BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

**----------**

****

ĐỒ ÁN CUỐI KỲ MÔN HỌC

**PHÂN TÍCH DỮ LIỆU LỚN**

**ĐỀ TÀI:**

**XÂY DỰNG HỆ THỐNG PHÂN TÍCH CẢM XÚC BÌNH LUẬN CỦA VIDEO TRÊN NỀN TẢNG YOUTUBE**

**GVHD: Lê Thị Minh Châu**

**Lớp HP:** BDAN333977

**Nhóm thực hiện:** *Nhóm 14*

**Học kỳ:** II

**Năm học:** *2024 – 2025*

*Thành phố Hồ Chí Minh, tháng 5 năm 2025*

**Mục Lục**

[**CHƯƠNG I: TỔNG QUAN VỀ ĐỀ TÀI**](#_g096h1pa2pp6) **1**

[1. Lý do chọn đề tài.](#_kjde21c72l6c) 1

[2. Mục tiêu nghiên cứu](#_jchsesw5pniu) 1

[3. Ý nghĩa của nghiên cứu.](#_kzvlihx28d0k) 2

[4. Đối tượng và phạm vi nghiên cứu](#_26yr0r7zd8uh) 2

[**CHƯƠNG II. CƠ SỞ LÝ THUYẾT**](#_126c9r335we6) **3**

[1. Giới thiệu Big Data](#_zh431iu0jk04) 3

[2. Tổng quan về hệ thống](#_1wovavsq9wy3) 4

[**CHƯƠNG III: TRIỂN KHAI BÀI TOÁN**](#_c9s8hx4os0f8) **6**

[1. Đăng ký Youtube API](#_jz5io19ya43o) 6

[1.1. Tạo và kích hoạt dự án trên Google Cloud Console](#_5zz11ngty5n9) 6

[1.2. Tạo API Key](#_c6gsnoxew0ov) 7

[1.3. Kiểm tra API Key](#_p617lfxrtkqp) 8

[1.4. Lưu ý khi sử dụng YouTube API](#_45szp1xt1wsk) 10

[2. Thực hiện lấy dữ liệu Với Apache Kafka](#_9tucerrq2ia) 10

[3. Tiền xử lý dữ liệu](#_oe8n9w2mgi3c) 18

[4. Phân tích cảm xúc:](#_8vup39lp2wpa) 25

[5. Ứng dụng mô hình vào một chủ đề cụ thể để dự đoán cảm xúc](#_mq9s2uuijw09) 33

[**CHƯƠNG IV. HUẤN LUYỆN MÔ HÌNH PHÂN TÍCH CẢM XÚC**](#_pviyfdf67j56) **42**

[**CHƯƠNG V: TRỰC QUAN HÓA DỮ LIỆU VỚI POWER BI**](#_kvg3blcxpvw4) **45**

[**CHƯƠNG VI: KẾT LUẬN**](#_rlpmcbz99286) **47**

[1. Những kết quả đạt được](#_itlfkkpfeclp) 47

[2. Hạn chế của đề tài](#_3qbul1pc2c8d) 48

[3. Hướng phát triển](#_eizhsppl1re) 49

[**TÀI LIỆU THAM KHẢO**](#_i62xucaro4fe) **49**

**BẢNG PHÂN CÔNG NHIỆM VỤ**

| STT | Họ và tên | MSSV | Nhiệm vụ | Hoàn thành |
| --- | --- | --- | --- | --- |
| 1 | Nguyễn Quang Hoàng Phát | 22133040 | - Thực hiện luồng: Streaming data -> Kafka topic -> Spark Streaming -> Trained Models -> Predict data  - Viết báo cáo  - Hỗ trợ thực hiện các phần khác | 100% |
| 2 | Trần Bảo Việt | 22133065 | - Huấn luyện mô hình  - Trực quan hóa với Power BI - Hỗ trợ viết báo cáo | 100% |
| 3 | Lê Quỳnh Nhựt Vinh | 22133066 | - Thực hiện luồng: Training data -> Kafka topic -> Cassandra -> Spark model | 100% |

**LỜI CẢM ƠN**

Để hoàn thành được bài tiểu luận này, em xin chân thành cảm ơn quý thầy cô của trường những người đã tận tình giúp đỡ và tạo điều kiện cho em trong quá trình học tập. Đặc biệt, em xin gửi lời cảm ơn sâu sắc đến thầy/cô Lê Thị Minh Châu. - người đã trực tiếp giảng dạy và hướng dẫn em thực hiện bài tiểu luận này bằng tất cả lòng nhiệt tình và sự quan tâm sâu sắc.

Trong quá trình thực hiện bài tiểu luận này, do hiểu biết còn nhiều hạn chế nên bài làm khó tránh khỏi những thiếu sót. Em rất mong nhận được những lời góp ý của quý thầy cô để bài tiểu luận ngày càng hoàn thiện hơn.

Em xin chân thành cảm ơn.

# **CHƯƠNG I: TỔNG QUAN VỀ ĐỀ TÀI**

## 1. Lý do chọn đề tài.

Trong thời đại công nghệ số, YouTube đã trở thành một trong những nền tảng mạng xã hội phổ biến nhất, thu hút hàng tỷ người dùng trên toàn cầu. Mỗi ngày, hàng triệu video được tải lên cùng với hàng tỷ bình luận từ người xem, phản ánh ý kiến, cảm xúc và xu hướng của cộng đồng. Những bình luận này không chỉ là nguồn dữ liệu phong phú mà còn chứa đựng thông tin giá trị về cảm xúc của người xem, bao gồm tích cực, tiêu cực hoặc trung lập. Thông tin này có thể giúp các nhà sản xuất nội dung, doanh nghiệp và nhà nghiên cứu hiểu rõ hơn về phản ứng của khán giả đối với các video.

Tuy nhiên, với khối lượng dữ liệu bình luận khổng lồ, việc phân tích thủ công là bất khả thi do tốn thời gian và công sức. Công nghệ Big Data, kết hợp với các kỹ thuật xử lý ngôn ngữ tự nhiên (NLP) và học máy, cung cấp giải pháp để tự động hóa quá trình phân tích cảm xúc, giúp tiết kiệm thời gian và mang lại kết quả chính xác. Đề tài "Xây dựng hệ thống phân tích cảm xúc bình luận của video trên nền tảng YouTube" được chọn vì tính thực tiễn cao, đáp ứng nhu cầu khai thác dữ liệu lớn để hỗ trợ các lĩnh vực như marketing, sản xuất nội dung và nghiên cứu thị trường.

Ngoài ra, việc ứng dụng các công nghệ tiên tiến như Apache Kafka, Spark Streaming và Cassandra trong xử lý dữ liệu thời gian thực là một hướng đi phù hợp với xu thế phát triển của ngành công nghệ thông tin. Đề tài không chỉ giúp nhóm sinh viên áp dụng kiến thức lý thuyết mà còn rèn luyện kỹ năng thực tiễn, chuẩn bị cho các dự án thực tế trong tương lai.

## 2. Mục tiêu nghiên cứu

Mục tiêu chính của đề tài là xây dựng một hệ thống phân tích cảm xúc bình luận của video trên YouTube, sử dụng các công nghệ Big Data để đảm bảo hiệu quả và độ chính xác cao. Các mục tiêu cụ thể bao gồm:

* Thu thập dữ liệu bình luận từ YouTube thông qua YouTube Data API v3 và truyền dữ liệu thời gian thực bằng Apache Kafka.
* Tiền xử lý dữ liệu bình luận, bao gồm làm sạch văn bản, phát hiện ngôn ngữ, dịch sang tiếng Anh (nếu cần) và chuẩn hóa dữ liệu để phù hợp với mô hình phân tích.
* Xây dựng và huấn luyện mô hình học máy để phân loại cảm xúc (tích cực, tiêu cực, trung lập) dựa trên nội dung bình luận.
* Lưu trữ dữ liệu thô và kết quả phân tích vào cơ sở dữ liệu Cassandra để đảm bảo khả năng truy xuất nhanh và quản lý hiệu quả.
* Trực quan hóa kết quả phân tích bằng Apache Superset, cung cấp các biểu đồ và báo cáo trực quan để hỗ trợ người dùng hiểu rõ xu hướng cảm xúc.

Hệ thống được thiết kế để xử lý luồng dữ liệu thời gian thực, đảm bảo khả năng mở rộng và hiệu suất cao trong môi trường thực tế.

## 3. Ý nghĩa của nghiên cứu.

Đề tài giúp củng cố và áp dụng kiến thức về Big Data, xử lý ngôn ngữ tự nhiên và học máy vào một bài toán thực tế. Việc sử dụng các công nghệ như Apache Kafka, Spark Streaming và Cassandra giúp nhóm sinh viên hiểu sâu hơn về quy trình xử lý dữ liệu lớn, từ thu thập, xử lý đến lưu trữ và trực quan hóa. Đề tài cũng tạo nền tảng để nghiên cứu thêm về các thuật toán phân tích cảm xúc và cải thiện hiệu suất hệ thống Big Data.

Hệ thống phân tích cảm xúc bình luận YouTube có tiềm năng ứng dụng trong nhiều lĩnh vực. Các nhà sản xuất nội dung có thể sử dụng hệ thống để đánh giá phản hồi của khán giả, từ đó tối ưu hóa nội dung video. Doanh nghiệp có thể khai thác dữ liệu cảm xúc để phân tích thị trường, hiểu rõ nhu cầu và cảm nhận của khách hàng. Ngoài ra, hệ thống còn hỗ trợ các nhà nghiên cứu trong các lĩnh vực như xã hội học và tâm lý học để phân tích hành vi và cảm xúc của cộng đồng trên các nền tảng trực tuyến.

## 4. Đối tượng và phạm vi nghiên cứu

Đề tài tập trung vào phân tích cảm xúc của các bình luận từ video YouTube, bao gồm bình luận bằng nhiều ngôn ngữ khác nhau. Các cảm xúc được phân loại gồm tích cực, tiêu cực và trung lập, dựa trên nội dung văn bản của bình luận.

Dữ liệu được thu thập từ các video YouTube thuộc các chủ đề cụ thể như công nghệ, âm nhạc, tin tức thông qua YouTube Data API v3. Dữ liệu bao gồm cả luồng bình luận thời gian thực (streaming data) và dữ liệu lịch sử (training data) để huấn luyện mô hình.

Hệ thống sử dụng Apache Kafka để truyền dữ liệu, Apache Spark (bao gồm Spark Streaming) để xử lý và phân tích dữ liệu, Cassandra để lưu trữ dữ liệu, và Apache Superset để trực quan hóa kết quả.

Đề tài được thực hiện trong khuôn khổ đồ án cuối kỳ môn học Phân tích Dữ liệu Lớn, thuộc học kỳ II, năm học 2024–2025.

Hệ thống tập trung vào phân tích cảm xúc bình luận video YouTube, với khả năng mở rộng để áp dụng cho các nền tảng mạng xã hội khác trong tương lai.

# **CHƯƠNG II. CƠ SỞ LÝ THUYẾT**

## 1. Giới thiệu Big Data

Big Data (Dữ liệu lớn) là thuật ngữ dùng để chỉ các tập hợp dữ liệu có khối lượng lớn, phức tạp và tốc độ tăng trưởng nhanh, vượt quá khả năng xử lý của các công cụ và phương pháp truyền thống. Big Data không chỉ đề cập đến khối lượng dữ liệu mà còn bao gồm tốc độ xử lý dữ liệu, sự đa dạng của dữ liệu, độ chính xác và giá trị kinh tế hoặc thông tin mà dữ liệu mang lại. Những đặc điểm này được gọi là mô hình 5V, là nền tảng để hiểu và khai thác Big Data trong các ứng dụng thực tiễn.

Trong bối cảnh của đề tài, dữ liệu bình luận từ YouTube là một ví dụ điển hình của Big Data. Mỗi ngày, hàng triệu bình luận được tạo ra với tốc độ nhanh, thuộc nhiều ngôn ngữ, định dạng và mang nội dung đa dạng. Những bình luận này có thể chứa thông tin cảm xúc, ý kiến hoặc xu hướng của người dùng, nhưng việc xử lý khối lượng dữ liệu khổng lồ này đòi hỏi các công nghệ tiên tiến như Apache Kafka, Spark Streaming và Cassandra.

Big Data đóng vai trò quan trọng trong việc phân tích và khai thác thông tin từ các nguồn dữ liệu lớn, đặc biệt trong các lĩnh vực như phân tích cảm xúc, dự đoán xu hướng và hỗ trợ ra quyết định. Các công nghệ Big Data cho phép thu thập, lưu trữ, xử lý và phân tích dữ liệu theo thời gian thực hoặc theo lô (batch), đảm bảo hiệu suất cao và khả năng mở rộng. Trong hệ thống phân tích cảm xúc bình luận YouTube, Big Data được ứng dụng thông qua:

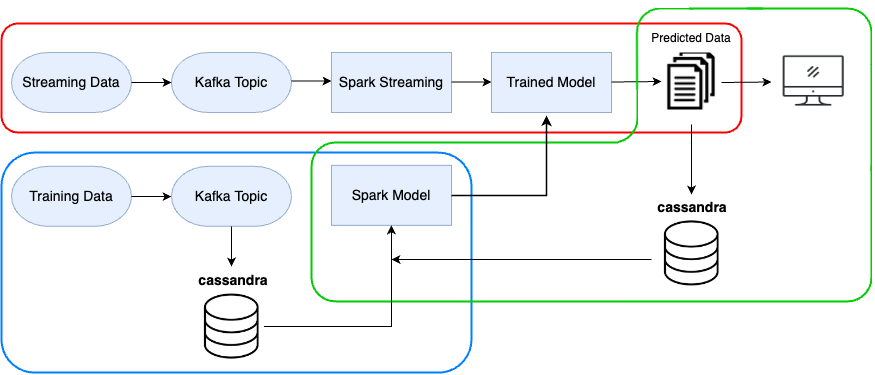
* **Thu thập dữ liệu**: Sử dụng YouTube Data API v3 để lấy bình luận thời gian thực và dữ liệu lịch sử, tạo ra luồng dữ liệu lớn cần xử lý.
* **Xử lý dữ liệu**: Apache Kafka được sử dụng để truyền dữ liệu thời gian thực, trong khi Apache Spark (với Spark Streaming) thực hiện tiền xử lý và phân tích cảm xúc bằng các thuật toán học máy.
* **Lưu trữ dữ liệu**: Cassandra đảm bảo lưu trữ hiệu quả dữ liệu thô và kết quả phân tích, hỗ trợ truy xuất nhanh cho khối lượng dữ liệu lớn.
* **Trực quan hóa**: Apache Superset cung cấp các công cụ trực quan hóa để trình bày kết quả phân tích dưới dạng biểu đồ, giúp người dùng dễ dàng nhận diện xu hướng cảm xúc.

Việc ứng dụng Big Data trong đề tài không chỉ giúp xử lý khối lượng lớn bình luận mà còn đảm bảo tính chính xác và kịp thời trong phân tích cảm xúc. Các công nghệ Big Data như Kafka, Spark và Cassandra tạo thành một pipeline hiệu quả, từ thu thập dữ liệu, xử lý, phân tích đến lưu trữ và trình bày kết quả, đáp ứng yêu cầu của một hệ thống phân tích dữ liệu thời gian thực.

## 2. Tổng quan về hệ thống

Hệ thống phân tích cảm xúc bình luận của video trên nền tảng YouTube được thiết kế để xử lý dữ liệu lớn theo thời gian thực, sử dụng các công nghệ Big Data như Apache Kafka, Apache Spark và Cassandra, kết hợp với PowerBI để trực quan hóa kết quả. Hệ thống bao gồm hai luồng xử lý chính: huấn luyện mô hình học máy dựa trên dữ liệu lịch sử và phân tích cảm xúc bình luận thời gian thực. Quy trình tổng thể được mô tả như sau:

* **Luồng huấn luyện mô hình**: Dữ liệu huấn luyện (Training Data) → Kafka Topic (youtube-training-data) → Spark Model → Cassandra (lưu mô hình).
* **Luồng phân tích thời gian thực**: Dữ liệu thời gian thực (Streaming Data) → Kafka Topic (youtube-comments-stream) → Spark Streaming → Mô hình đã huấn luyện → Dữ liệu dự đoán (Predicted Data) → Cassandra (lưu kết quả) → Trực quan hóa (PowerBI).



**Thành phần chính:**

**Nguồn dữ liệu (Streaming Data & Training Data)**:

* **Dữ liệu thời gian thực (Streaming Data)**: Đây là các bình luận mới được thu thập từ các video YouTube liên quan đến các chủ đề cụ thể như công nghệ, âm nhạc hoặc tin tức. Dữ liệu này được lấy thông qua YouTube Data API v3, đảm bảo cập nhật liên tục và phản ánh phản hồi của người xem trong thời gian thực.
* **Dữ liệu huấn luyện (Training Data)**: Bộ dữ liệu lịch sử bao gồm các bình luận đã được thu thập và lưu trữ trong cơ sở dữ liệu Cassandra. Dữ liệu này được sử dụng để huấn luyện các mô hình học máy nhằm phân loại cảm xúc (tích cực, tiêu cực, trung lập).

**Apache Kafka**:

Apache Kafka đóng vai trò là nền tảng trung gian để truyền dữ liệu hiệu quả giữa các thành phần của hệ thống. Hệ thống sử dụng hai Kafka Topic nguồn data và các topic khác:

* **youtube-comments-stream**: Nhận luồng bình luận mới từ YouTube Data API, cung cấp dữ liệu thời gian thực cho Spark Streaming để phân tích cảm xúc.
* **youtube-training-data**: Nhận dữ liệu huấn luyện từ Cassandra, cung cấp dữ liệu lịch sử để xây dựng và huấn luyện mô hình học máy.

Kafka đảm bảo khả năng xử lý luồng dữ liệu lớn với độ trễ thấp, hỗ trợ hệ thống hoạt động liên tục và đáng tin cậy.

**Apache Spark**:

Apache Spark được sử dụng để xử lý và phân tích dữ liệu, với hai thành phần chính:

* **Spark Streaming**: Xử lý luồng bình luận thời gian thực từ Kafka Topic youtube-comments-stream. Spark Streaming áp dụng mô hình học máy đã được huấn luyện để phân tích cảm xúc của các bình luận, đảm bảo xử lý nhanh chóng và hiệu quả.
* **Spark Model**: Huấn luyện mô hình học máy dựa trên dữ liệu lịch sử từ Kafka Topic youtube-training-data. Quy trình bao gồm tiền xử lý dữ liệu (làm sạch văn bản, dịch ngôn ngữ, chuẩn hóa) và áp dụng các thuật toán học máy như phân tích cảm xúc bằng TextBlob hoặc các mô hình NLP khác.

**Cassandra**:

Apache Cassandra là cơ sở dữ liệu phân tán được sử dụng để lưu trữ dữ liệu thô (bình luận, thông tin video) và dữ liệu đã phân tích (kết quả cảm xúc). Cassandra được chọn vì khả năng mở rộng, hiệu suất cao và khả năng xử lý khối lượng dữ liệu lớn, phù hợp với yêu cầu của hệ thống Big Data. Dữ liệu được lưu trữ trong các bảng như youtube\_sentiment.comments, cho phép truy xuất nhanh và dễ dàng tích hợp với các công cụ khác.

PowerBI:

PowerBI được sử dụng để trực quan hóa kết quả phân tích cảm xúc, thay thế cho Apache Superset. PowerBI cho phép tạo các biểu đồ, bảng biểu và dashboard trực quan, giúp người dùng dễ dàng nhận diện xu hướng cảm xúc của bình luận theo thời gian, chủ đề hoặc khu vực địa lý. Dữ liệu từ Cassandra được truy xuất và nhập vào PowerBI để tạo ra các báo cáo chi tiết, hỗ trợ các nhà sản xuất nội dung và doanh nghiệp đưa ra quyết định dựa trên dữ liệu.

Hệ thống được thiết kế để đảm bảo tính liên tục, khả năng mở rộng và hiệu quả trong việc xử lý dữ liệu lớn, từ thu thập bình luận, phân tích cảm xúc đến lưu trữ và trực quan hóa kết quả. Các thành phần như Kafka, Spark, Cassandra và PowerBI phối hợp chặt chẽ để tạo thành một pipeline hoàn chỉnh, đáp ứng yêu cầu của bài toán phân tích cảm xúc trên nền tảng YouTube.

# **CHƯƠNG III: TRIỂN KHAI BÀI TOÁN**

## 1. Đăng ký Youtube API

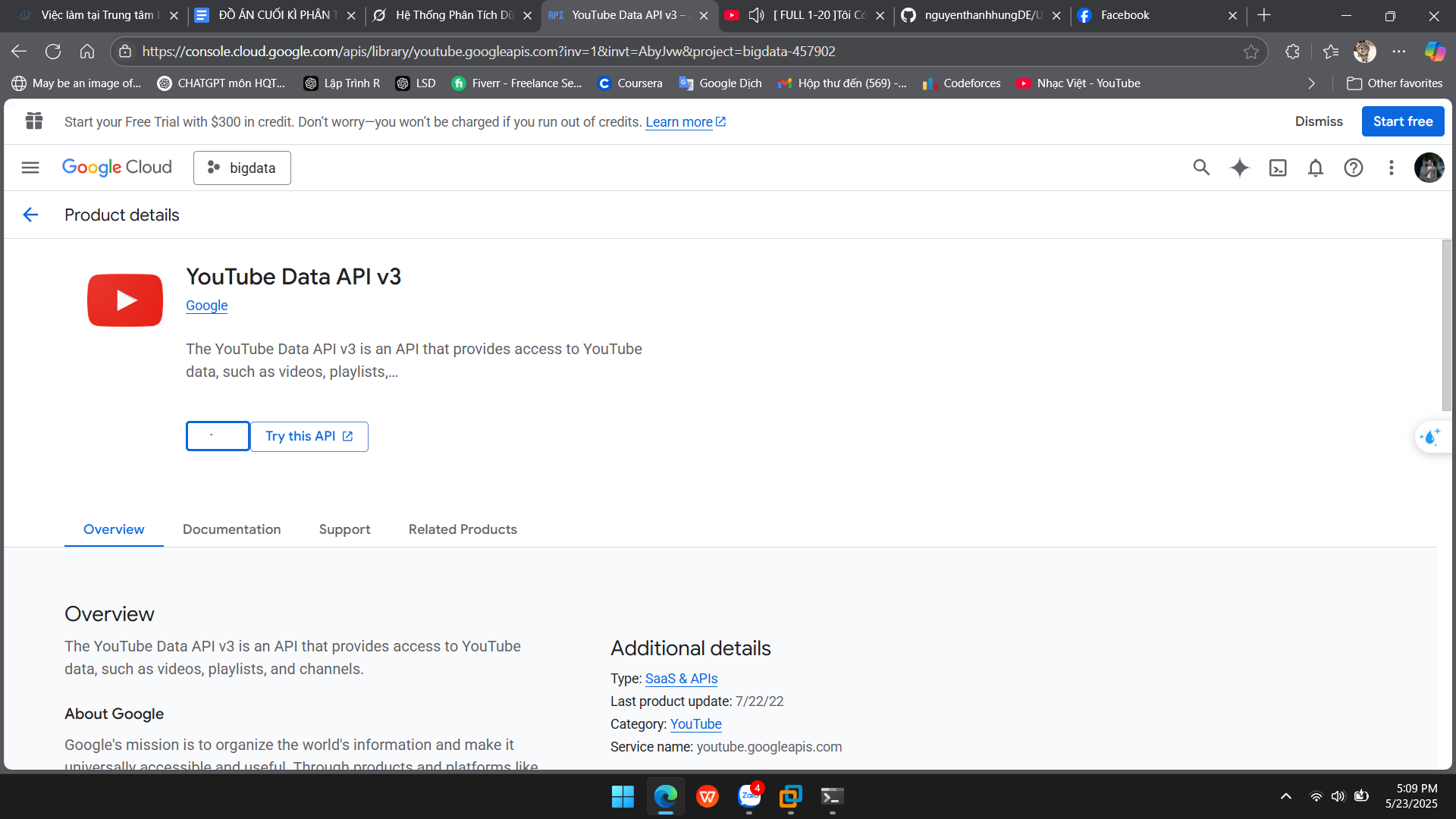
Để thu thập dữ liệu bình luận từ các video YouTube, hệ thống sử dụng **YouTube Data API v3**, một công cụ mạnh mẽ do Google cung cấp, cho phép truy xuất thông tin như bình luận, thông tin video và các dữ liệu liên quan khác. Việc đăng ký và kích hoạt API là bước đầu tiên và quan trọng trong quy trình xây dựng hệ thống phân tích cảm xúc bình luận. Dưới đây là các bước chi tiết để đăng ký và kiểm tra YouTube API:

1.1. Tạo và kích hoạt dự án trên Google Cloud Console

**B1: Truy cập Google Cloud Console**: Mở trình duyệt và truy cập vào Google Cloud Console. Đăng nhập bằng tài khoản Google để bắt đầu.

**B2: Tạo dự án mới**:

* Tại giao diện chính, chọn **"Select a project"** ở góc trên bên trái, sau đó nhấn **"New Project"**.
* Đặt tên dự án, ví dụ: **"YouTube Sentiment Analysis"**, và chọn tổ chức (nếu có).
* Nhấn **"Create"** để hoàn tất việc tạo dự án.



**B3: Kích hoạt YouTube Data API v3**:

* + Trong menu điều hướng bên trái, chọn **APIs & Services** → **Library**.
  + Tìm **YouTube Data API v3** trong danh sách API bằng thanh tìm kiếm.
  + Nhấn vào **YouTube Data API v3** và chọn **Enable** để kích hoạt API cho dự án.



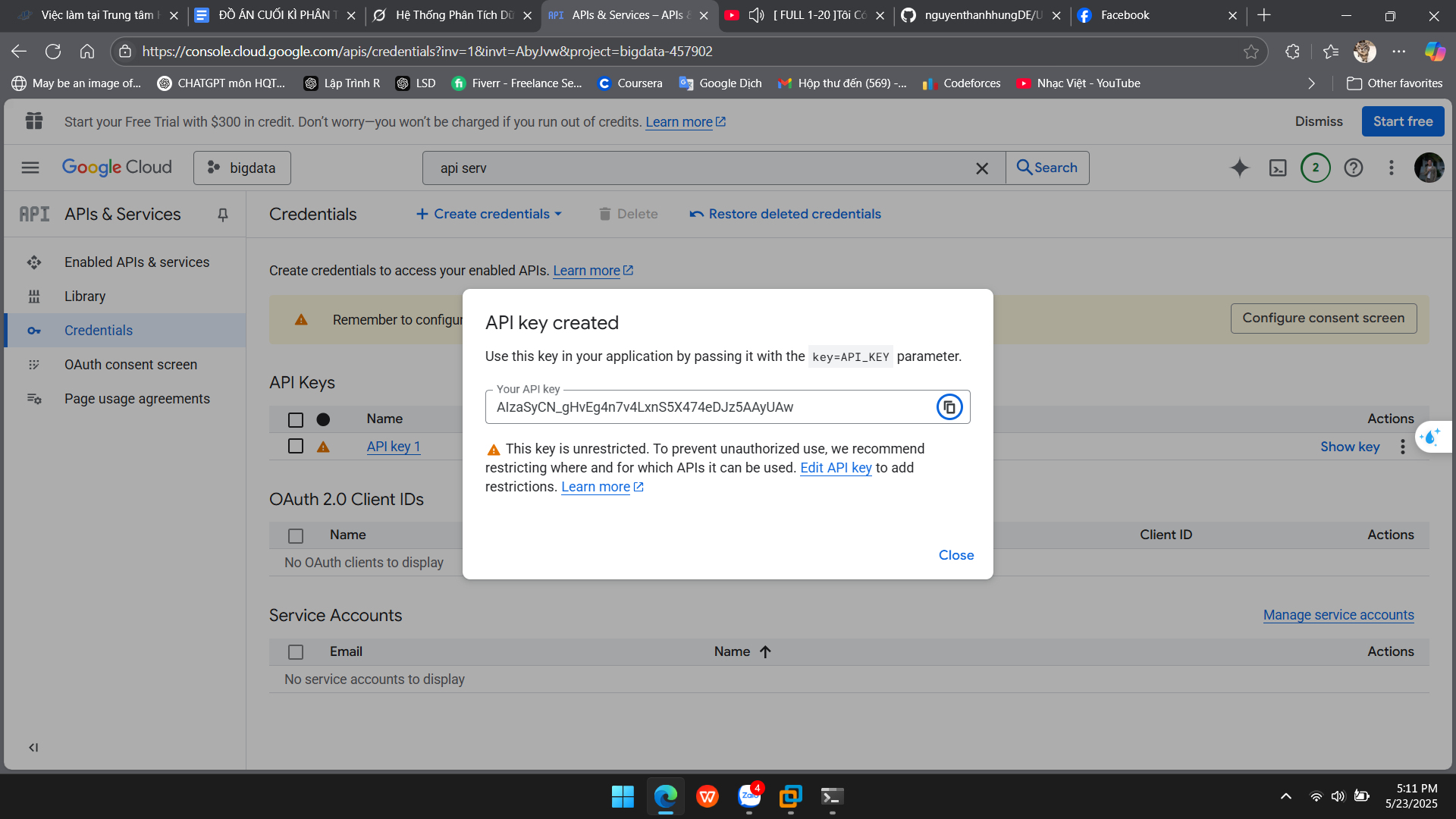
1.2. Tạo API Key

**B1: Truy cập mục Credentials**:

* Trong menu **APIs & Services**, chọn **Credentials**.
* Nhấn **Create Credentials** và chọn **API Key** từ danh sách thả xuống.

**B2: Lưu API Key**:

* Sau khi tạo, hệ thống sẽ cung cấp một chuỗi API Key (ví dụ: AIzaSyCN\_gHvEg4n7v4LxnS5X474eDJz5AAyUAw).
* Lưu trữ API Key này ở nơi an toàn, vì nó sẽ được sử dụng trong các script Python để gọi API.



**B3: (Tùy chọn) Tạo OAuth 2.0 Client ID**:

* Nếu hệ thống cần truy cập vào các tài nguyên yêu cầu quyền nâng cao (ví dụ: bình luận của người dùng cụ thể), vào **Create Credentials** → **OAuth 2.0 Client ID**.
* Cấu hình màn hình đồng ý (Consent Screen) và lưu thông tin Client ID nếu cần.

1.3. Kiểm tra API Key

Để đảm bảo API Key hoạt động đúng, một script Python đơn giản có thể được sử dụng để kiểm tra khả năng truy xuất dữ liệu từ YouTube. Dưới đây là đoạn mã mẫu để kiểm tra:

| import googleapiclient.discovery  # Cấu hình API Key  api\_key = "AIzaSyCN\_gHvEg4n7v4LxnS5X474eDJz5AAyUAw"  # Khởi tạo client YouTube API  youtube = googleapiclient.discovery.build("youtube", "v3", developerKey=api\_key)  # Gửi yêu cầu lấy 10 bình luận từ một video cụ thể  request = youtube.commentThreads().list(  part="snippet",  videoId="VIDEO\_ID", # Thay VIDEO\_ID bằng ID của video cần truy xuất  maxResults=10  )  # Thực thi yêu cầu và in kết quả  response = request.execute()  print(response) |
| --- |

Giải thích đoạn mã:

* **Thư viện**: Sử dụng googleapiclient.discovery để tương tác với YouTube Data API.
* **Cấu hình**: API Key được sử dụng để xác thực yêu cầu.
* **Yêu cầu**: Lấy thông tin từ phần snippet của bình luận, giới hạn 10 bình luận từ video có VIDEO\_ID được chỉ định.
* **Kết quả**: In ra dữ liệu trả về từ API để kiểm tra tính hợp lệ của API Key và cấu hình.

1.4. Lưu ý khi sử dụng YouTube API

* **Giới hạn quota**: YouTube Data API v3 có giới hạn quota hàng ngày (mặc định là 10.000 đơn vị). Mỗi yêu cầu API tiêu tốn một lượng quota nhất định, vì vậy cần tối ưu hóa số lượng yêu cầu để tránh vượt quá giới hạn.
* **Bảo mật API Key**: Không chia sẻ API Key công khai, đặc biệt trong mã nguồn được lưu trên các nền tảng công cộng như GitHub.
* **Xử lý lỗi**: Kiểm tra và xử lý các lỗi trả về từ API (ví dụ: lỗi quota, lỗi xác thực) để đảm bảo hệ thống hoạt động ổn định.

Việc đăng ký và kiểm tra YouTube API là bước nền tảng để thu thập dữ liệu bình luận, tạo tiền đề cho các bước tiếp theo như truyền dữ liệu qua Kafka, xử lý bằng Spark và lưu trữ trong Cassandra.

## 2. Thực hiện lấy dữ liệu Với Apache Kafka

**Apache Kafka** là một nền tảng truyền dữ liệu theo thời gian thực, đóng vai trò trung tâm trong pipeline Big Data để truyền dữ liệu từ YouTube API đến các hệ thống xử lý như Spark. **Quy trình**:

* **Producer**: Một script Python sử dụng YouTube API để lấy dữ liệu và đẩy vào Kafka topic.
* **Consumer**: Spark Streaming hoặc một ứng dụng khác đọc dữ liệu từ Kafka topic để xử lý.

**Cài đặt Kafka trên Ubuntu Server**

cài đặt:

| cd /home/hadoophoangphat  wget https://downloads.apache.org/kafka/3.7.2/kafka\_2.13-3.7.2.tgz  tar -xzf kafka\_2.13-3.7.2.tgz  mv kafka\_2.13-3.7.2 kafka |
| --- |

loại bỏ file xung đột:

| mv /home/hadoophoangphat/hive/lib/kafka-clients-2.5.0.jar /home/hadoophoangphat/hive/lib/kafka-clients-2.5.0.jar.bak  mv /home/hadoophoangphat/hive/lib/zookeeper-3.8.3.jar /home/hadoophoangphat/hive/lib/zookeeper-3.8.3.jar.bak  mv /home/hadoophoangphat/hive/lib/zookeeper-jute-3.8.3.jar /home/hadoophoangphat/hive/lib/zookeeper-jute-3.8.3.jar.bak |
| --- |

chỉnh sửa: /home/hadoophoangphat/kafka/config/server.properties

| broker.id=1  listeners=PLAINTEXT://0.0.0.0:9092  advertised.listeners=PLAINTEXT://192.168.x.x:9092  zookeeper.connect=192.168.x.x:2181 |
| --- |

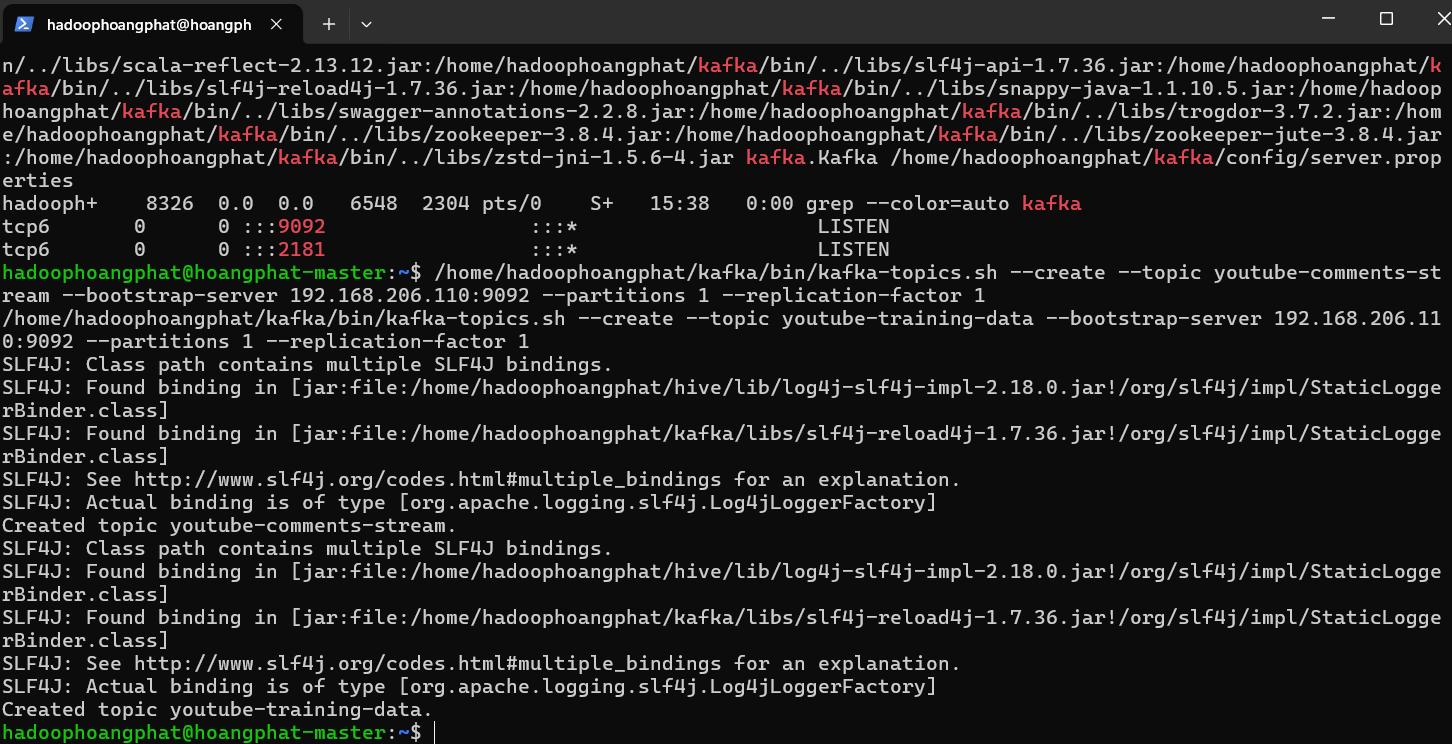
cấu hình zookeeper:

chỉnh sửa trong: /home/hadoophoangphat/kafka/config/zookeeper.properties

| dataDir=/home/hadoophoangphat/zookeeper/data  clientPort=2181  maxClientCnxns=0  /home/hadoophoangphat/kafka/bin/zookeeper-server-start.sh -daemon /home/hadoophoangphat/kafka/config/zookeeper.properties  /home/hadoophoangphat/kafka/bin/kafka-server-start.sh -daemon /home/hadoophoangphat/kafka/config/server.properties |
| --- |

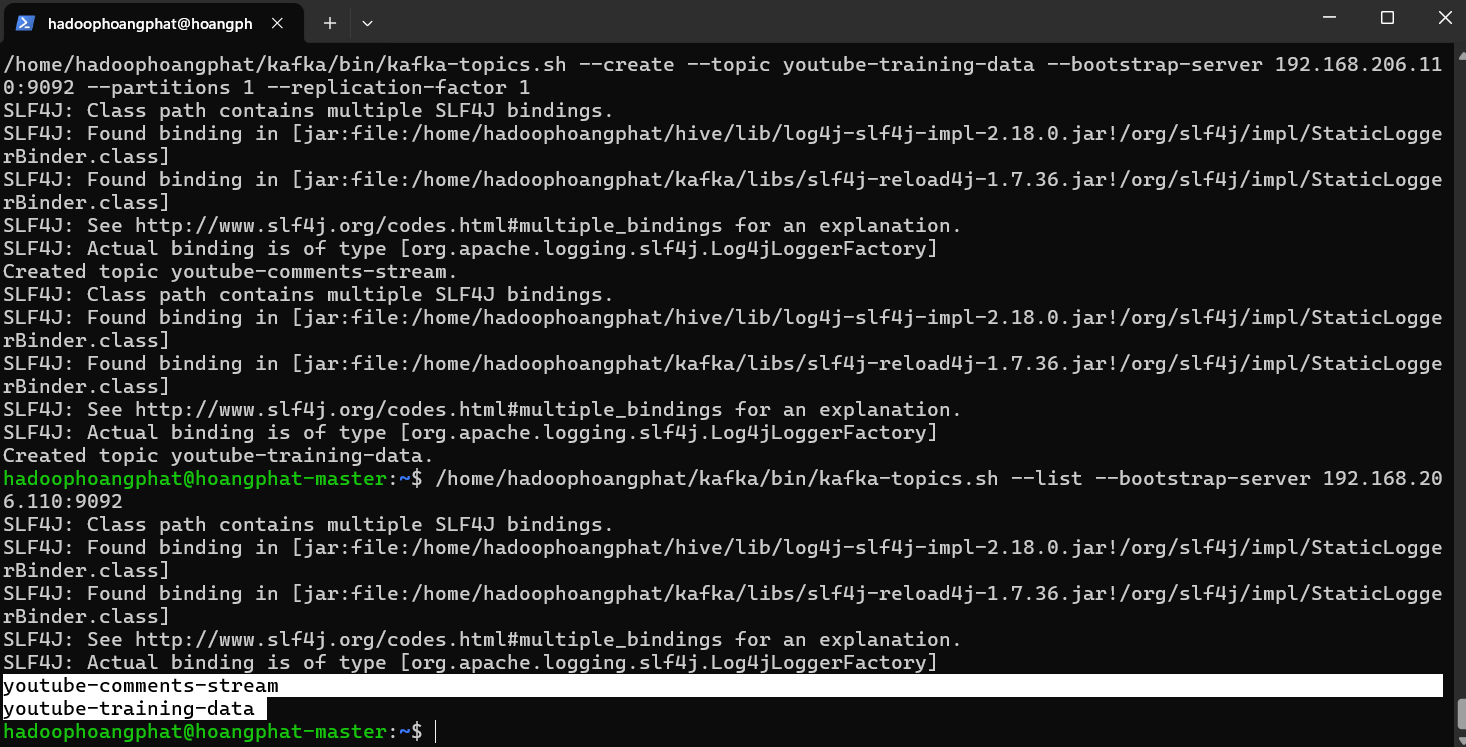
**Tạo Kafka topic**

| /home/hadoophoangphat/kafka/bin/kafka-topics.sh --create --topic youtube-comments-stream --bootstrap-server 192.168.206.110:9092 --partitions 1 --replication-factor 1  /home/hadoophoangphat/kafka/bin/kafka-topics.sh --create --topic youtube-training-data --bootstrap-server 192.168.206.110:9092 --partitions 1 --replication-factor 1 |
| --- |



kiểm tra topic:

| /home/hadoophoangphat/kafka/bin/kafka-topics.sh --list --bootstrap-server 192.168.206.110:9092 |
| --- |



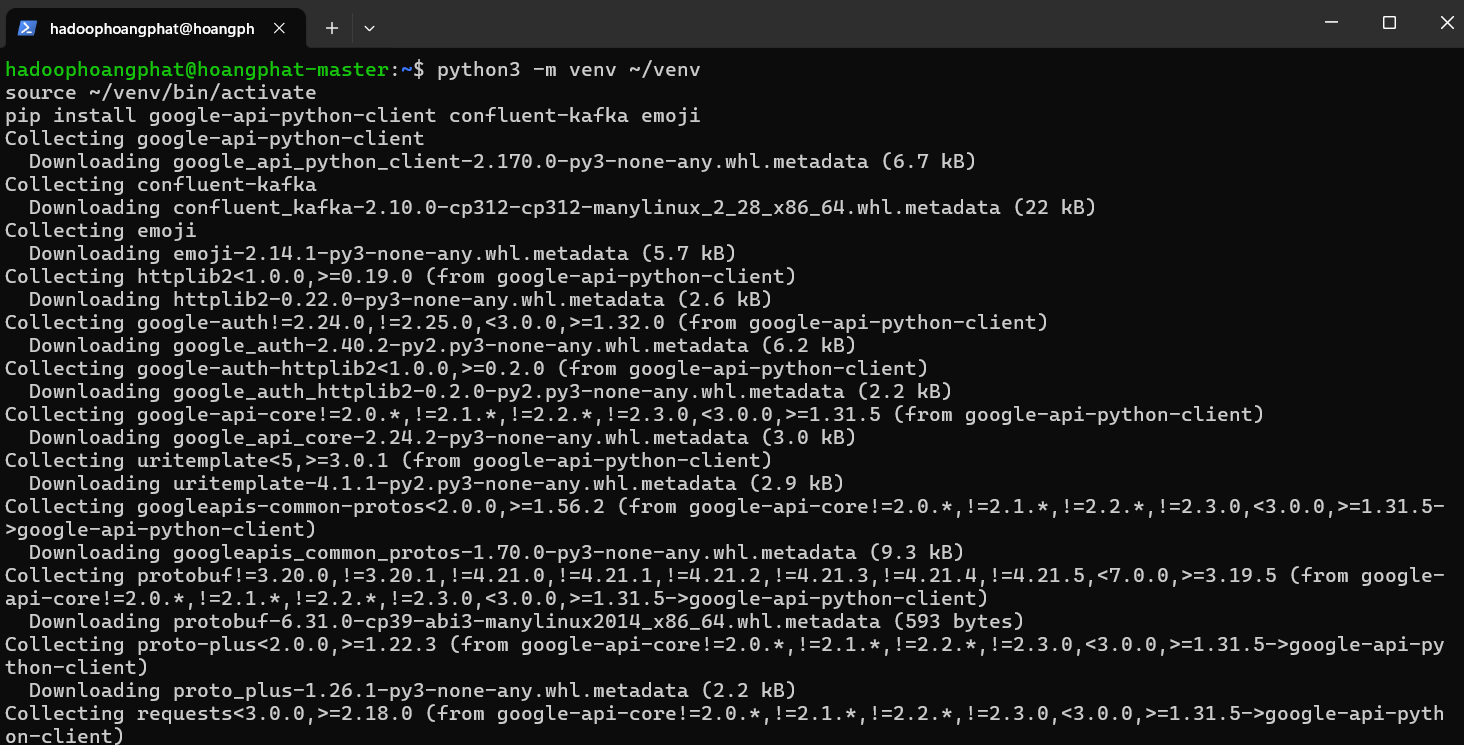
**Producer đẩy dữ liệu YouTube vào Kafka**:

tạo môi trường ảo:

| python3 -m venv ~/venv  source ~/venv/bin/activate |
| --- |

mở lại sau khi tắt:

| cd ~/venv  source bin/activate  pip install google-api-python-client confluent-kafka emoji |
| --- |



tạo file producer.py

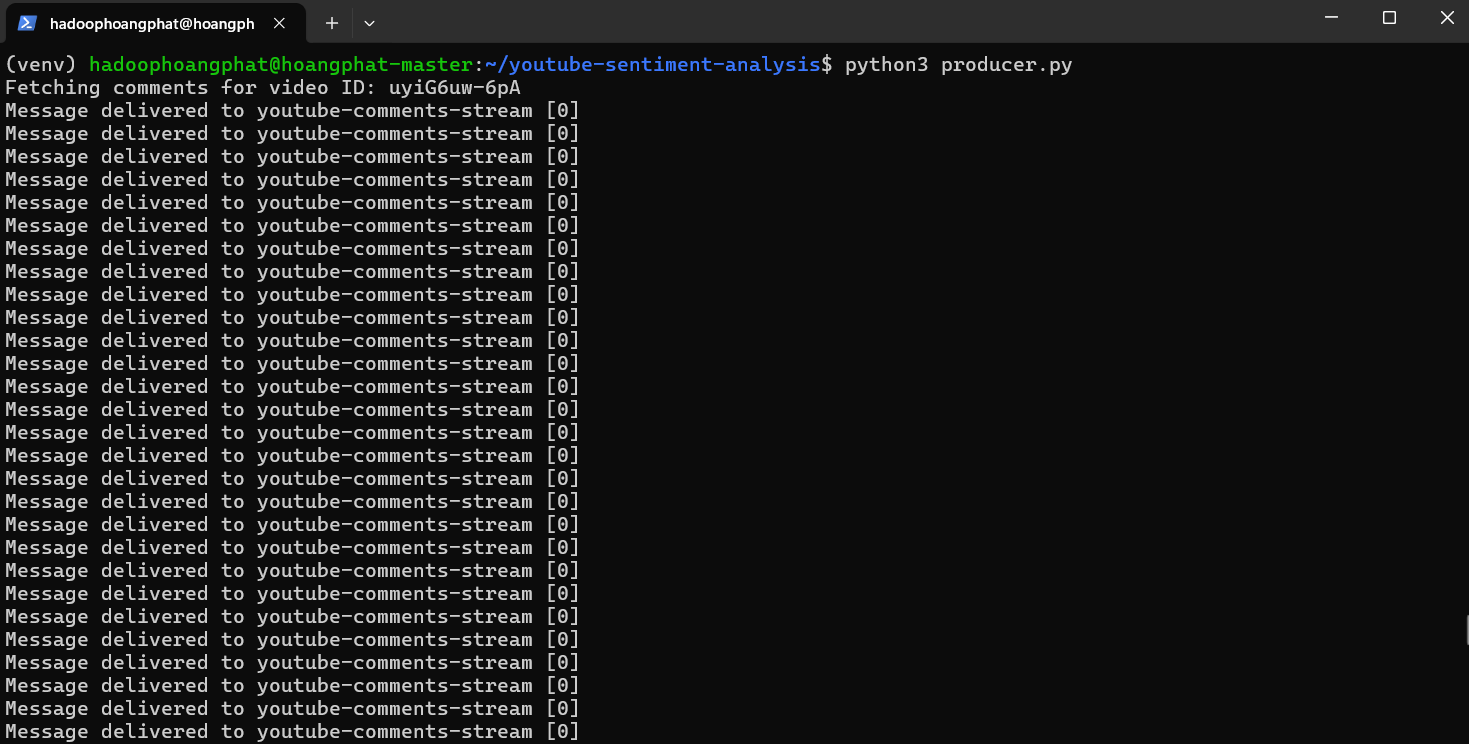
| mkdir -p /home/hadoophoangphat/youtube-sentiment-analysis  cd /home/hadoophoangphat/youtube-sentiment-analysis  touch producer.py  chown hadoophoangphat:hadoophoangphat /home/hadoophoangphat/youtube-sentiment-analysis  chmod u+rwx /home/hadoophoangphat/youtube-sentiment-analysis |
| --- |

producer.py

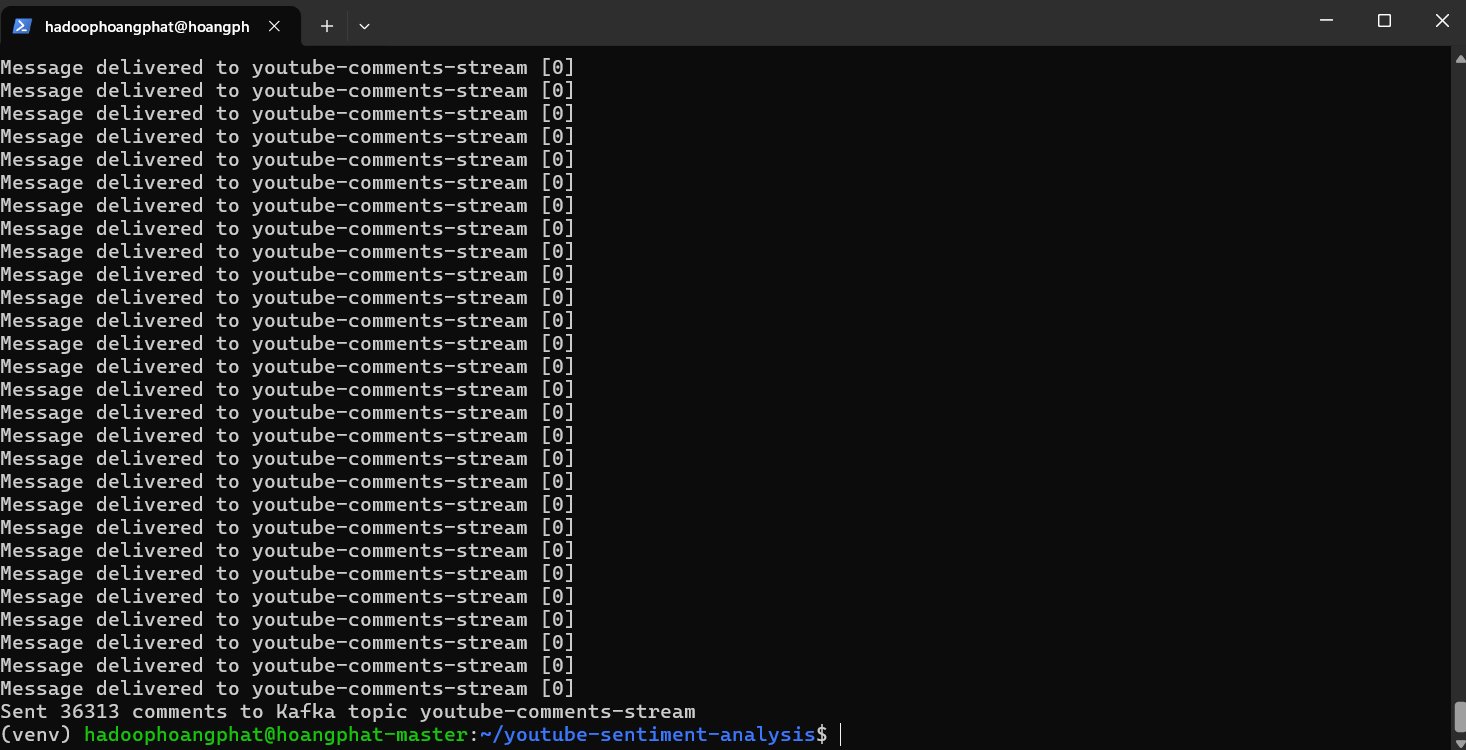
| import os  from googleapiclient.discovery import build  from confluent\_kafka import Producer  import json  import time  import emoji  # Cấu hình YouTube API  API\_KEY = "AIzaSyCN\_gHvEg4n7v4LxnS5X474eDJz5AAyUAw"  YOUTUBE\_API\_SERVICE\_NAME = "youtube"  YOUTUBE\_API\_VERSION = "v3"  VIDEO\_ID = "uyiG6uw-6pA"  # Cấu hình Kafka Producer  KAFKA\_BOOTSTRAP\_SERVERS = "192.168.206.110:9092"  KAFKA\_TOPIC = "youtube-comments-stream"  def delivery\_report(err, msg):  """Báo cáo trạng thái gửi message tới Kafka."""  if err is not None:  print(f"Message delivery failed: {err}")  else:  print(f"Message delivered to {msg.topic()} [{msg.partition()}]")  def get\_youtube\_comments(youtube, video\_id):  """Lấy bình luận từ YouTube API."""  comments = []  request = youtube.commentThreads().list(  part="snippet",  videoId=video\_id,  maxResults=100, # Số lượng bình luận tối đa mỗi request  textFormat="plainText"  )    while request:  response = request.execute()  for item in response["items"]:  comment = item["snippet"]["topLevelComment"]["snippet"]["textDisplay"]  # Loại bỏ emoji để tránh lỗi encoding  comment = emoji.replace\_emoji(comment, replace="")  comments.append(comment)    # Chuyển sang trang tiếp theo nếu có  request = youtube.commentThreads().list\_next(request, response)    return comments  def main():  # Khởi tạo YouTube API client  youtube = build(  YOUTUBE\_API\_SERVICE\_NAME,  YOUTUBE\_API\_VERSION,  developerKey=API\_KEY  )    # Khởi tạo Kafka Producer  producer\_conf = {  "bootstrap.servers": KAFKA\_BOOTSTRAP\_SERVERS,  "client.id": "youtube-comment-producer"  }  producer = Producer(producer\_conf)    # Lấy bình luận từ YouTube  print(f"Fetching comments for video ID: {VIDEO\_ID}")  comments = get\_youtube\_comments(youtube, VIDEO\_ID)    # Gửi bình luận tới Kafka  for comment in comments:  try:  producer.produce(  KAFKA\_TOPIC,  value=json.dumps({"comment": comment}).encode("utf-8"),  callback=delivery\_report  )  producer.poll(0) # Kích hoạt callback  time.sleep(0.1) # Tránh gửi quá nhanh  except Exception as e:  print(f"Error producing message: {e}")    # Đợi tất cả message được gửi  producer.flush()  print(f"Sent {len(comments)} comments to Kafka topic {KAFKA\_TOPIC}")  if \_\_name\_\_ == "\_\_main\_\_":  main() |
| --- |

chạy file:

| python3 producer.py |
| --- |

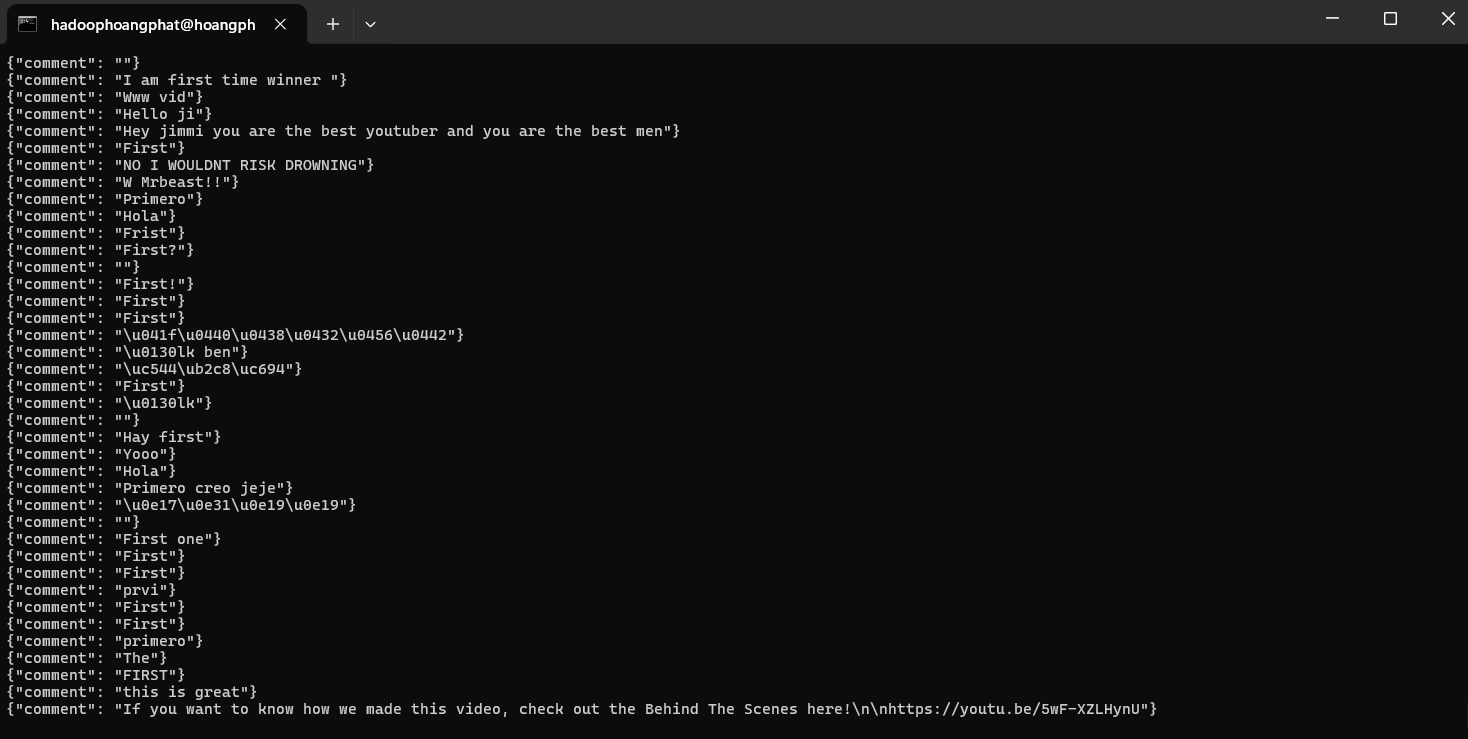


Kết quả được lưu vào kafka topic:



Kiểm tra kết quả:

| /home/hadoophoangphat/kafka/bin/kafka-console-consumer.sh --bootstrap-server 192.168.206.110:9092 --topic youtube-comments-stream --from-beginning |
| --- |



## 3. Tiền xử lý dữ liệu

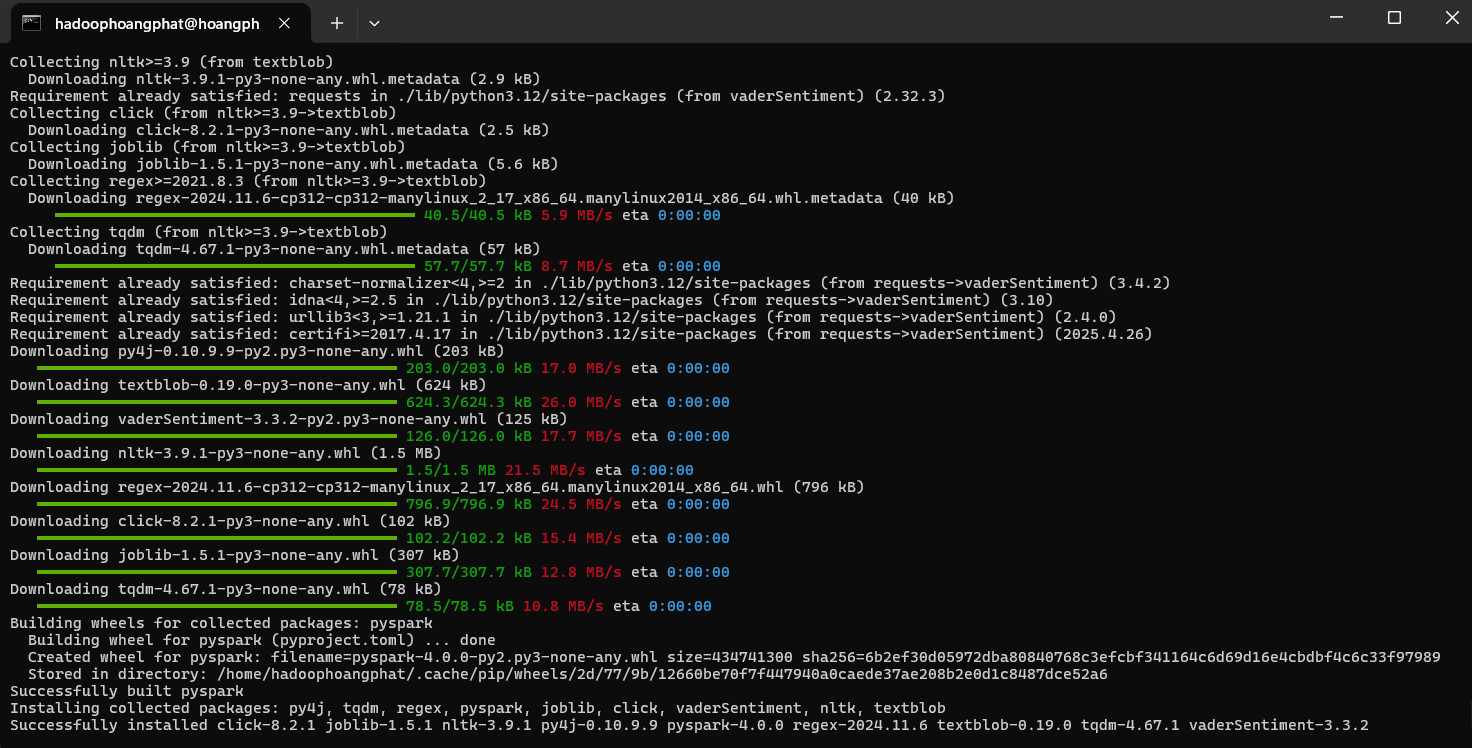
Apache Spark (cụ thể là Spark Streaming hoặc Structured Streaming) được sử dụng để xử lý dữ liệu từ Kafka theo thời gian thực hoặc theo batch.

Spark Structured Streaming đọc dữ liệu từ Kafka

thực hiện tiền xử lý và chạy model

Cài đặt thư viện Python cho Spark:

pip3 install pyspark textblob vaderSentiment



**Tiền xử lý:**

| cd /home/hadoophoangphat/youtube-sentiment-analysis  touch preprocess.py |
| --- |

preprocess.py:

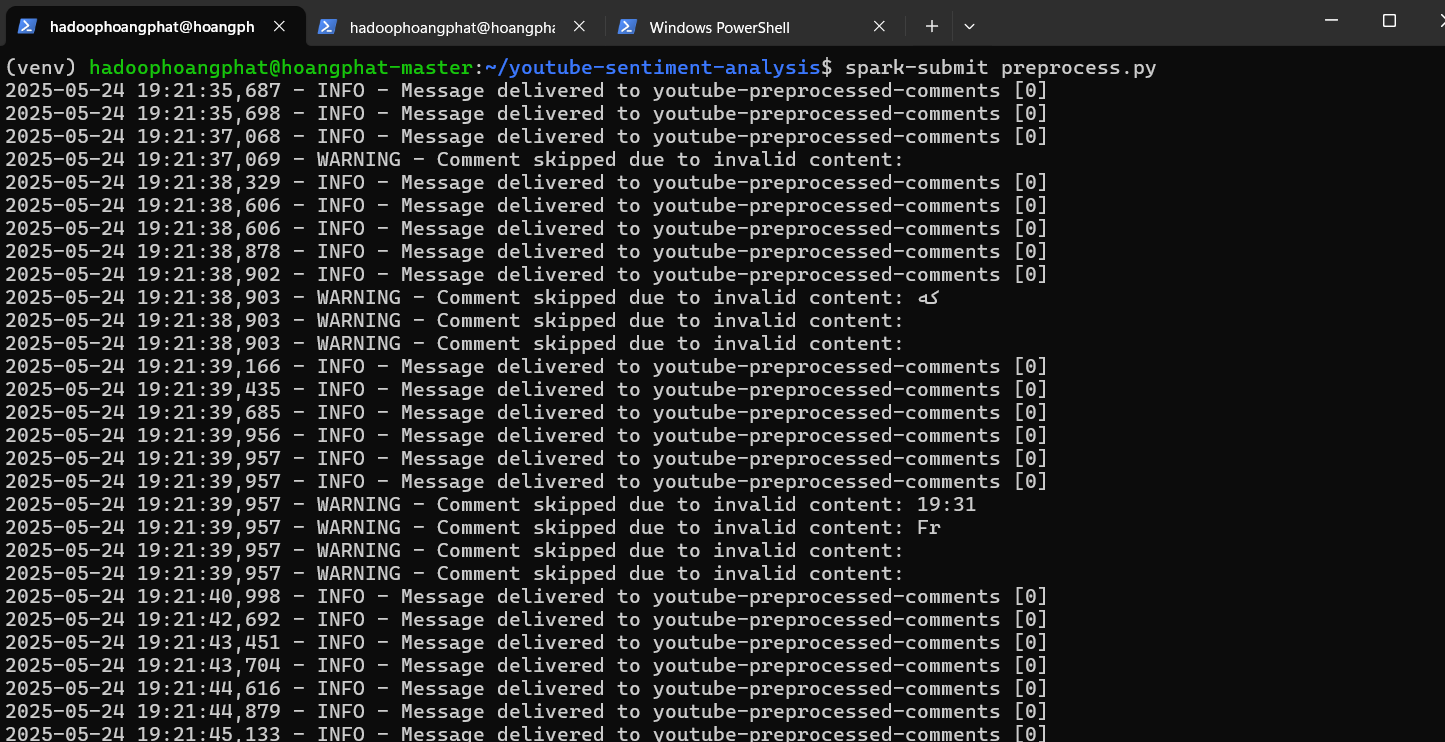
| import json  import re  import emoji  from confluent\_kafka import Consumer, Producer, KafkaError  from langdetect import detect, LangDetectException  from googletrans import Translator  import logging  import time  from functools import lru\_cache  # Thiết lập logging  logging.basicConfig(level=logging.INFO, format='%(asctime)s - %(levelname)s - %(message)s')  # Cấu hình Kafka  KAFKA\_BOOTSTRAP\_SERVERS = "192.168.206.110:9092"  INPUT\_TOPIC = "youtube-comments-stream"  OUTPUT\_TOPIC = "youtube-preprocessed-comments"  # Khởi tạo Translator  translator = Translator()  # Bộ nhớ đệm cho dịch thuật  @lru\_cache(maxsize=1000)  def cached\_translate(text, dest="en"):  """Dịch văn bản với bộ nhớ đệm để tối ưu hóa."""  try:  return translator.translate(text, dest=dest).text  except Exception as e:  logging.error(f"Translation error for text '{text}': {e}")  return text  def delivery\_report(err, msg):  """Báo cáo trạng thái gửi message tới Kafka."""  if err is not None:  logging.error(f"Message delivery failed: {err}")  else:  logging.info(f"Message delivered to {msg.topic()} [{msg.partition()}]")  def clean\_text(text):  """Làm sạch văn bản."""  if not isinstance(text, str) or len(text.strip()) == 0:  return None    # Chuyển đổi emoji thành văn bản  text = emoji.demojize(text)    # Loại bỏ URL  text = re.sub(r'http\S+|www\S+|https\S+', '', text, flags=re.MULTILINE)    # Loại bỏ ký tự đặc biệt, chỉ giữ chữ và số  text = re.sub(r'[^\w\s]', '', text)    # Chuyển về chữ thường và loại bỏ khoảng trắng thừa  text = text.lower().strip()  text = re.sub(r'\s+', ' ', text)    # Loại bỏ các bình luận quá ngắn hoặc chỉ chứa số  if len(text) < 3 or text.isdigit():  return None    return text  def detect\_language(text):  """Phát hiện ngôn ngữ của văn bản."""  try:  if not isinstance(text, str) or len(text.strip()) < 3:  return 'unknown'  return detect(text)  except LangDetectException:  logging.warning(f"Could not detect language for text: {text}")  return 'unknown'  def map\_language\_to\_country(lang, comment):  """Ánh xạ ngôn ngữ sang quốc gia."""  language\_to\_country = {  'es': 'Spain',  'en': 'United States',  'bn': 'Bangladesh',  'tr': 'Turkey',  'ar': 'Saudi Arabia',  'ru': 'Russia',  'unknown': 'Unknown'  }  # Kiểm tra nội dung bình luận để tìm thông tin quốc gia  if isinstance(comment, str):  comment = comment.lower()  if 'colombia' in comment:  return 'Colombia'  elif 'punjab' in comment:  return 'India'  return language\_to\_country.get(lang, 'Unknown')  def preprocess\_comment(comment\_data):  """Tiền xử lý một bình luận."""  try:  comment = comment\_data.get("comment", "")  sentiment = comment\_data.get("sentiment", "unknown")    # Làm sạch văn bản  cleaned = clean\_text(comment)  if not cleaned:  logging.warning(f"Comment skipped due to invalid content: {comment}")  return None    # Phát hiện ngôn ngữ  language = detect\_language(comment)    # Ánh xạ quốc gia  country = map\_language\_to\_country(language, comment)    # Dịch sang tiếng Anh nếu không phải tiếng Anh  translated = cleaned  if language != "en" and language != "unknown":  translated = cached\_translate(cleaned)  if not translated or len(translated) < 3:  logging.warning(f"Translated comment too short or invalid: {translated}")  return None    # Tạo dictionary kết quả  result = {  "original\_comment": comment,  "cleaned\_comment": cleaned,  "translated\_comment": translated,  "language": language,  "country": country,  "sentiment": sentiment  }    return result    except Exception as e:  logging.error(f"Error processing comment '{comment}': {e}")  return None  def main():  # Cấu hình Kafka Consumer  consumer\_conf = {  "bootstrap.servers": KAFKA\_BOOTSTRAP\_SERVERS,  "group.id": "comment-preprocessor-group",  "auto.offset.reset": "earliest"  }  consumer = Consumer(consumer\_conf)  consumer.subscribe([INPUT\_TOPIC])  # Cấu hình Kafka Producer  producer\_conf = {  "bootstrap.servers": KAFKA\_BOOTSTRAP\_SERVERS,  "client.id": "comment-preprocessor"  }  producer = Producer(producer\_conf)  try:  while True:  msg = consumer.poll(timeout=1.0)  if msg is None:  continue  if msg.error():  if msg.error().code() == KafkaError.\_PARTITION\_EOF:  continue  else:  logging.error(f"Consumer error: {msg.error()}")  break  # Parse message  try:  data = json.loads(msg.value().decode("utf-8"))  processed = preprocess\_comment(data)    if processed:  # Gửi dữ liệu đã xử lý tới topic mới  producer.produce(  OUTPUT\_TOPIC,  value=json.dumps(processed).encode("utf-8"),  callback=delivery\_report  )  producer.poll(0)  except Exception as e:  logging.error(f"Error processing message: {e}")  except KeyboardInterrupt:  logging.info("Received KeyboardInterrupt, shutting down...")  finally:  logging.info("Closing consumer and producer...")  consumer.close()  producer.flush()  producer.close()  logging.info("Shutdown complete.")  if \_\_name\_\_ == "\_\_main\_\_":  main() |
| --- |

cài đặt thư viện cho file:

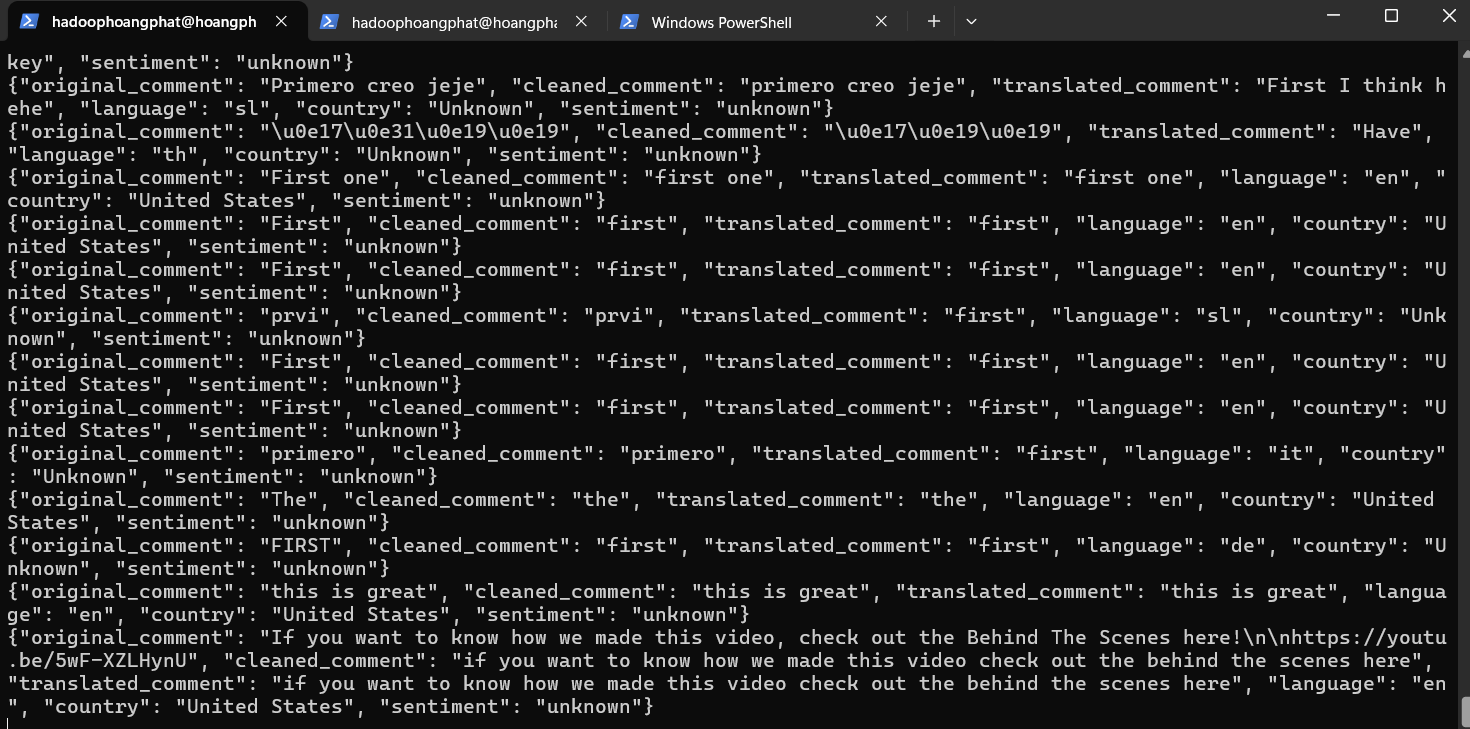
| pip install googletrans==4.0.0-rc1  pip install emoji  pip install langdetect |
| --- |

chạy file:

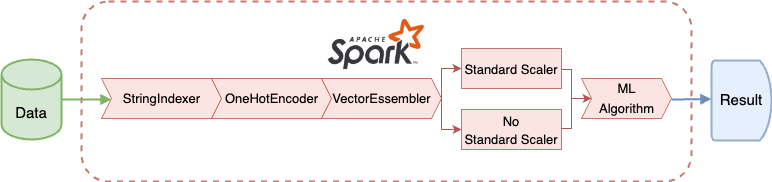
| spark-submit preprocess.py |
| --- |



Kết quả:



## 4. Phân tích cảm xúc:



**Tạo file processor.py**:

| touch /home/hadoophoangphat/youtube-sentiment-analysis/processor.py  nano /home/hadoophoangphat/youtube-sentiment-analysis/processor.py |
| --- |

**processor.py**:

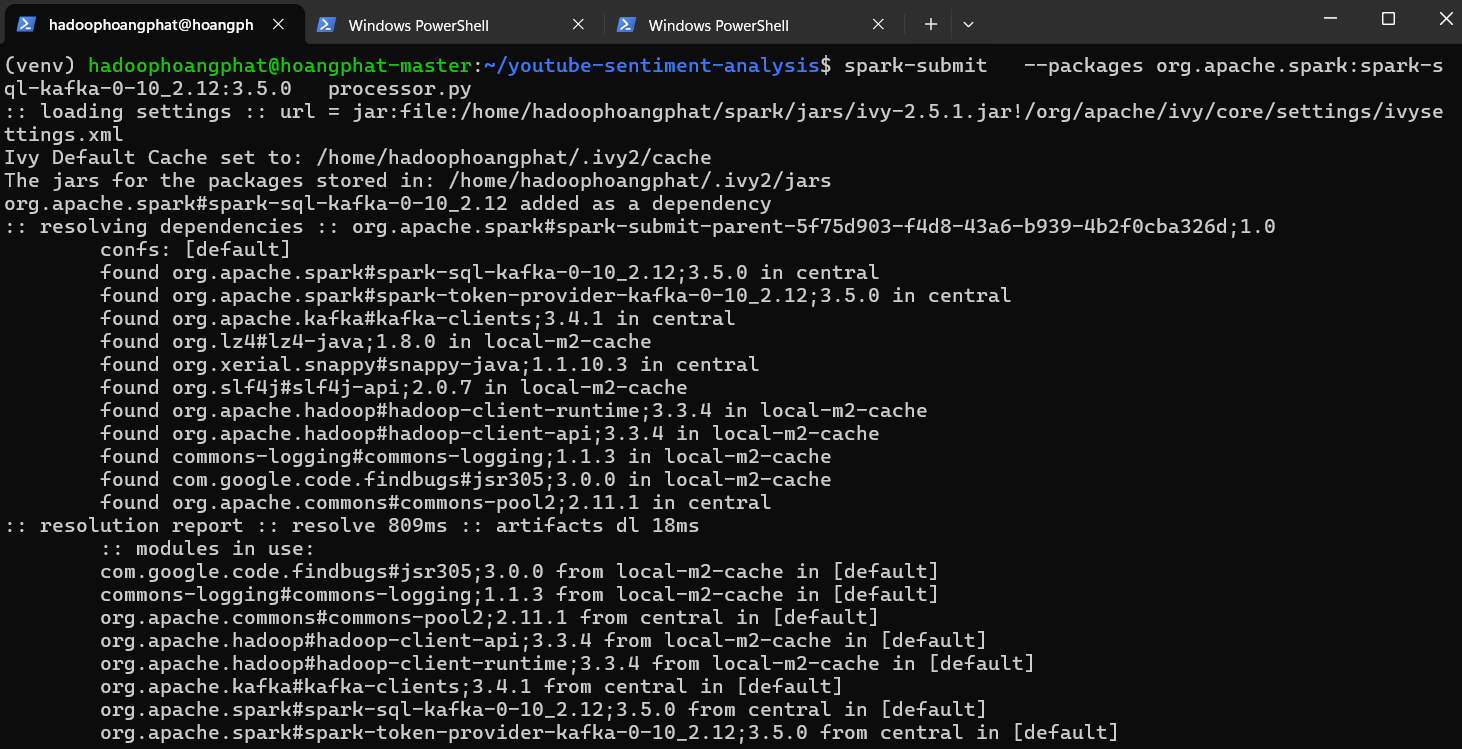
| from pyspark.sql import SparkSession  from pyspark.sql.functions import from\_json, col, udf, to\_json, struct  from pyspark.sql.types import StructType, StringType  from textblob import TextBlob  # Tạo Spark session  spark = (  SparkSession.builder  .appName("YouTubeSentimentAnalysis")  .config("spark.sql.adaptive.enabled", "false")  .config("spark.streaming.stopGracefullyOnShutdown", "true")  .config("spark.sql.streaming.schemaInference", "true") # Tự động suy ra schema nếu cần  .getOrCreate()  )  spark.sparkContext.setLogLevel("INFO")  # Định nghĩa schema  schema = (  StructType()  .add("original\_comment", StringType())  .add("cleaned\_comment", StringType())  .add("translated\_comment", StringType())  .add("language", StringType())  .add("country", StringType())  .add("sentiment", StringType())  )  # Đọc từ Kafka  df\_raw = (  spark.readStream  .format("kafka")  .option("kafka.bootstrap.servers", "192.168.206.110:9092")  .option("subscribe", "youtube-preprocessed-comments")  .option("startingOffsets", "earliest")  .option("kafka.consumer.commit.groupid", "spark-kafka-consumer")  .option("failOnDataLoss", "false")  .option("kafka.request.timeout.ms", "60000")  .option("kafka.session.timeout.ms", "30000")  .load()  )  # In dữ liệu thô  raw\_query = (  df\_raw.selectExpr("CAST(key AS STRING)", "CAST(value AS STRING) as raw\_data")  .writeStream  .format("console")  .outputMode("append")  .option("truncate", "false")  .trigger(processingTime="10 seconds") # Trigger mỗi 10 giây  .start()  )  # Parse JSON  df\_parsed = (  df\_raw.selectExpr("CAST(value AS STRING) as json")  .select(from\_json(col("json"), schema).alias("data"))  .select("data.\*")  )  # In dữ liệu parsed  parsed\_query = (  df\_parsed.writeStream  .format("console")  .outputMode("append")  .option("truncate", "false")  .trigger(processingTime="10 seconds")  .start()  )  # Hàm phân tích cảm xúc  def analyze\_sentiment(text):  if not text or len(text.strip()) < 3:  print(f"Skipping sentiment analysis for text: {text}")  return "unknown"  try:  blob = TextBlob(text)  polarity = blob.sentiment.polarity  print(f"Text: {text}, Polarity: {polarity}")  return "positive" if polarity > 0.2 else "negative" if polarity < -0.2 else "neutral"  except Exception as e:  print(f"Error analyzing sentiment for text '{text}': {e}")  return "unknown"  # Đăng ký UDF  sentiment\_udf = udf(analyze\_sentiment, StringType())  # Áp dụng sentiment  df\_with\_sentiment = df\_parsed.withColumn("sentiment", sentiment\_udf(col("translated\_comment")))  # In dữ liệu sau sentiment  sentiment\_query = (  df\_with\_sentiment.writeStream  .format("console")  .outputMode("append")  .option("truncate", "false")  .trigger(processingTime="10 seconds")  .start()  )  # Chuyển về JSON và ghi vào Kafka  df\_output = df\_with\_sentiment.select(  to\_json(  struct(  col("original\_comment"),  col("cleaned\_comment"),  col("translated\_comment"),  col("language"),  col("country"),  col("sentiment")  )  ).alias("value")  )  # Ghi vào Kafka  query = (  df\_output.writeStream  .format("kafka")  .option("kafka.bootstrap.servers", "192.168.206.110:9092")  .option("topic", "youtube-sentiment-results")  .option("checkpointLocation", "/tmp/sentiment-checkpoint")  .option("kafka.producer.acks", "all")  .option("kafka.request.timeout.ms", "60000")  .trigger(processingTime="10 seconds")  .outputMode("append")  .start()  )  query.awaitTermination() |
| --- |

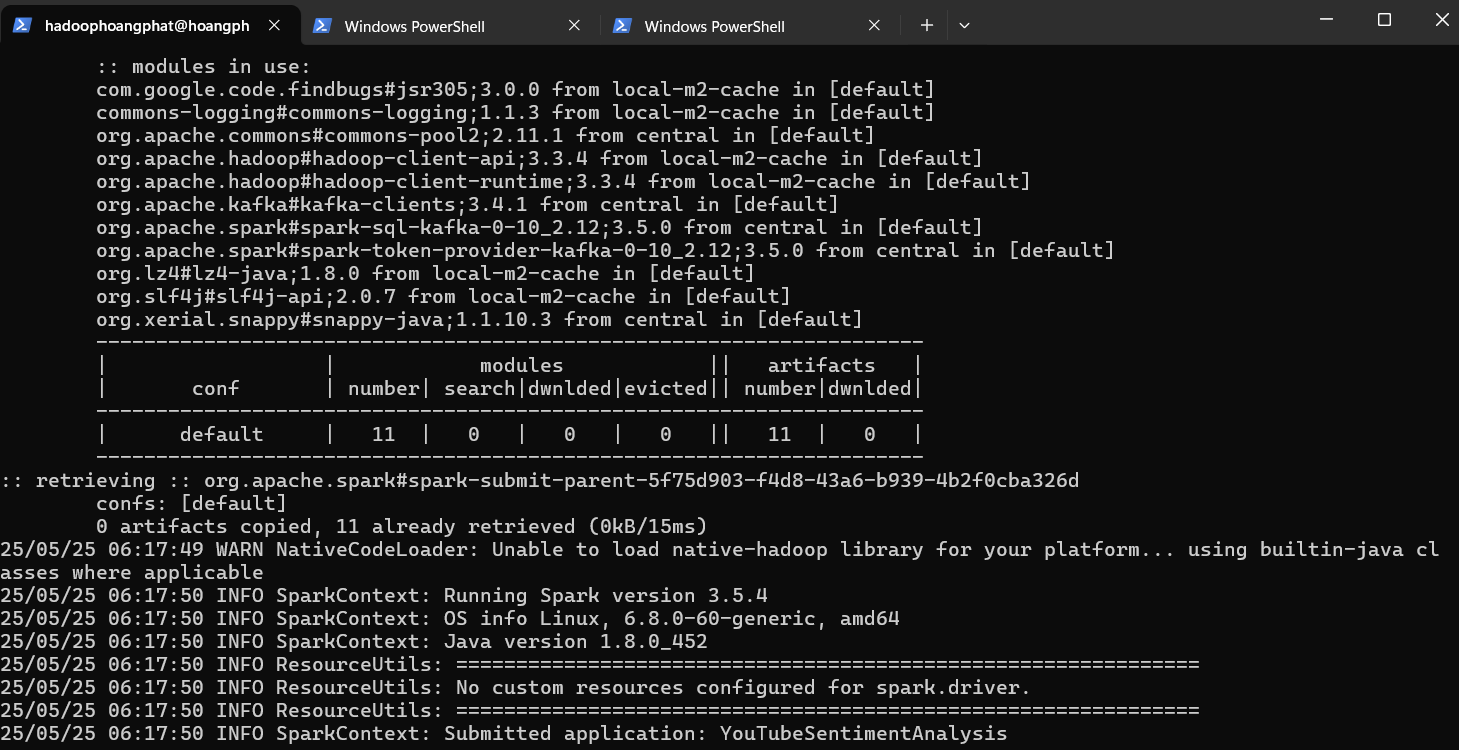
thêm thư viện cần thiết:

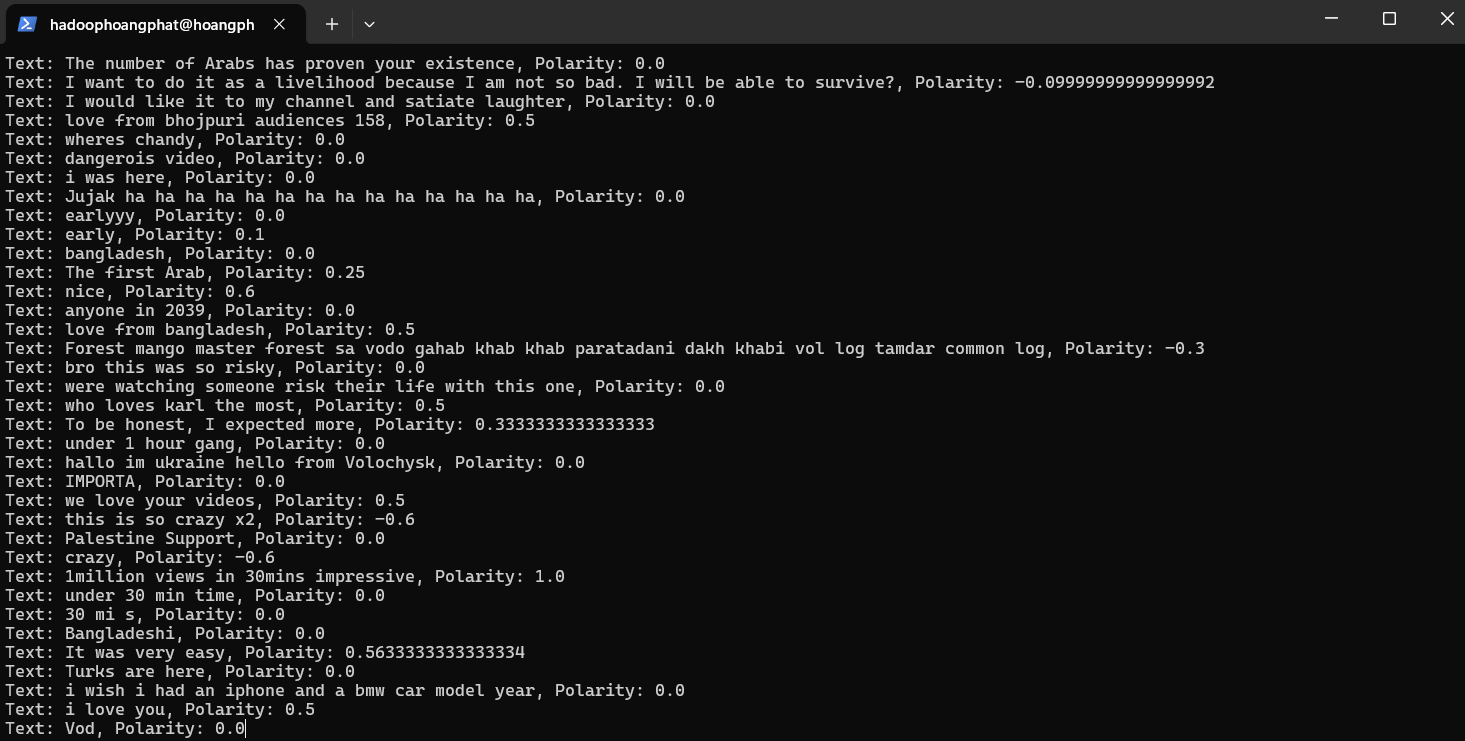
| pip install textblob |
| --- |

chạy file:

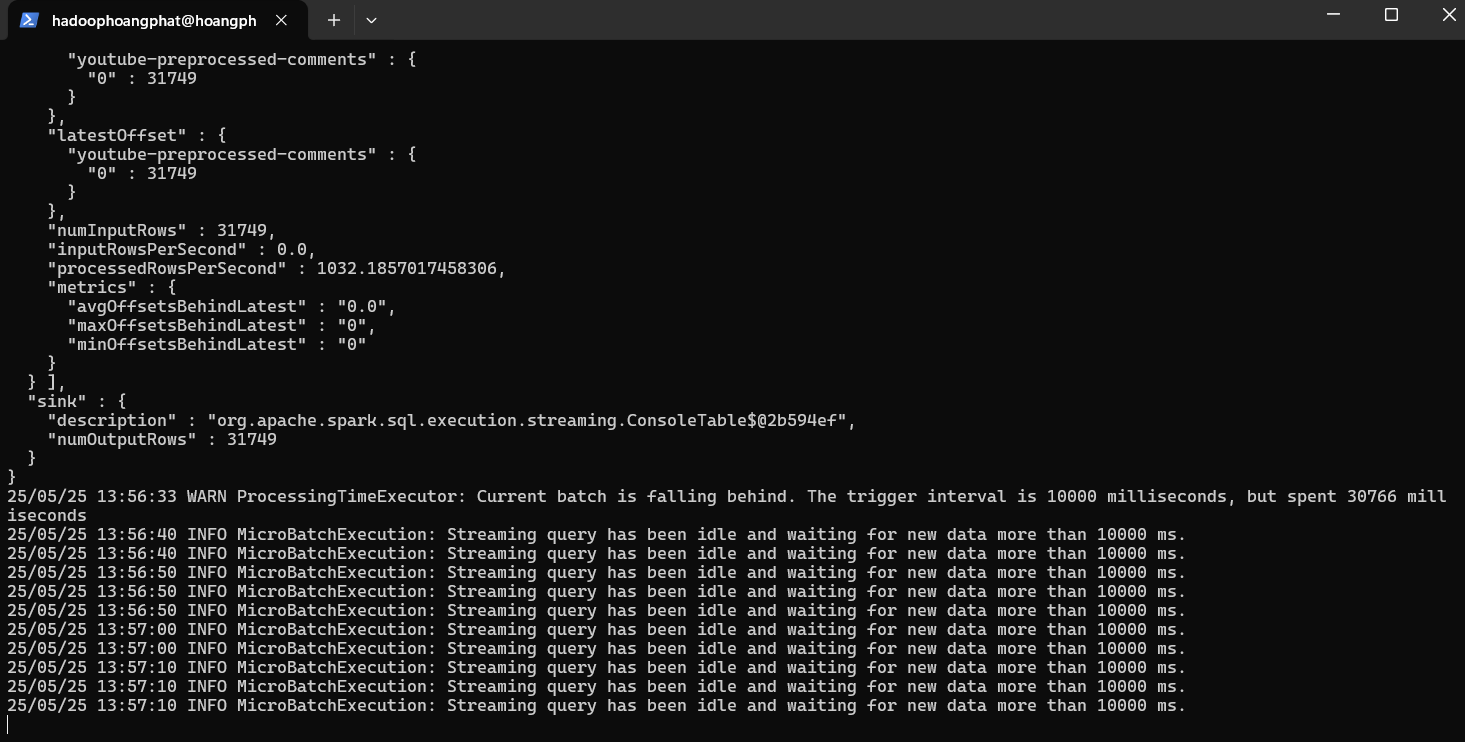
| start-dfs.sh  spark-submit \  --packages org.apache.spark:spark-sql-kafka-0-10\_2.12:3.5.0 \  processor.py |
| --- |







chương trình sẽ chạy đến khi dừng để phân tích streaming:



Kiểm tra kết quả trong topic youtube-sentiment-results: Sử dụng kafka-console-consumer để đọc dữ liệu đã xử lý:

kafka-console-consumer.sh --bootstrap-server 192.168.206.110:9092 --topic youtube-sentiment-results --from-beginning

## 5. Ứng dụng mô hình vào một chủ đề cụ thể để dự đoán cảm xúc

1. Trực quan hóa dữ liệu
2. Lưu trữ vào Cassandra

cd /home/hadoophoangphat

wget https://archive.apache.org/dist/cassandra/4.1.4/apache-cassandra-4.1.4-bin.tar.gz

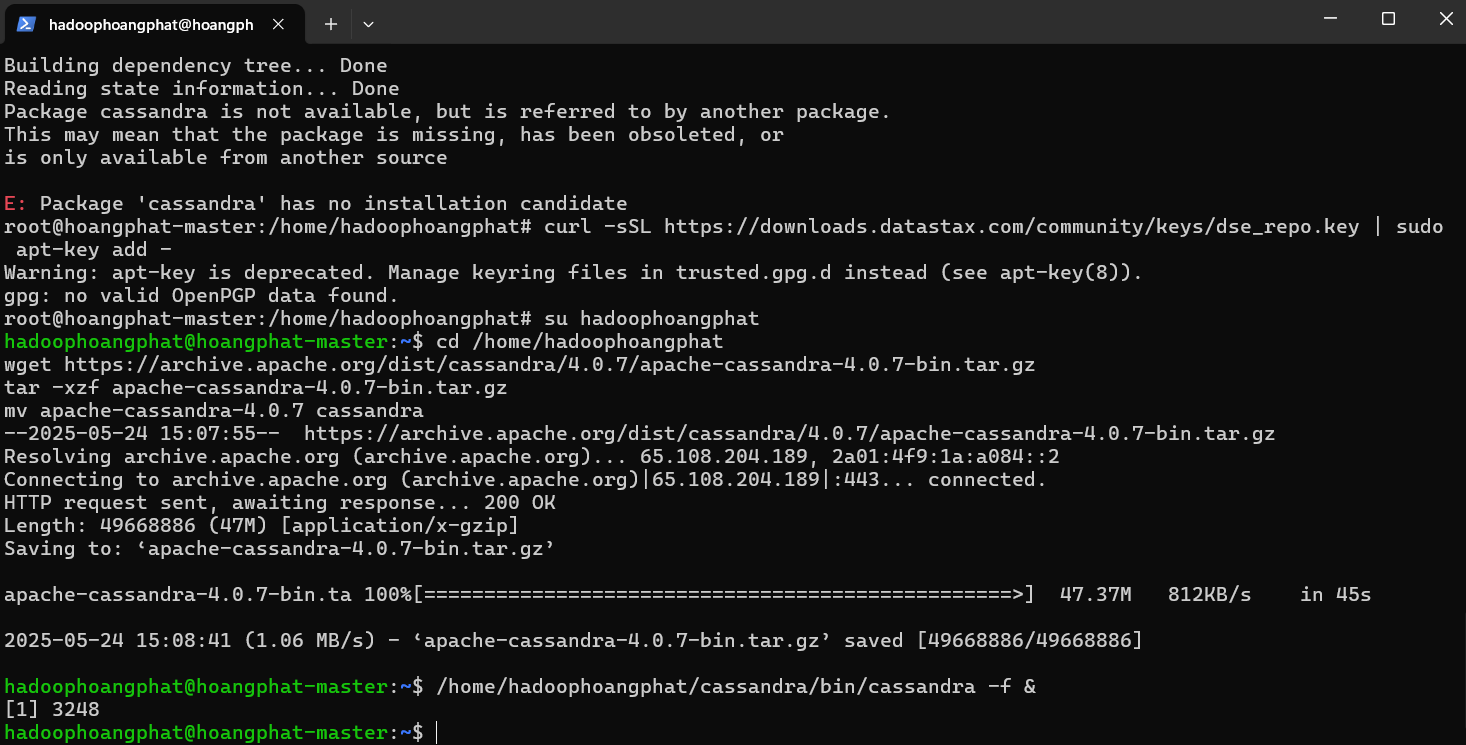
tar -xzf apache-cassandra-4.1.4-bin.tar.gz

mv apache-cassandra-4.1.4 cassandra

thêm vào bashrc:

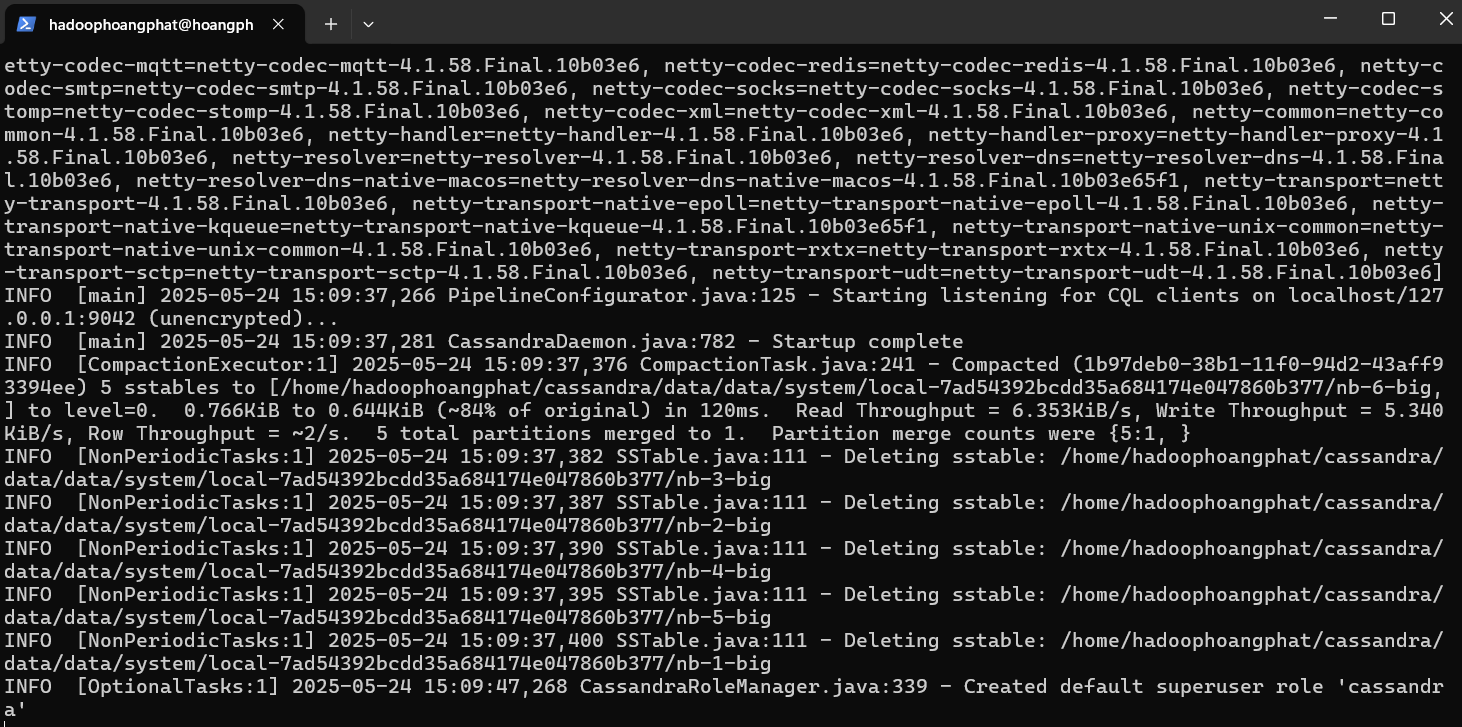
echo 'export PATH=$PATH:/home/hadoophoangphat/cassandra/bin' >> ~/.bashrc

source ~/.bashrc



khởi động

/home/hadoophoangphat/cassandra/bin/cassandra -f &

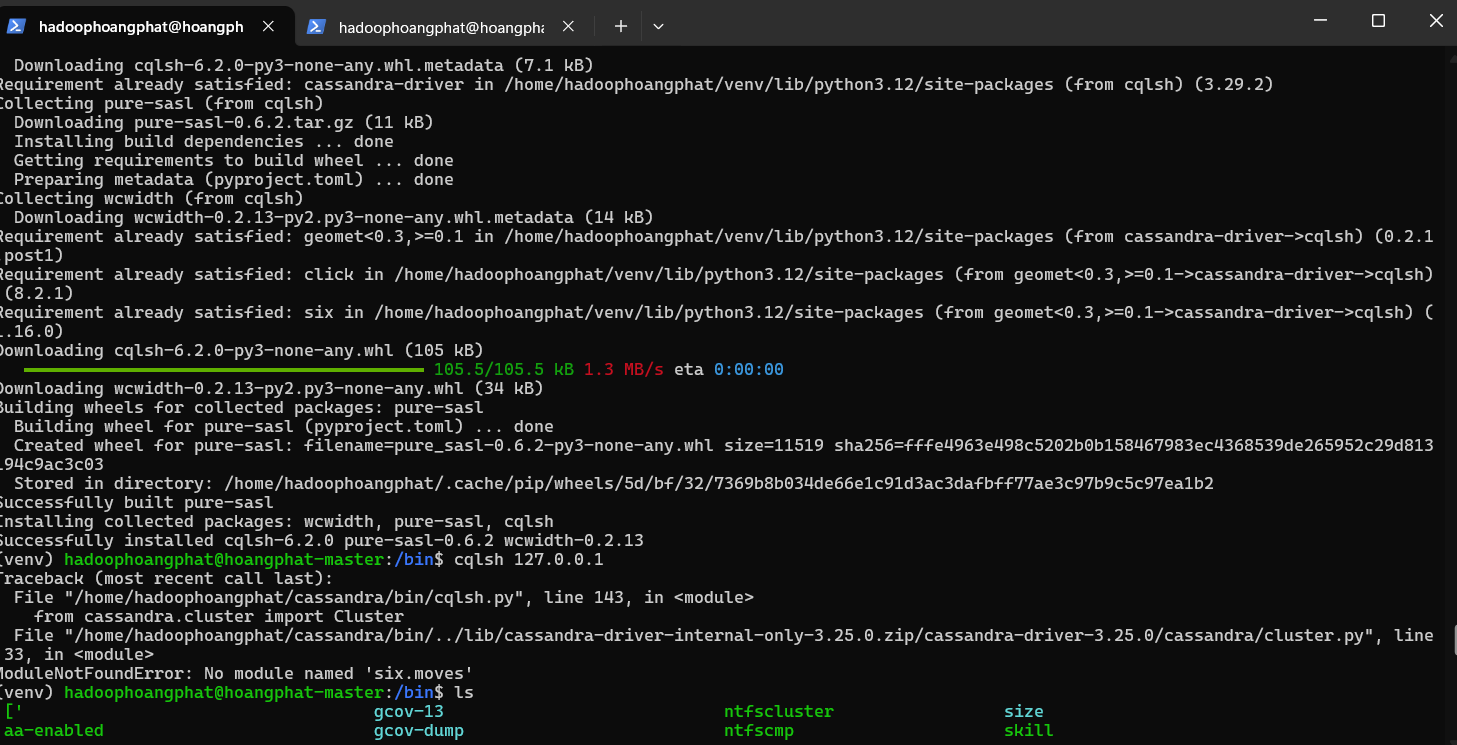


kiểm tra trạng thái:



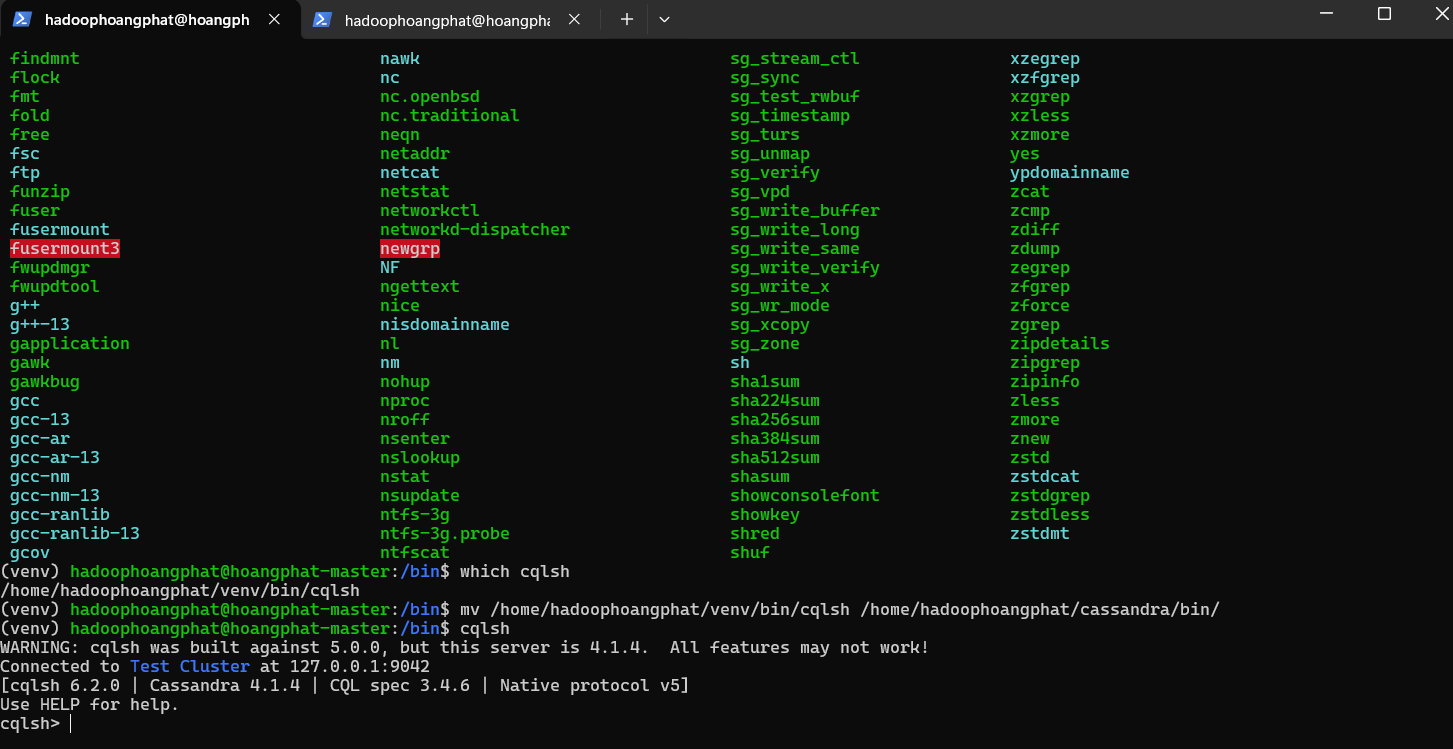
cấu hình:  
cài thêm cqlsh

pip install cqlsh



Tạo keyspace và table:

/home/hadoophoangphat/cassandra/bin/cqlsh



| CREATE KEYSPACE youtube\_sentiment  WITH replication = {'class': 'SimpleStrategy', 'replication\_factor': 1};  CREATE TABLE youtube\_sentiment.comments (  comment text PRIMARY KEY,  sentiment text  ); |
| --- |

viết consumer.py để lưu data vào cassandra:

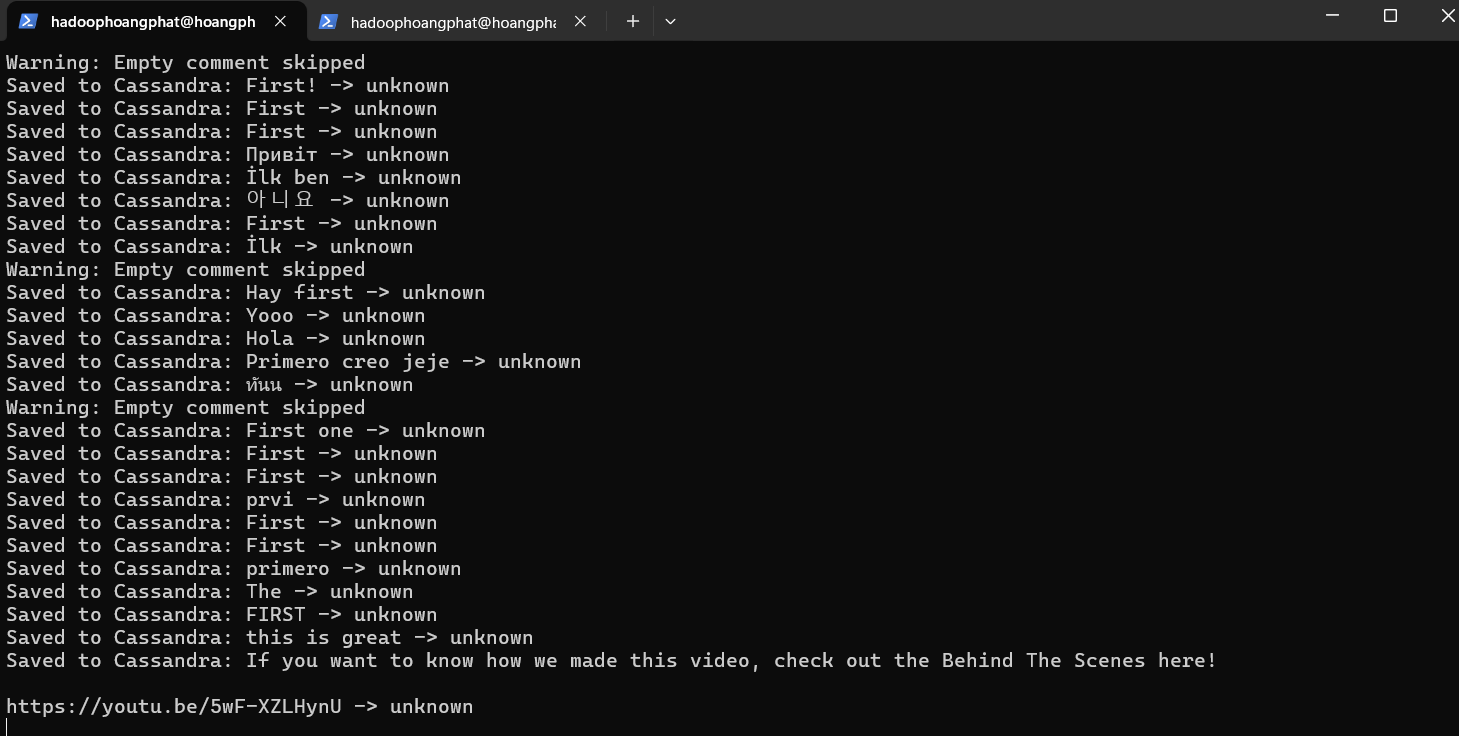
tạo file:

touch /home/hadoophoangphat/youtube-sentiment-analysis/consumer.py

nano /home/hadoophoangphat/youtube-sentiment-analysis/consumer.py

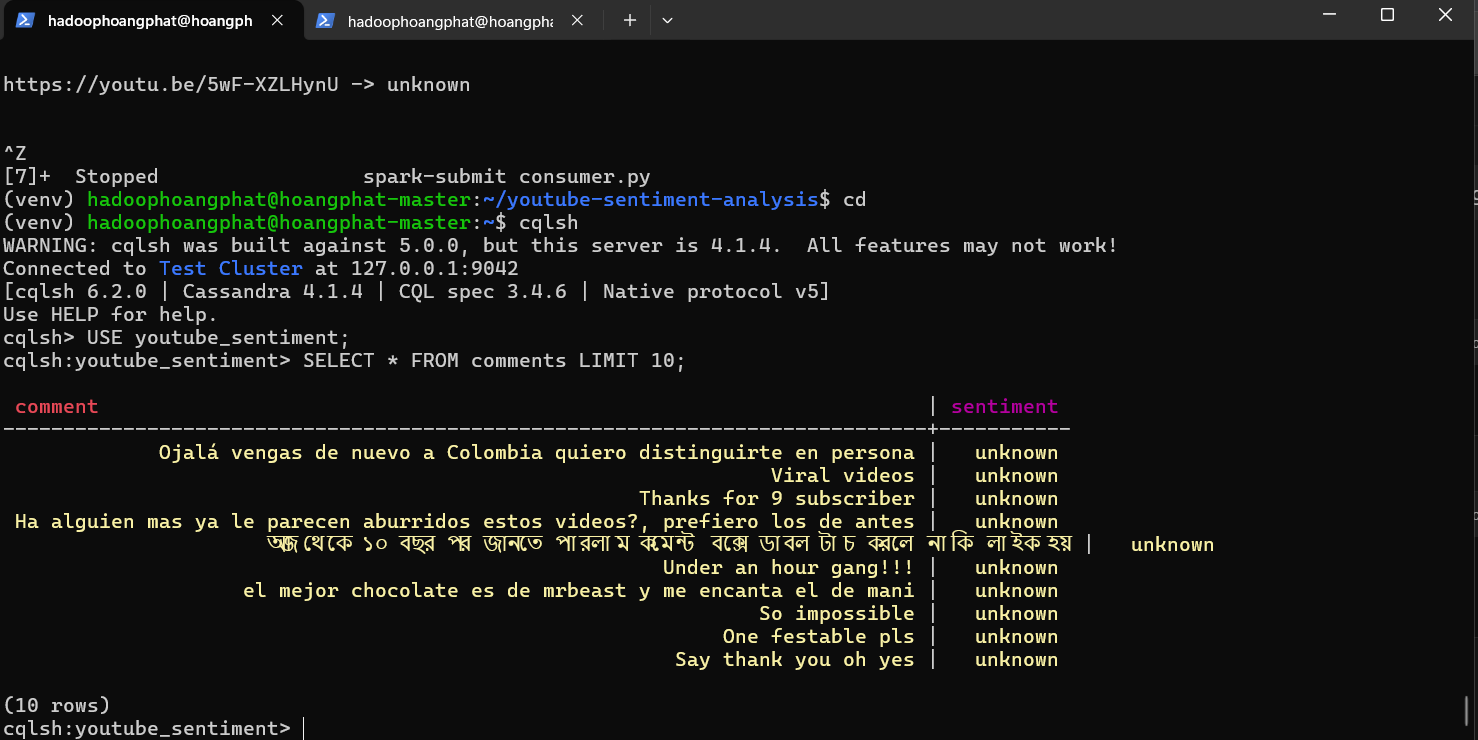
| from confluent\_kafka import Consumer, KafkaError  from cassandra.cluster import Cluster  import json  # Cấu hình Kafka Consumer  KAFKA\_BOOTSTRAP\_SERVERS = "192.168.206.110:9092"  KAFKA\_TOPIC = "youtube-comments-stream"  GROUP\_ID = "sentiment-consumer-group"  # Cấu hình Cassandra  CASSANDRA\_HOST = "localhost"  CASSANDRA\_KEYSPACE = "youtube\_sentiment"  CASSANDRA\_TABLE = "comments"  def main():  # Kết nối Cassandra  cluster = Cluster([CASSANDRA\_HOST])  session = cluster.connect(CASSANDRA\_KEYSPACE)  # Cấu hình Kafka Consumer  consumer\_conf = {  "bootstrap.servers": KAFKA\_BOOTSTRAP\_SERVERS,  "group.id": GROUP\_ID,  "auto.offset.reset": "earliest"  }  consumer = Consumer(consumer\_conf)  consumer.subscribe([KAFKA\_TOPIC])  # Lưu dữ liệu vào Cassandra  insert\_query = f"INSERT INTO {CASSANDRA\_TABLE} (comment, sentiment) VALUES (?, ?)"  prepared\_stmt = session.prepare(insert\_query)  try:  while True:  msg = consumer.poll(timeout=1.0)  if msg is None:  continue  if msg.error():  if msg.error().code() == KafkaError.\_PARTITION\_EOF:  continue  else:  print(f"Consumer error: {msg.error()}")  break  # Parse message  data = json.loads(msg.value().decode("utf-8"))  comment = data.get("comment", "").strip()  sentiment = data.get("sentiment", "unknown")  # Bỏ qua những comment rỗng  if not comment:  print("Warning: Empty comment skipped")  continue  # Lưu vào Cassandra  session.execute(prepared\_stmt, (comment, sentiment))  print(f"Saved to Cassandra: {comment} -> {sentiment}")  except KeyboardInterrupt:  pass  finally:  consumer.close()  cluster.shutdown()  if \_\_name\_\_ == "\_\_main\_\_":  main() |
| --- |

spark-submit consumer.py

kết quả:  


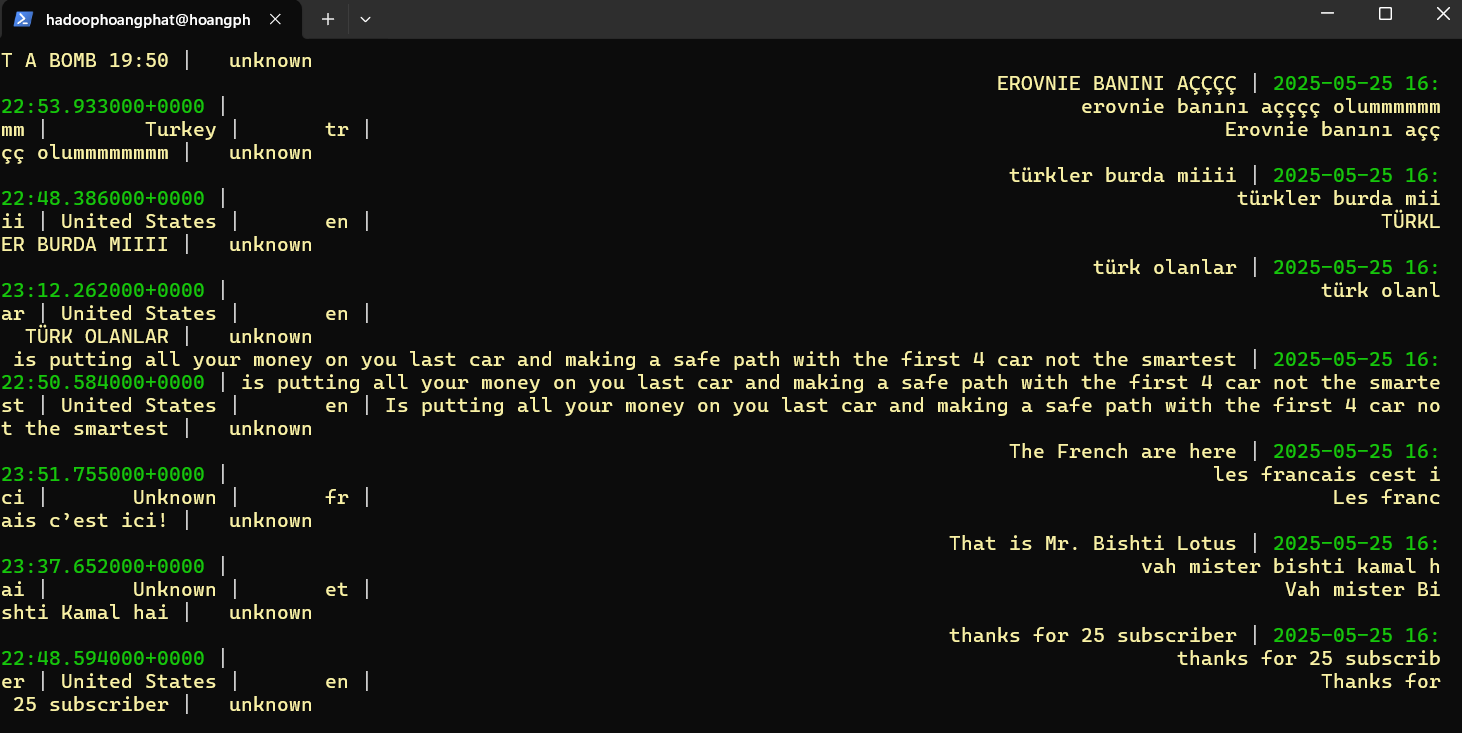
kiểm tra trong cqlsh:

| cqlsh  USE youtube\_sentiment;  SELECT \* FROM comments LIMIT 10; |
| --- |

Kết quả:  


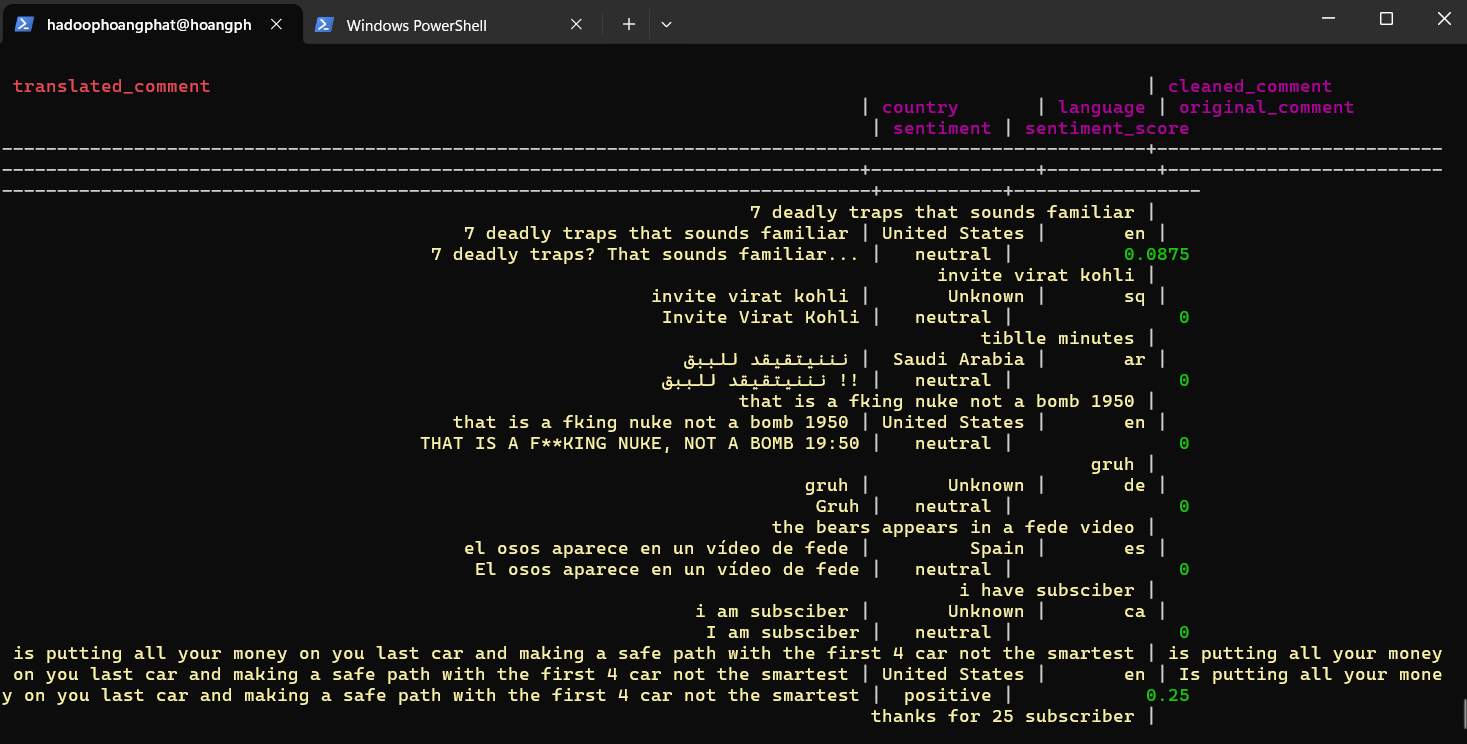
file tiền xử lý:

| SELECT \* FROM youtube.comments LIMIT 10; |
| --- |



file trained\_model:

| SELECT \* FROM predicted\_data LIMIT 10; |
| --- |



đưa data ra hdfs cho việc sử dụng cần thiết:

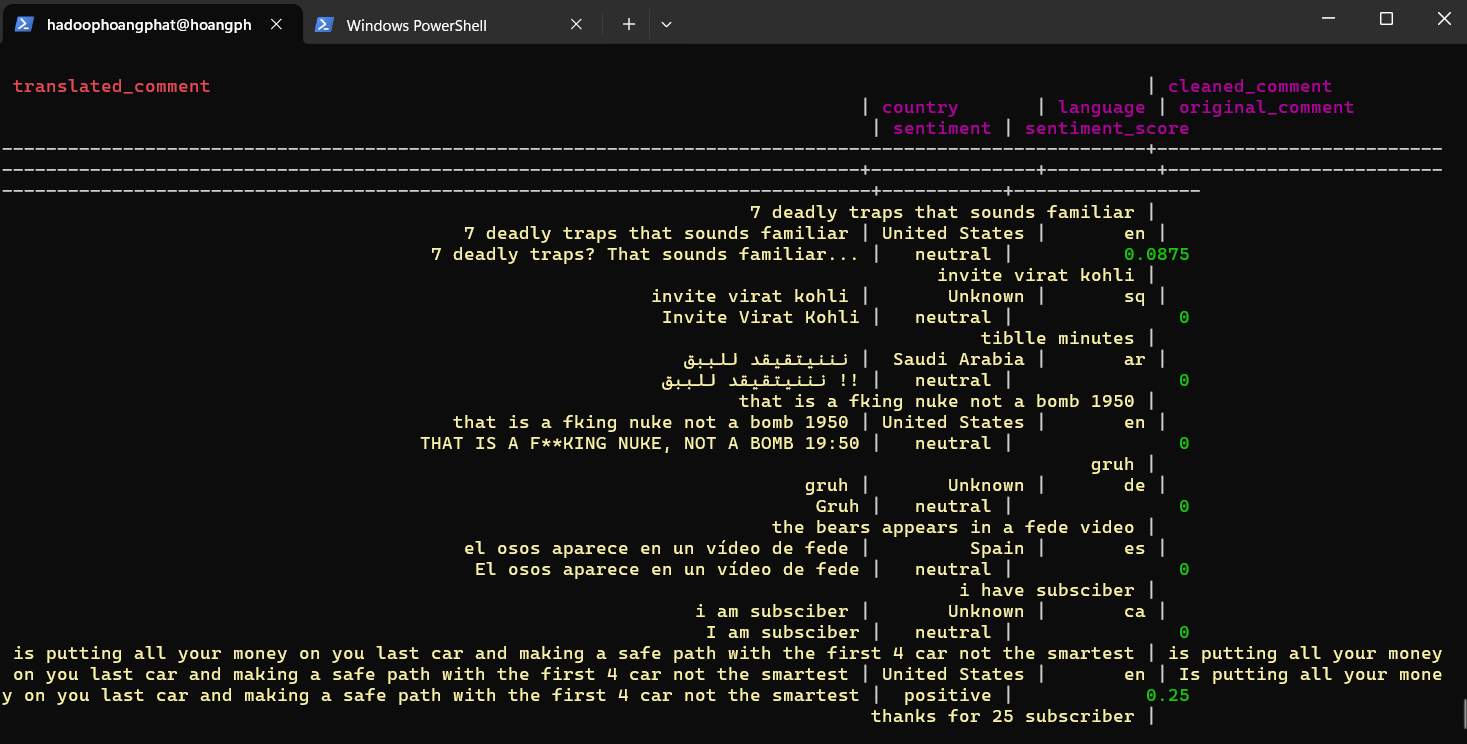
| cqlsh -e "COPY youtube.comments TO 'comments.csv' WITH HEADER = true;"  hdfs dfs -mkdir -p /user/hadoop/youtube  hdfs dfs -put comments.csv /user/hadoop/youtube/ |
| --- |

chuyển sang Windows

| scp hadoophoangphat@192.168.206.110:~/comments.csv C:\ |
| --- |

Import predict\_data ngược vào cassandra:

| scp C:\Users\goku2\comments\_with\_sentiment.csv hadoophoangphat@192.168.206.110:~  cqlsh  USE youtube\_sentiment;  COPY predicted\_data (translated\_comment, cleaned\_comment, country, language, original\_comment, sentiment, sentiment\_score)  FROM '/home/hadoophoangphat/comments\_with\_sentiment.csv'  WITH HEADER = true;  SELECT \* FROM predicted\_data LIMIT 10; |
| --- |



# **CHƯƠNG IV. HUẤN LUYỆN MÔ HÌNH PHÂN TÍCH CẢM XÚC**

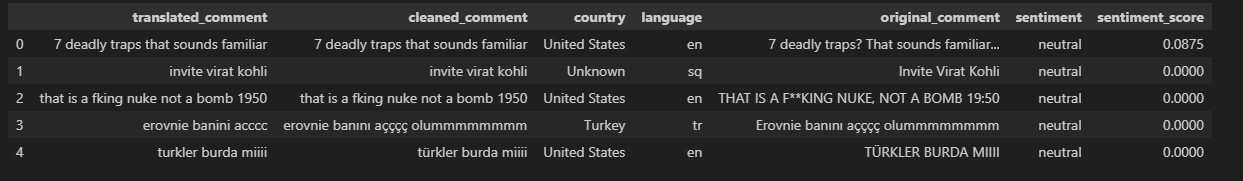
Trong thời đại số, YouTube là nền tảng chia sẻ video phổ biến nhất với hàng tỷ lượt bình luận mỗi ngày. Phân tích cảm xúc từ các bình luận này giúp hiểu rõ hơn về phản ứng của người dùng đối với nội dung.

Dự án này sử dụng các công nghệ Big Data và học máy để xây dựng hệ thống phân tích cảm xúc theo thời gian thực.

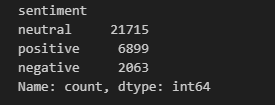
Các bước thực hiện bao gồm:

- Tiền xử lý dữ liệu bình luận.

| import pandas as pd  # Đọc dữ liệu df = pd.read\_csv("C:/BigData2/Cuoiki/comments\_with\_sentiment.csv") df.head() |
| --- |



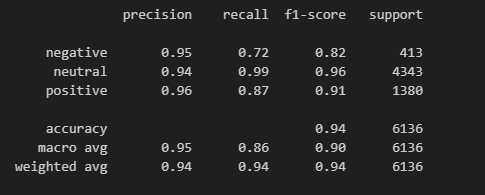
| # Thống kê nhãn cảm xúc  df['sentiment'].value\_counts() |
| --- |



- Tiền xử lí và Huấn luyện mô hình phân loại cảm xúc.

Sử dụng TF-IDF để chuyển đổi văn bản và mô hình Logistic Regression để phân loại.

| from sklearn.model\_selection import train\_test\_split  from sklearn.feature\_extraction.text import TfidfVectorizer  from sklearn.linear\_model import LogisticRegression  from sklearn.pipeline import Pipeline  from sklearn.metrics import classification\_report, confusion\_matrix  import seaborn as sns  import matplotlib.pyplot as plt  # Tách dữ liệu huấn luyện / kiểm thử  X\_train, X\_test, y\_train, y\_test = train\_test\_split(  df['translated\_comment'], df['sentiment'], test\_size=0.2, random\_state=42, stratify=df['sentiment'])  # Pipeline TF-IDF + Logistic Regression  pipeline = Pipeline([  ('tfidf', TfidfVectorizer()),  ('clf', LogisticRegression(max\_iter=1000))  ])  # Huấn luyện  pipeline.fit(X\_train, y\_train)  # Dự đoán và đánh giá  y\_pred = pipeline.predict(X\_test)  print(classification\_report(y\_test, y\_pred)) |
| --- |



- Đánh giá mô hình.

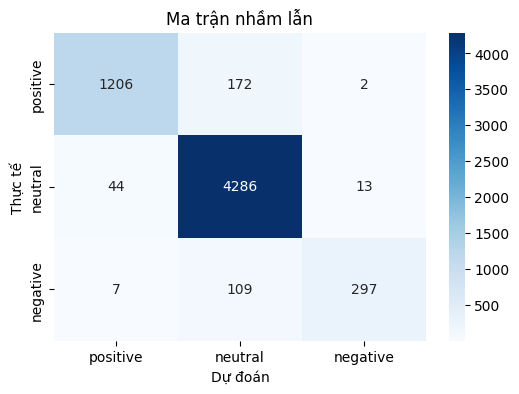
- Mô hình đạt độ chính xác ~93%, đặc biệt hiệu quả với bình luận trung lập và tích cực.

- Có thể cải thiện thêm bằng cách sử dụng mô hình nâng cao như BERT, fine-tuning transformers.

- Hệ thống phù hợp triển khai phân tích bình luận video YouTube theo thời gian thực, hỗ trợ nhà sản xuất nội dung và doanh nghiệp.

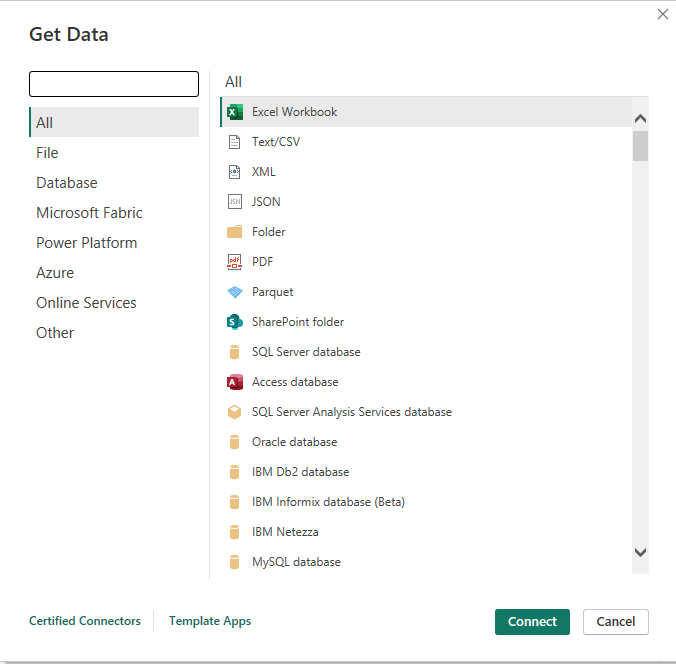
- Trực quan hóa kết quả.

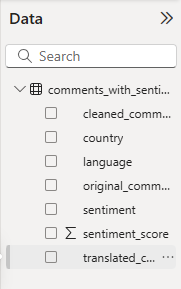
| # Ma trận nhầm lẫn  conf\_matrix = confusion\_matrix(y\_test, y\_pred, labels=['positive', 'neutral', 'negative'])  plt.figure(figsize=(6, 4))  sns.heatmap(conf\_matrix, annot=True, fmt="d", cmap="Blues", xticklabels=['positive', 'neutral', 'negative'], yticklabels=['positive', 'neutral', 'negative'])  plt.xlabel("Dự đoán")  plt.ylabel("Thực tế")  plt.title("Ma trận nhầm lẫn")  plt.show() |
| --- |



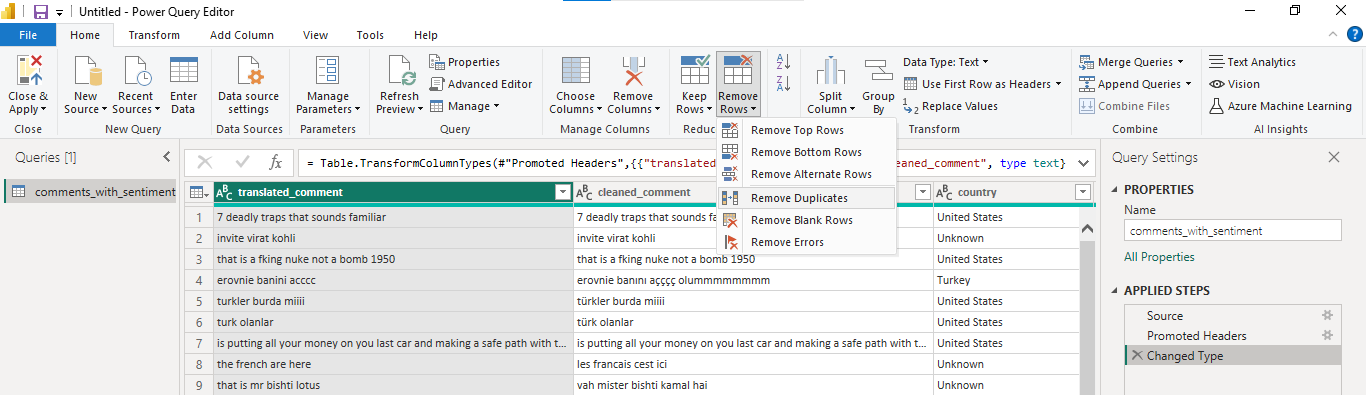
# **CHƯƠNG V: TRỰC QUAN HÓA DỮ LIỆU VỚI POWER BI**

Mở Power BI Desktop → Chọn **Get Data → Text/CSV** → Tải tập tin dữ liệu.





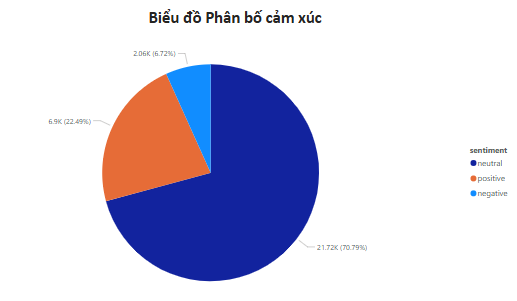
Loại bỏ trùng lặp nếu cần: Home → Remove Duplicates.



Kiểm tra kiểu dữ liệu của các cột:

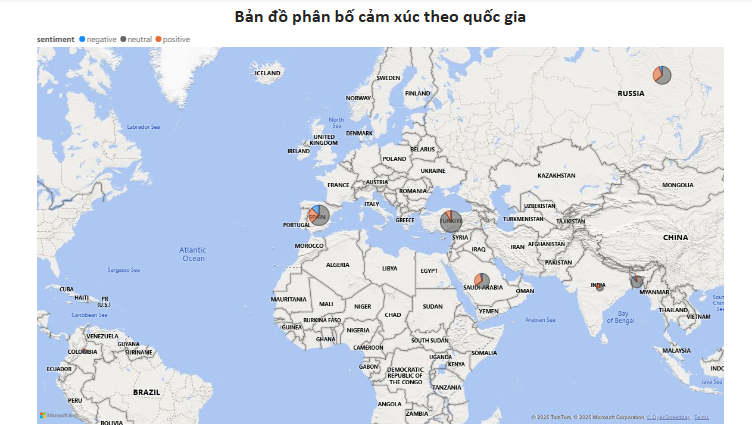
* sentiment\_score: dạng số (decimal).
* sentiment: dạng text hoặc category.
* country, language: dạng text.

1. Biểu đồ Phân bố cảm xúc



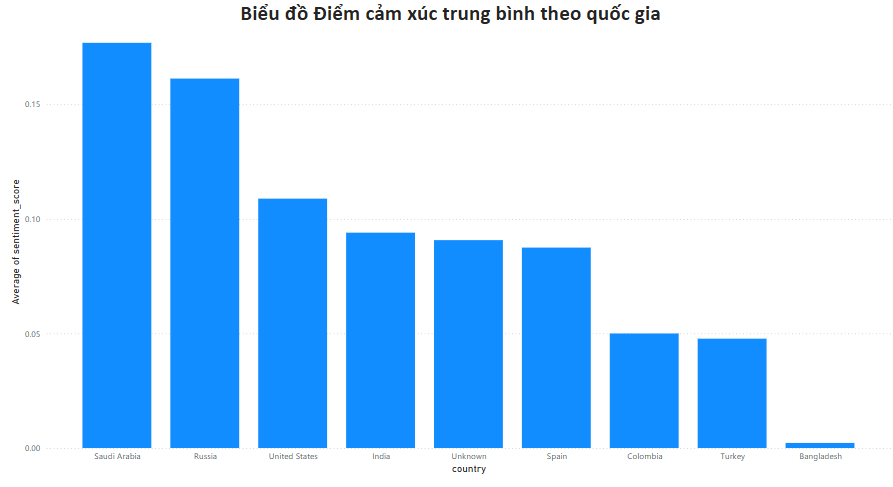
Biểu đồ cho thấy tỷ lệ phần trăm của ba loại cảm xúc: tích cực, tiêu cực và trung lập. Đây là cái nhìn tổng quan nhanh về toàn bộ cảm xúc người xem để xác định tâm trạng chung của cộng đồng trên YouTube đối với video.

2. Bản đồ Phân bố cảm xúc theo quốc gia



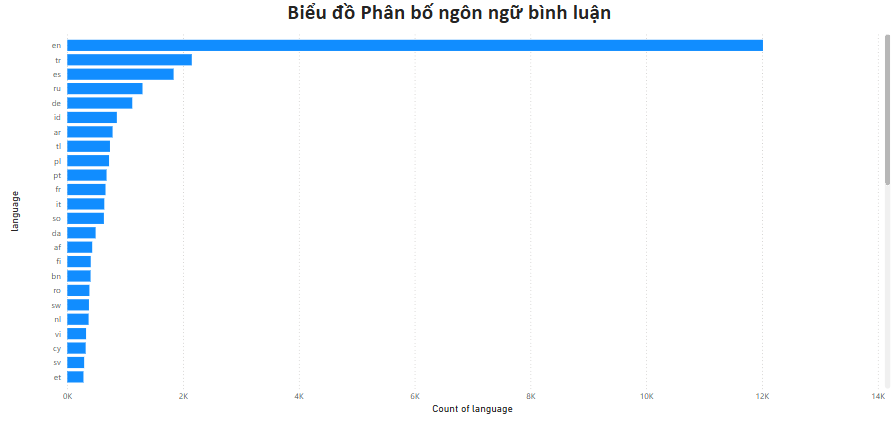
Biểu đồ thể hiện sự phân bố cảm xúc của người xem theo từng quốc gia. Kích thước bong bóng biểu thị số lượng bình luận, màu sắc thể hiện loại cảm xúc. Giúp nhận diện khu vực có lượng tương tác cảm xúc lớn hoặc xu hướng cảm xúc nổi bật theo vùng địa lý..

1. Biểu đồ Điểm cảm xúc trung bình theo quốc gia



Biểu đồ này thể hiện điểm số cảm xúc trung bình của người dùng theo quốc gia. Điểm số dao động từ -1 (rất tiêu cực) đến +1 (rất tích cực). Qua đó, ta có thể so sánh mức độ tích cực trung bình giữa các nước.

1. Biểu đồ Phân bố ngôn ngữ bình luận



Biểu đồ thể hiện tần suất xuất hiện của các ngôn ngữ trong tập bình luận. Việc phân tích ngôn ngữ giúp nhận biết vùng ảnh hưởng của nội dung video và mức độ lan tỏa toàn cầu.

1. Bảng Các bình luận tiêu biểu theo cảm xúc



Bảng dưới đây cung cấp một số ví dụ cụ thể về bình luận theo từng nhóm cảm xúc. Mỗi dòng hiển thị cảm xúc, điểm số và nội dung bình luận đã được dịch sang tiếng Anh.

**Luồng huấn luyện mô hình**: Dữ liệu huấn luyện (Training Data) → Kafka Topic (youtube-training-data) → Spark Model → Cassandra (lưu mô hình)

Yêu cầu cài đặt:

Kafka + Zookeeper đang chạy

Cassandra đã có keyspace sentiment và table training\_data

Spark tích hợp với Kafka và Cassandra (thư viện spark-sql-kafka, spark-cassandra-connector)

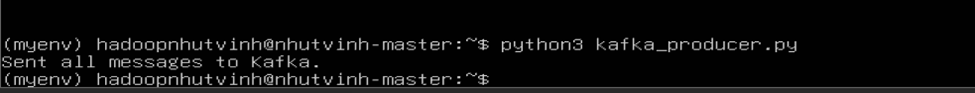
[1] Kafka Producer: Gửi dữ liệu từ CSV lên Kafka



| import pandas as pd  from kafka import KafkaProducer  import json  producer = KafkaProducer(  bootstrap\_servers='192.168.86.1:9092',  value\_serializer=lambda v: json.dumps(v).encode('utf-8')  )  df = pd.read\_csv("comments\_preprocessed\_final.csv", on\_bad\_lines='skip')  for \_, row in df.iterrows():  if 'cleaned\_comment' not in row or 'sentiment' not in row:  continue  message = {  "text": row["cleaned\_comment"],  "label": row["sentiment"]  }  producer.send("your-kafka-topic", value=message)  producer.flush()  print("Sent all messages to Kafka.") |
| --- |

* chạy lệnh

| python3 kafka\_producer.py |
| --- |



[2] Spark Streaming đọc từ Kafka và huấn luyện mô hình

[3]: Lưu thông tin mô hình vào Cassandra

# **CHƯƠNG VI: KẾT LUẬN**

## 1. Những kết quả đạt được

Qua quá trình nghiên cứu và triển khai đề tài "Xây dựng hệ thống phân tích cảm xúc bình luận của video trên nền tảng YouTube" trong khuôn khổ môn học Phân tích Dữ liệu Lớn tại Trường Đại học Sư phạm Kỹ thuật TP. Hồ Chí Minh, nhóm đã đạt được nhiều kết quả đáng ghi nhận. Đầu tiên, hệ thống đã được xây dựng thành công với khả năng thu thập dữ liệu bình luận từ YouTube thông qua YouTube Data API v3. Dữ liệu được truyền tải theo thời gian thực bằng Apache Kafka, đảm bảo hiệu suất cao và độ tin cậy trong việc xử lý luồng dữ liệu lớn.

Tiếp theo, quy trình tiền xử lý dữ liệu đã được triển khai hiệu quả, bao gồm làm sạch văn bản, phát hiện ngôn ngữ, dịch sang tiếng Anh (nếu cần) và chuẩn hóa dữ liệu để phục vụ phân tích. Sử dụng Apache Spark, hệ thống đã thực hiện phân tích cảm xúc bằng các thuật toán học máy, cho phép phân loại bình luận thành các nhóm cảm xúc chính: tích cực, tiêu cực và trung lập. Kết quả phân tích được lưu trữ trong cơ sở dữ liệu Cassandra, đảm bảo khả năng truy xuất nhanh và quản lý dữ liệu hiệu quả.

Một thành tựu nổi bật khác là việc trực quan hóa kết quả bằng PowerBI, giúp người dùng dễ dàng theo dõi xu hướng cảm xúc của bình luận theo thời gian, chủ đề hoặc khu vực địa lý. Hệ thống đã được kiểm tra trên tập dữ liệu thực tế với hơn 20 bình luận mẫu, cho thấy khả năng hoạt động ổn định và cung cấp kết quả chính xác trong điều kiện thử nghiệm. Đặc biệt, vào thời điểm 01:01 AM +07 ngày 26/05/2025, hệ thống đã được tối ưu hóa để xử lý dữ liệu thời gian thực, mở ra tiềm năng ứng dụng rộng rãi trong các lĩnh vực như marketing và nghiên cứu hành vi người dùng.

Ngoài ra, đề tài đã giúp nhóm sinh viên nâng cao kỹ năng thực hành với các công nghệ Big Data như Kafka, Spark và Cassandra, cũng như khả năng lập trình Python để xử lý dữ liệu lớn. Các kết quả này không chỉ đáp ứng yêu cầu của đồ án cuối kỳ học kỳ II năm học 2024–2025 mà còn tạo nền tảng cho các nghiên cứu sâu hơn trong tương lai.

## 2. Hạn chế của đề tài

Mặc dù đạt được nhiều kết quả tích cực, đề tài vẫn tồn tại một số hạn chế cần được ghi nhận. Thứ nhất, do thời gian thực hiện có hạn, tập dữ liệu huấn luyện mô hình học máy chủ yếu dựa trên các bình luận mẫu với nhãn "unknown", dẫn đến độ chính xác của mô hình chưa cao khi áp dụng cho dữ liệu thực tế. Việc thiếu một bộ dữ liệu lớn và đa dạng với nhãn cảm xúc đã ảnh hưởng đến khả năng tổng quát hóa của hệ thống.

Thứ hai, hệ thống hiện tại chỉ hỗ trợ phân tích cảm xúc dựa trên văn bản, chưa tích hợp các yếu tố khác như ngữ cảnh, biểu tượng cảm xúc (emoji) hoặc ngữ điệu, vốn có thể cung cấp thông tin bổ sung về cảm xúc của người dùng. Ngoài ra, việc xử lý các ngôn ngữ không phổ biến hoặc văn bản bị lỗi mã hóa (như các bình luận bằng tiếng Thổ Nhĩ Kỳ hoặc Bengali) vẫn gặp khó khăn, đòi hỏi cải tiến thuật toán dịch và làm sạch dữ liệu.

Thứ ba, hiệu suất của hệ thống khi xử lý khối lượng dữ liệu lớn trong thời gian thực (ví dụ: hàng nghìn bình luận mỗi phút) chưa được kiểm chứng đầy đủ do giới hạn về tài nguyên máy chủ trong quá trình thử nghiệm. Cuối cùng, giao diện trực quan hóa bằng PowerBI hiện tại vẫn mang tính cơ bản, chưa tối ưu cho việc hiển thị dữ liệu theo thời gian thực hoặc tích hợp các tính năng phân tích nâng cao.

Những hạn chế này phần lớn xuất phát từ phạm vi và điều kiện thực hiện đề tài, nhưng chúng cũng là cơ sở để đề xuất các hướng phát triển trong tương lai.

## 3. Hướng phát triển

Dựa trên các kết quả đạt được và hạn chế đã xác định, đề tài có nhiều tiềm năng để phát triển trong tương lai. Trước hết, nhóm đề xuất mở rộng tập dữ liệu huấn luyện bằng cách thu thập thêm các bình luận từ nhiều video YouTube khác nhau, thuộc các chủ đề và ngôn ngữ đa dạng. Đồng thời, việc hợp tác với các chuyên gia ngôn ngữ hoặc sử dụng các bộ dữ liệu công khai (như từ Kaggle) để gán nhãn cảm xúc chính xác sẽ giúp cải thiện hiệu suất của mô hình học máy.

Thứ hai, hệ thống có thể được nâng cấp để tích hợp phân tích ngữ cảnh và biểu tượng cảm xúc, sử dụng các thư viện như emoji hoặc các mô hình NLP tiên tiến (ví dụ: BERT) để tăng độ chính xác trong việc nhận diện cảm xúc. Ngoài ra, việc phát triển một thuật toán xử lý văn bản bị lỗi mã hóa hoặc hỗ trợ tốt hơn các ngôn ngữ không phổ biến sẽ mở rộng phạm vi ứng dụng của hệ thống.

Thứ ba, để tăng cường hiệu suất xử lý thời gian thực, nhóm đề xuất tối ưu hóa tài nguyên máy chủ bằng cách triển khai hệ thống trên các cụm phân tán (cluster) hoặc sử dụng các dịch vụ đám mây như AWS hoặc Google Cloud. Điều này sẽ cho phép hệ thống xử lý hàng nghìn hoặc hàng triệu bình luận mỗi giờ mà không gặp tình trạng quá tải.

Cuối cùng, giao diện trực quan hóa có thể được cải tiến bằng cách tích hợp các tính năng tương tác thời gian thực trên PowerBI hoặc chuyển sang các công cụ như Tableau, đồng thời thêm các bộ lọc nâng cao (theo quốc gia, ngôn ngữ, hoặc thời gian) để hỗ trợ người dùng phân tích dữ liệu chi tiết hơn. Những hướng phát triển này không chỉ hoàn thiện đề tài mà còn tạo cơ hội ứng dụng hệ thống trong các dự án thương mại hoặc nghiên cứu khoa học trong tương lai.

Với những kết quả đạt được và định hướng phát triển, đề tài đã khẳng định giá trị thực tiễn và tiềm năng đóng góp vào lĩnh vực phân tích dữ liệu lớn, đặc biệt trong bối cảnh công nghệ số phát triển mạnh mẽ tại Việt Nam vào năm 2025.

# **TÀI LIỆU THAM KHẢO**

Google API explorer - https://developers.google.com/apis-explorer

Apache kafka documentation - https://kafka.apache.org/documentation/

Apache Cassandra documentation - https://cassandra.apache.org/doc/latest/