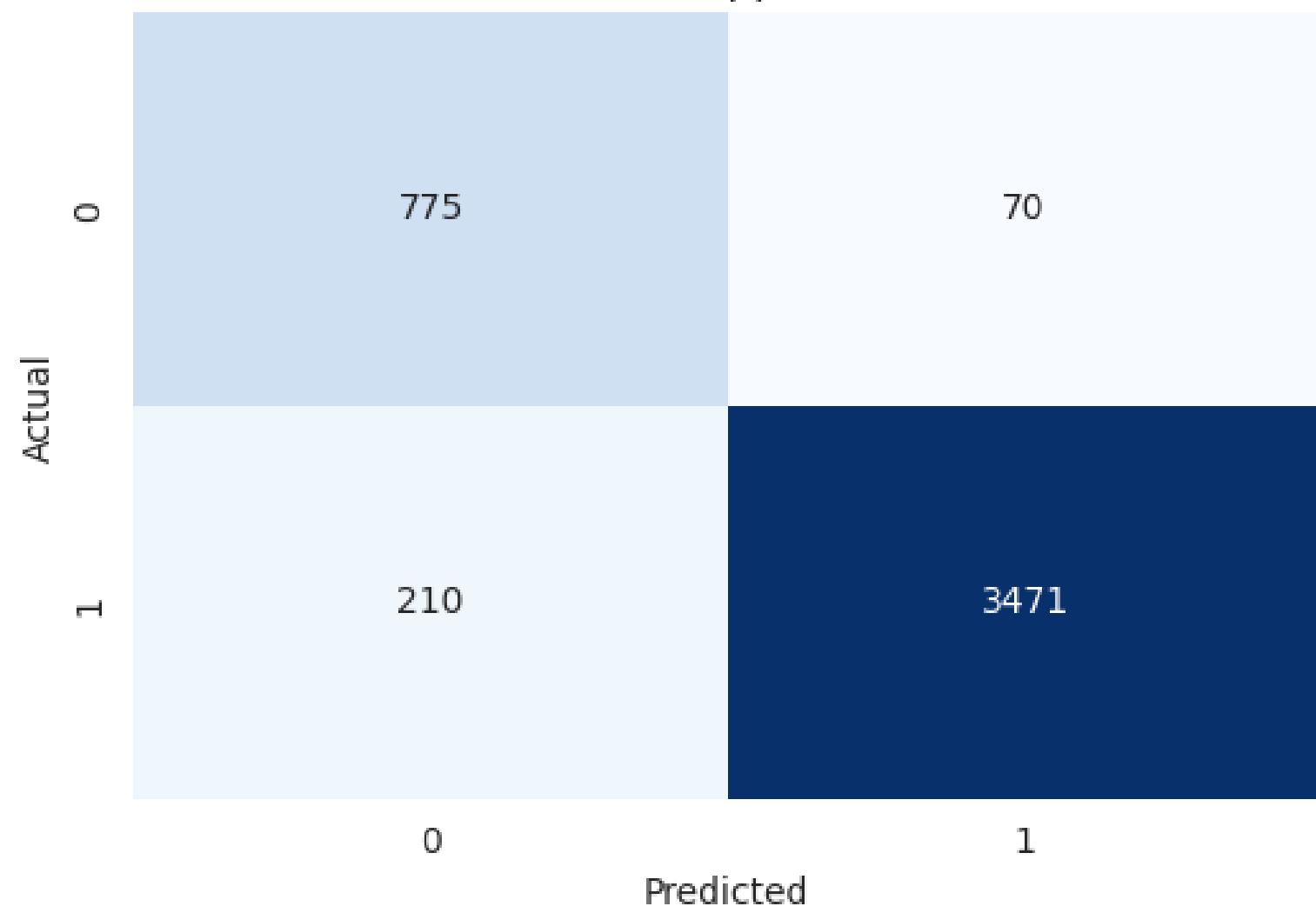
Tại sao??

- Sức mạnh đam mê: Việc kết hợp hàng trăm cây quyết định giúp mô hình chống lại nhiễu tốt, giữ độ chính xác rất cao (~94%).
- Hạn chế: Với dữ liệu văn bản vốn dĩ đã tách biệt khá rõ ràng (Linear Separable), khả năng cắt gọt phi tuyến tính phức tạp của Random Forest trở nên "thùa thãi", không tạo ra lợi thế vượt trội so với đường thẳng đơn giản của Logistic Regression.

Kết quả - SVM

MODEL: Support Vector Machine				
Accuracy: 0.9381				
	precision	recall	f1-score	support
0	0.79	0.92	0.85	845
1	0.98	0.94	0.96	3681
accuracy			0.94	4526
macro avg	0.88	0.93	0.90	4526
weighted avg	0.94	0.94	0.94	4526

Confusion Matrix - Support Vector Machine

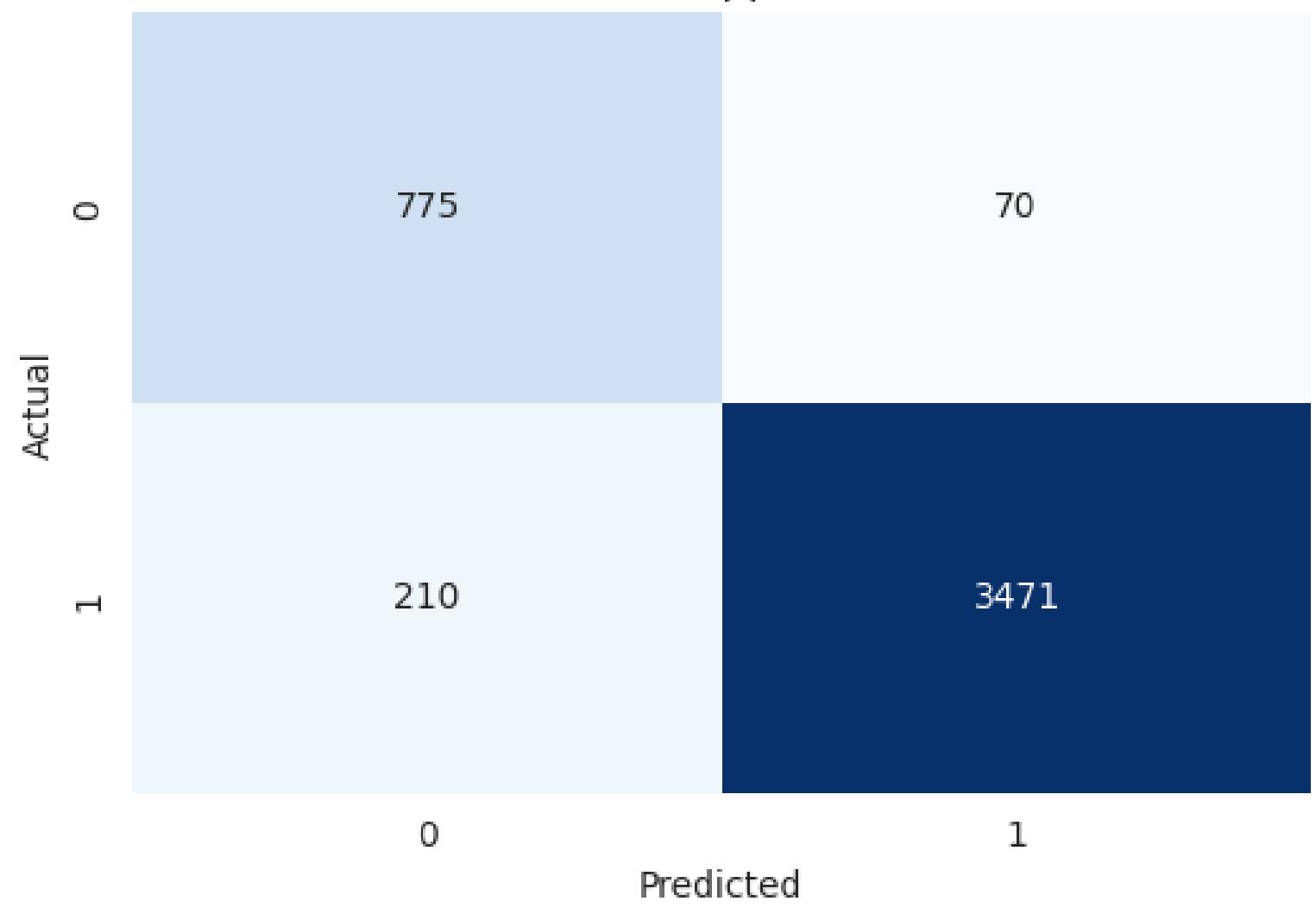


- Độ chính xác (Accuracy): 93.81%.
- Đánh giá: Kết quả gần như sao y bản chính của Logistic Regression (chỉ lệch vài phần nghìn).
- Chỉ số phụ: Precision và Recall tương đương với Logistic, không có sự chênh lệch đáng kể.

Kết quả - SVM

MODEL: Support Vector Machine				
Accuracy: 0.9381				
	precision	recall	f1-score	support
0	0.79	0.92	0.85	845
1	0.98	0.94	0.96	3681
accuracy			0.94	4526
macro avg	0.88	0.93	0.90	4526
weighted avg	0.94	0.94	0.94	4526

Confusion Matrix - Support Vector Machine



Tại sao nó lại giống với LR?

- Bản chất hình học: SVM làm việc cực tốt trong không gian nhiều chiều (High-dimensional space) như TF-IDF.
- Kernel: Trong bài toán phân loại văn bản, SVM thường sử dụng Linear Kernel. Lúc này, cách nó tìm ranh giới phân loại (Hyperplane) về mặt toán học là tương tự như Logistic Regression, nên kết quả đầu ra gần như tương đương nhau.

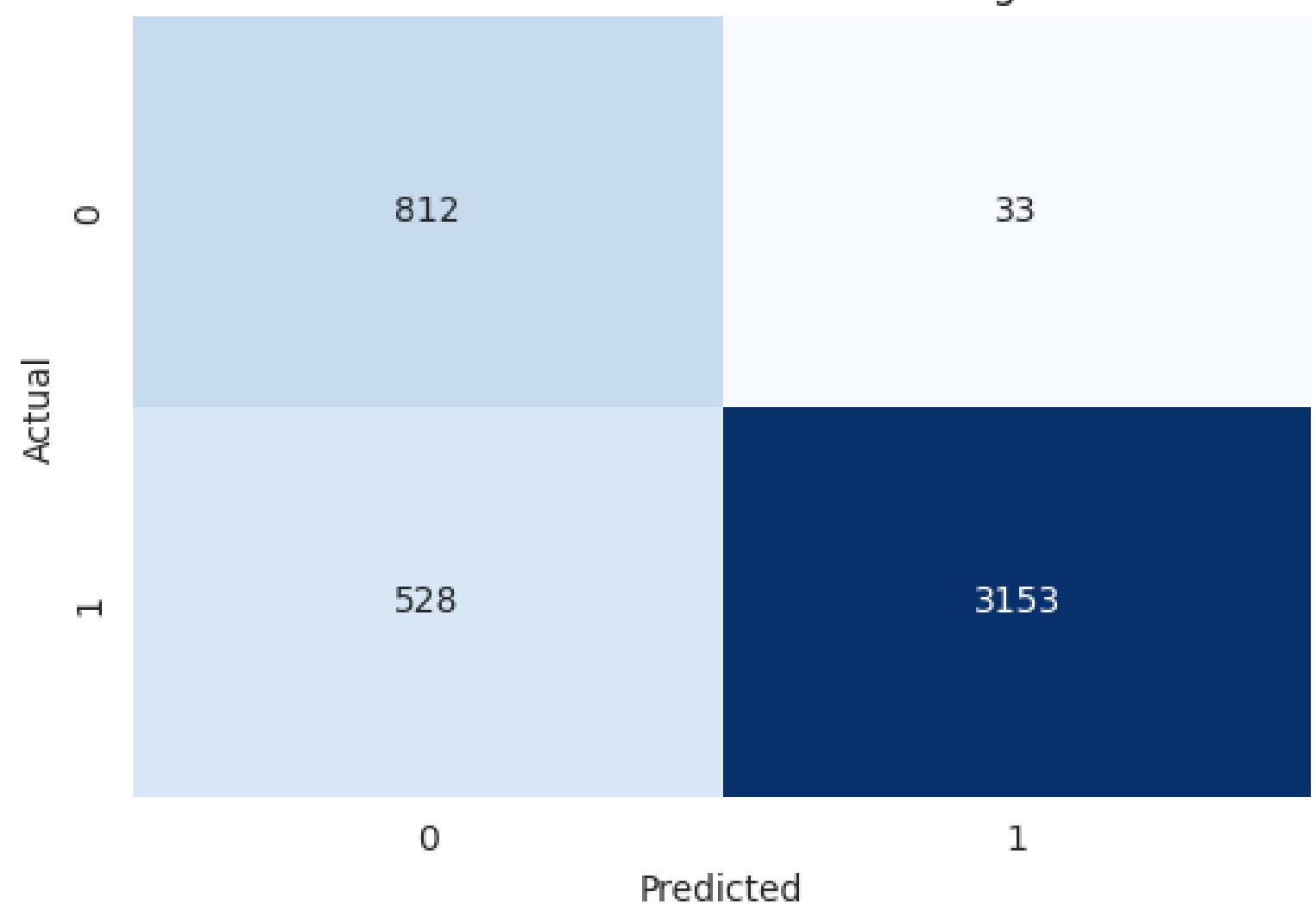
Kết quả - KNN

MODEL: K-Nearest Neighbors

Accuracy: 0.8760

	precision	recall	f1-score	support
0	0.61	0.96	0.74	845
1	0.99	0.86	0.92	3681
accuracy			0.88	4526
macro avg	0.80	0.91	0.83	4526
weighted avg	0.92	0.88	0.89	4526

Confusion Matrix - K-Nearest Neighbors

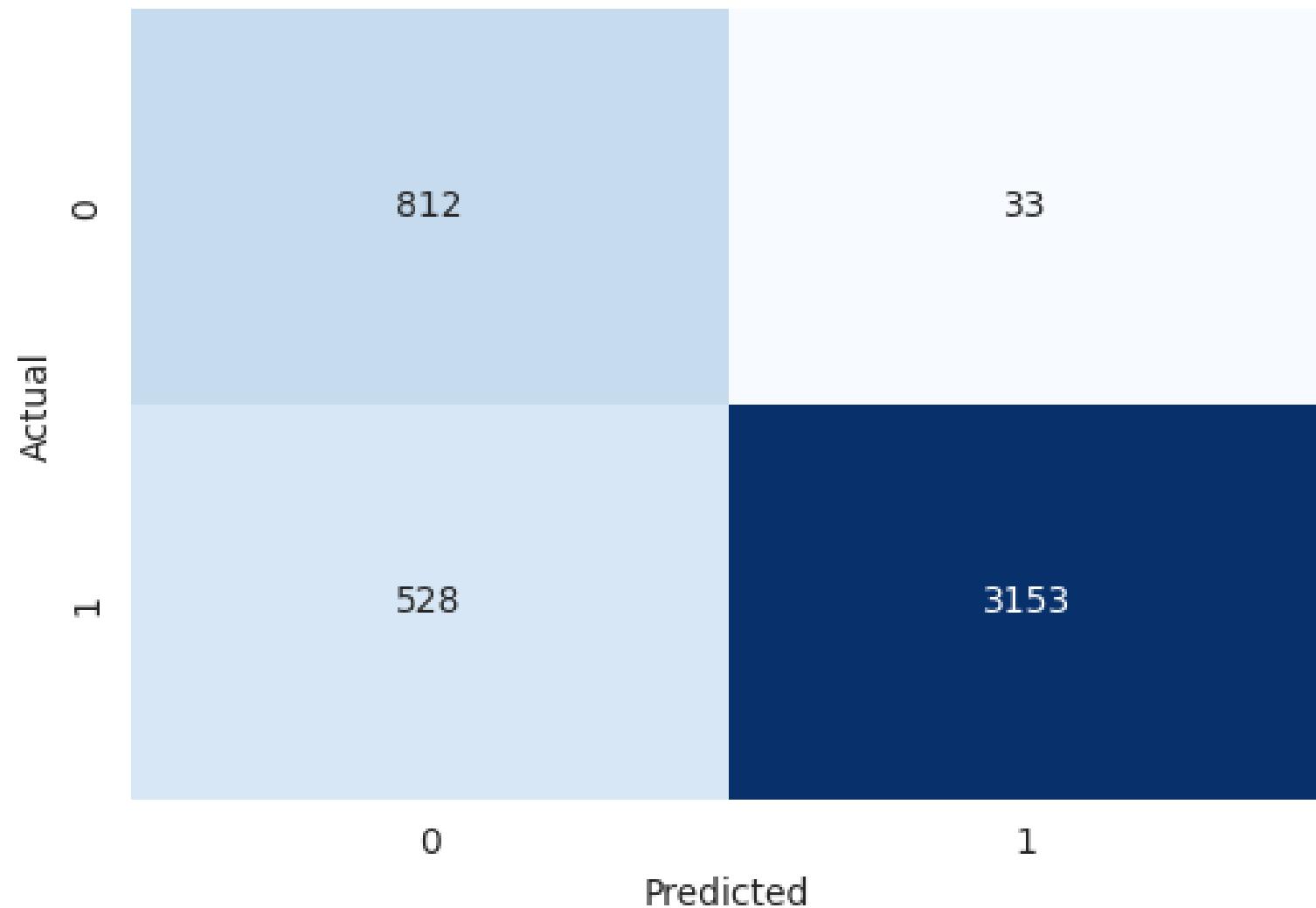


- Độ chính xác (Accuracy): 87.60% (Thấp nhất, bị bỏ xa so với nhóm trên).
- Vấn đề nghiêm trọng:
- Recall lớp 0 rất cao (0.96): Bắt nhầm rất nhiều.
- Precision lớp 0 rất thấp (0.61): Dự đoán sai 528 trường hợp (False Positive cao độ biến). Tức là rất nhiều review Tốt bị nó đoán nhầm thành Tệ.

Kết quả - KNN

MODEL: K-Nearest Neighbors					
	precision	recall	f1-score	support	
0	0.61	0.96	0.74	845	
1	0.99	0.86	0.92	3681	
accuracy			0.88	4526	
macro avg	0.80	0.91	0.83	4526	
weighted avg	0.92	0.88	0.89	4526	

Confusion Matrix - K-Nearest Neighbors



Tại sao nó thất bại??

- Lời nguyền số chiều (Curse of Dimensionality): Trong không gian hàng nghìn chiều của vector văn bản, khái niệm "khoảng cách" giữa các điểm trở nên vô nghĩa. Các điểm dữ liệu đều nằm rất xa nhau, khiến việc tìm "hàng xóm" thiếu chính xác.
- Nhiều: KNN rất nhạy cảm với nhiều cục bộ. Chỉ cần vài điểm dữ liệu ngoại lai nằm sai chỗ cũng kéo lêch kết quả dự đoán của cả vùng.

Bảng tổng hợp so sánh

Tiêu chí	Logistic Regression	Random Forest	SVM	KNN
Accuracy (Tổng thể)	94.28% (Top 1)	93.99% (Top 2)	93.81% (Top 3)	87.60% (Thấp nhất)
Precision (Class 0)	0.79	0.80	0.79	0.61 (Rất tệ)
Recall (Class 0)	0.94	0.90	0.92	0.96 (Cao nhất)
Số sai đoán sai (Class 0)	Bỏ sót: 50 Nhầm: 209	Bỏ sót: 86 Nhầm: 186	Bỏ sót: 70 Nhầm: 210	Bỏ sót: 33 Nhầm: 528
Đánh giá bản chất	Cân bằng nhất. Hiệu quả tuyệt đối trên dữ liệu thưa (Sparse Data).	Thận trọng nhất. Precision cao nhất nghĩa là ít đoán bừa, nhưng lại bỏ sót (Recall thấp) hơn LR.	Tương đồng LR. Chứng minh dữ liệu phân tách tốt tuyến tính trong không gian nhiều chiều.	Mất cân đối. Recall cao ảo (do đoán toàn bộ là 0), dẫn đến đoán nhầm (False Positive) quá nhiều.

Phần 6: Tổng kết

Tổng kết

- Đã xây dựng thành công pipeline xử lý dữ liệu chuẩn
- Chứng minh SVM là thuật toán tốt nhất cho bài toán này
- Khẳng định vai trò không thể thiếu của SMOTE trong việc xử lý dữ liệu thực tế