

BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC CÔNG THƯƠNG TP. HCM
KHOA CÔNG NGHỆ THÔNG TIN



PHÂN TÍCH VÀ DỰ ĐOÁN XU HƯỚNG VIDEO TRÊN YOUTUBE

Đồ án môn học: Big Data

SINH VIÊN THỰC HIỆN:
2001222641 – Trần Công Minh

2001225676 – Lê Đức Trung

TP. HỒ CHÍ MINH, THÁNG 09 NĂM 2025

BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC CÔNG THƯƠNG TP. HCM
KHOA CÔNG NGHỆ THÔNG TIN



**PHÂN TÍCH VÀ DỰ ĐOÁN XU
HƯỚNG VIDEO TRÊN YOUTUBE**

Đồ án môn học: Big Data

SINH VIÊN THỰC HIỆN:
2001222641 – Trần Công Minh

2001225676 – Lê Đức Trung

TP. HỒ CHÍ MINH, THÁNG 09 NĂM 2025

MỤC LỤC

CHƯƠNG 1. GIỚI THIỆU	1
1.1. Lý do chọn đề tài.....	1
1.2. Mục tiêu nghiên cứu	2
1.3. Phạm vi và đóng góp.....	2
CHƯƠNG 2. CƠ SỞ LÝ THUYẾT VÀ TỔNG QUAN	4
2.1. YouTube Trending và ý nghĩa	4
2.2. Big Data: khái niệm và công nghệ chính.....	5
2.3. Apache Spark và hệ sinh thái.....	6
2.4. Tổng quan các nghiên cứu liên quan	7
CHƯƠNG 3. PHÂN TÍCH YÊU CẦU VÀ THIẾT KẾ HỆ THỐNG ...	9
3.1. Yêu cầu chức năng	9
3.2. Yêu cầu phi chức năng.....	9
3.3. Kiến trúc tổng thể.....	10
3.4. Sơ đồ luồng dữ liệu (Data Flow)	12
3.4.1. Luồng xử lý và huấn luyện (Offline)	12
3.4.2. Luồng dự đoán và tương tác (Online).....	13
CHƯƠNG 4. THU THẬP VÀ TIỀN XỬ LÝ DỮ LIỆU	14
4.1. Nguồn dữ liệu.....	14
4.2. Kịch bản thu thập bằng Python/Spark	14
4.3. Làm sạch và xử lý thiếu dữ liệu	15
4.4. Lưu trữ tạm thời (MongoDB/File).....	15
CHƯƠNG 5. KHÁM PHÁ VÀ PHÂN TÍCH DỮ LIỆU (EDA)	17

5.1. Thông kê mô tả	17
5.2. Trực quan hóa xu hướng theo quốc gia, thể loại	18
5.3. Phát hiện bất thường và mối liên hệ	19
CHƯƠNG 6. XÂY DỰNG MÔ HÌNH DỰ ĐOÁN	20
6.1. Chuẩn bị dữ liệu (Feature Engineering)	20
6.2. Lựa chọn và huấn luyện mô hình.....	21
6.3. Đánh giá hiệu năng	23
CHƯƠNG 7. TRIỂN KHAI ỨNG DỤNG	25
7.1. Backend API với FastAPI.....	25
7.2. Dịch vụ dự đoán (ML Service)	26
7.3. Giao diện người dùng (Frontend)	27
CHƯƠNG 8. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	30
8.1. Kết luận	30
8.2. Hạn chế	31
8.3. Hướng phát triển	31
TÀI LIỆU THAM KHẢO.....	34

BẢNG PHÂN CÔNG CÔNG VIỆC

MSSV	Tên	Công việc	Đánh giá
2001222641	Trần Công Minh	<ul style="list-style-type: none"> • Xây dựng và vận hành pipeline dữ liệu (ETL) và feature engineering. • Huấn luyện, lưu trữ và kiểm chứng mô hình ML. • Triển khai/điều phối dịch vụ dự đoán (API) và đảm bảo hoạt động backend. • Deliverables: dataset đã xử lý, model đã huấn luyện, endpoint dự đoán hoạt động. 	100%
2001225676	Lê Đức Trung	<ul style="list-style-type: none"> • Thiết kế và triển khai giao diện người dùng cho phân tích và dự đoán. • Tích hợp UI với API (gửi dữ liệu dự đoán, hiển thị kết quả và visualizations). • Viết hướng dẫn sử dụng ngắn và kiểm thử flow người dùng cơ bản. • Deliverables: giao diện hoạt động, flow dự đoán end-to-end, tài liệu hướng dẫn ngắn. 	100%

DANH MỤC HÌNH

Hình 1. Sơ đồ kiến trúc tổng thể	10
Hình 2. Luồng xử lý ngoại tuyến	12
Hình 3. Luồng dự đoán trực tuyến	13
Hình 4. Các chỉ số đánh giá	23
Hình 5. Ma trận nhầm lẫn	24
Hình 6. Biểu đồ cột thể hiện độ quan trọng của các đặc trưng (Feature Importance)	25
Hình 7. Giao diện trang Phân tích Trending	28
Hình 8. Giao diện trang dự đoán Trending	29

DANH MỤC BẢNG

Bảng 1. Bảng đặc trưng tương tác (Engagement Metrics)	20
Bảng 2. Bảng đặc trưng nội dung (Content Features)	21
Bảng 3. Bảng đặc trưng kênh (Channel Features):	21

CHƯƠNG 1. GIỚI THIỆU

Trong bối cảnh bùng nổ của nội dung số, YouTube đã trở thành một trong những nền tảng chia sẻ video lớn nhất thế giới, thu hút hàng tỷ người dùng mỗi ngày. Việc một video lọt vào danh sách "Thịnh hành" (Trending) không chỉ mang lại danh tiếng cho người sáng tạo mà còn tạo ra những cơ hội to lớn về mặt thương mại và tầm ảnh hưởng. Tuy nhiên, cơ chế đằng sau việc lựa chọn video thịnh hành vẫn là một ẩn số phức tạp, phụ thuộc vào vô số yếu tố tương tác với nhau.

1.1. Lý do chọn đề tài

Đề tài "Phân tích và Dự đoán Xu hướng Video trên YouTube bằng Big Data" được chọn vì những lý do sau:

- a) **Tính thời sự và thực tiễn:** Việc hiểu được các yếu tố then chốt giúp một video trở nên thịnh hành là một bài toán có giá trị thực tiễn cao, giúp các nhà sáng tạo nội dung, các nhà tiếp thị và nhà phân tích tối ưu hóa chiến lược của mình.
- b) **Thách thức về dữ liệu lớn (Big Data):** YouTube tạo ra một lượng dữ liệu khổng lồ mỗi giây. Việc xử lý, phân tích và rút ra tri thức từ tập dữ liệu này đòi hỏi phải áp dụng các công nghệ và kỹ thuật của Big Data, điển hình là Apache Spark. Đây là cơ hội tuyệt vời để áp dụng kiến thức môn học vào một bài toán thực tế.
- c) **Sự phức tạp và hấp dẫn của bài toán:** Dự đoán xu hướng không chỉ đơn thuần dựa vào lượt xem hay lượt thích, mà còn là sự kết hợp của nhiều yếu tố như thời gian đăng, tiêu đề, mô tả, thẻ (tags), và cả phản ứng của cộng đồng. Việc xây dựng một mô hình dự đoán cho bài toán này mang tính thách thức và rất thú vị về mặt khoa học dữ liệu.

1.2. Mục tiêu nghiên cứu

Đồ án tập trung vào các mục tiêu chính sau:

- Xây dựng một hệ thống xử lý dữ liệu lớn hoàn chỉnh (data pipeline) để thu thập, làm sạch, và xử lý dữ liệu về các video YouTube.
- Thực hiện Phân tích Dữ liệu Khám phá (EDA) để tìm ra các đặc điểm, quy luật và yếu tố ảnh hưởng đến khả năng một video lọt vào danh sách thịnh hành.
- Xây dựng và huấn luyện mô hình học máy (Machine Learning) có khả năng dự đoán một video có trở thành video thịnh hành hay không dựa trên các thuộc tính của nó.
- Phát triển một ứng dụng web (Web Application) trực quan để trình bày các kết quả phân tích và cho phép người dùng tương tác, thử nghiệm với mô hình dự đoán.

1.3. Phạm vi và đóng góp

- Phạm vi:
 - Dữ liệu: Sử dụng các bộ dữ liệu công khai về video thịnh hành trên YouTube (dạng file CSV).
 - Công nghệ: Tập trung vào hệ sinh thái công nghệ bao gồm: Apache Spark để xử lý dữ liệu phân tán, Python (với các thư viện như Pandas, Scikit-learn) để xây dựng mô hình, FastAPI để xây dựng backend API, React và TailwindCSS để phát triển frontend, và Docker/Docker-Compose để đóng gói và triển khai hệ thống.
 - Bài toán: Giới hạn ở việc dự đoán khả năng thịnh hành dựa trên các siêu dữ liệu (metadata) của video, không phân tích nội dung hình ảnh hay âm thanh.
- Đóng góp:

- Xây dựng thành công một hệ thống end-to-end, minh họa toàn bộ vòng đời của một dự án Big Data, từ khâu thu thập dữ liệu thô đến khi ra được sản phẩm là một ứng dụng web có giá trị.
- Cung cấp một cái nhìn sâu sắc, dựa trên dữ liệu, về các yếu tố quyết định sự thành công của một video trên nền tảng YouTube.
- Tạo ra một mô hình dự đoán có thể được xem như một công cụ tham khảo hữu ích cho các nhà sáng tạo nội dung.
- Đóng góp một bộ mã nguồn mở, được kiến trúc tốt, có thể làm tài liệu học tập và tham khảo cho các dự án tương tự.

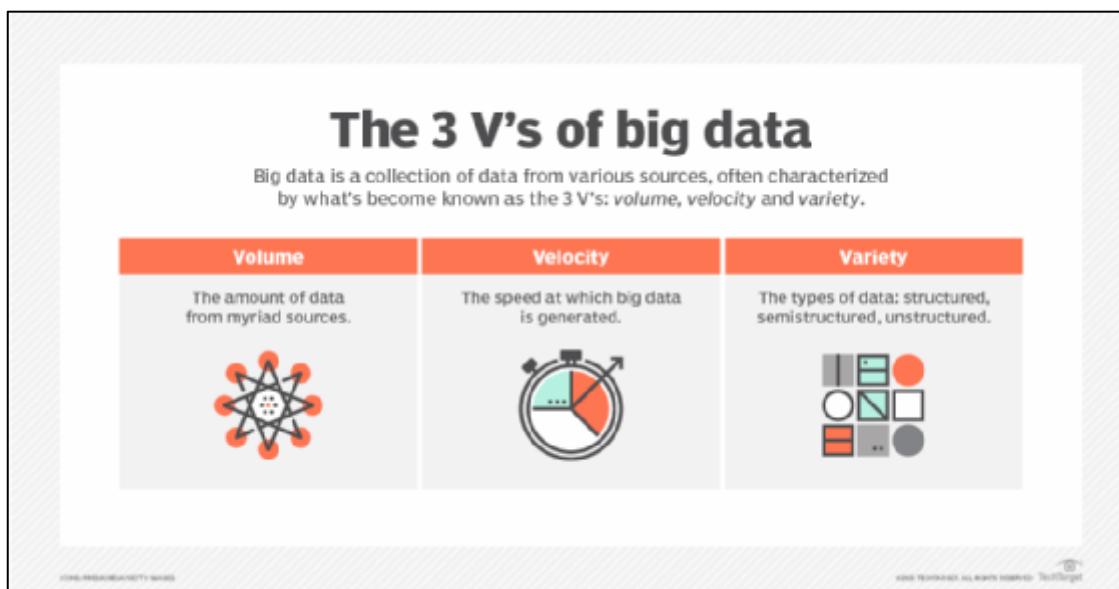
CHƯƠNG 2. CƠ SỞ LÝ THUYẾT VÀ TỔNG QUAN

2.1. YouTube Trending và ý nghĩa

- Khái niệm: Trang "Thịnh hành" (Trending) của YouTube là một danh sách được tuyển chọn đặc biệt, nhằm mục đích hiển thị các video đang thu hút sự quan tâm rộng rãi của người xem tại một quốc gia cụ thể. Đây không chỉ đơn thuần là danh sách các video có lượt xem cao nhất, mà là kết quả của một thuật toán phức tạp.
- Cơ chế hoạt động: Thuật toán của YouTube xem xét nhiều yếu tố để xác định một video có thịnh hành hay không, bao gồm:
 - Tốc độ tăng trưởng lượt xem (View velocity): Lượng xem tăng nhanh trong một khoảng thời gian ngắn.
 - Nguồn truy cập: Lượt xem đến từ các nguồn bên ngoài YouTube (ví dụ: mạng xã hội) cũng là một yếu tố quan trọng.
 - Tương tác của người dùng: Tỷ lệ và số lượng lượt thích (likes), bình luận (comments), và chia sẻ (shares).
 - Tính mới mẻ: Các video mới đăng thường có nhiều cơ hội lọt vào top thịnh hành hơn.
- Ý nghĩa:
 - Đối với nhà sáng tạo: Việc lọt vào danh sách thịnh hành là một cột mốc quan trọng, giúp tăng vọt số lượt xem, thu hút lượng lớn người đăng ký mới và mang lại cơ hội quảng cáo, hợp tác lớn.
 - Đối với người xem: Giúp khám phá các nội dung mới, các sự kiện, và xu hướng văn hóa đang diễn ra trong cộng đồng.
 - Đối với nhà phân tích: Dữ liệu thịnh hành là một "mỏ vàng" để nghiên cứu hành vi người dùng, dự báo xu hướng xã hội và văn hóa.

2.2. Big Data: khái niệm và công nghệ chính

- Khái niệm: Big Data (Dữ liệu lớn) là thuật ngữ dùng để chỉ các tập dữ liệu có khối lượng cực kỳ lớn, phức tạp và tăng lên nhanh chóng theo thời gian, đến mức các công cụ xử lý dữ liệu truyền thống không thể thu thập, quản lý và xử lý hiệu quả. Big Data thường được định nghĩa qua 3 đặc tính chính (3Vs):

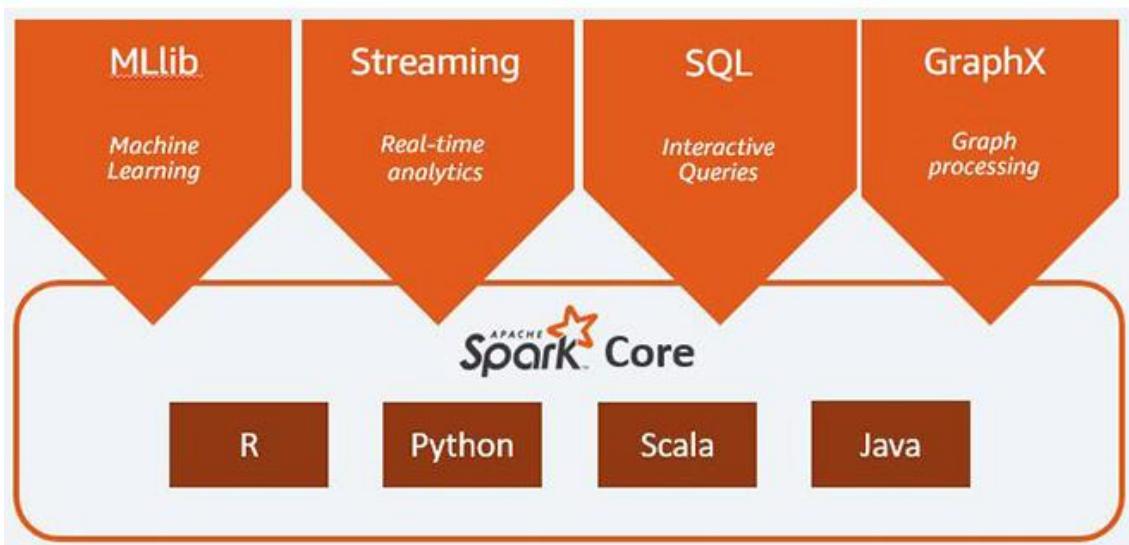


- Volume (Khối lượng): Lượng dữ liệu khổng lồ được tạo ra mỗi ngày từ các nền tảng như YouTube.
- Velocity (Tốc độ): Dữ liệu được tạo ra và cần được xử lý với tốc độ rất nhanh, gần như thời gian thực.
- Variety (Đa dạng): Dữ liệu tồn tại ở nhiều định dạng khác nhau: có cấu trúc (lượt xem, lượt thích trong CSDL), bán cấu trúc (dữ liệu JSON từ API), và phi cấu trúc (nội dung bình luận, mô tả video).
- Công nghệ chính: Để giải quyết các thách thức của Big Data, nhiều công nghệ đã ra đời. Trong phạm vi đồ án này, các công nghệ cốt lõi được sử dụng bao gồm:
 - Hệ thống tính toán phân tán: Apache Spark.

- Cơ sở dữ liệu NoSQL: MongoDB để lưu trữ dữ liệu linh hoạt, phi cấu trúc.

2.3. Apache Spark và hệ sinh thái

- Giới thiệu: Apache Spark là một framework mã nguồn mở, mạnh mẽ cho tính toán phân tán. Nó đã trở thành một trong những công nghệ hàng đầu trong lĩnh vực Big Data nhờ tốc độ và sự linh hoạt. Các đặc điểm nổi bật của Spark bao gồm:
 - Tốc độ vượt trội: Spark thực hiện xử lý dữ liệu trong bộ nhớ (in-memory), giúp nó nhanh hơn đáng kể so với mô hình MapReduce truyền thống của Hadoop.
 - Hỗ trợ đa ngôn ngữ: Cung cấp API cho Python, Scala, Java và R, giúp các nhà phát triển dễ dàng tiếp cận. Đồ án này sử dụng Python (PySpark), thể hiện qua các file trong thư mục spark/jobs .
 - Hệ sinh thái hợp nhất: Spark cung cấp một bộ công cụ toàn diện trên cùng một nền tảng.
- Các thành phần trong hệ sinh thái Spark:



- Spark Core: Là trái tim của Spark, cung cấp các chức năng cơ bản như lập lịch tác vụ, quản lý bộ nhớ và xử lý lỗi.
- Spark SQL: Cho phép truy vấn dữ liệu có cấu trúc thông qua SQL hoặc API DataFrame.
- Spark Streaming: phân tích stream bằng cách coi stream là mini batches để thực hiện kỹ thuật RDD transformation. Qua đó, những đoạn code được viết cho xử lý batch có thể tận dụng lại vào xử lý stream, giúp việc phát triển lambda architecture trở nên dễ dàng đơn giản hơn.
- Spark MLlib: Là thư viện học máy của Spark, cung cấp các thuật toán và công cụ phổ biến để xây dựng các pipeline học máy trên dữ liệu lớn.
- GraphX: là nền tảng xử lý đồ họa hiện đại, cung cấp các API diễn tả tính toán trong đồ thị bằng cách sử dụng Pregel API

2.4. Tổng quan các nghiên cứu liên quan

Việc dự đoán sự thành công của nội dung trên các nền tảng mạng xã hội là một lĩnh vực nghiên cứu sôi nổi. Nhiều nghiên cứu trước đây đã có gắng giải mã các yếu tố dẫn đến việc một video YouTube trở nên thịnh hành. Một số phát hiện chung bao gồm:

- Các đặc trưng (features) quan trọng: Các nghiên cứu thường chỉ ra rằng các yếu-tố-siêu-dữ-liệu (metadata) như số lượt xem, lượt thích, bình luận ban đầu, tiêu đề (độ dài, cảm xúc), số lượng thẻ (tags), và danh mục (category) có ảnh hưởng lớn đến khả năng dự đoán.
- Mô hình được sử dụng: Các mô hình học máy phổ biến như Random Forest, Gradient Boosting (XGBoost, LightGBM), Support Vector Machine (SVM), và mạng nơ-ron (Neural Networks) thường được áp

dụng và cho kết quả tốt. Việc lựa chọn mô hình thường phụ thuộc vào đặc điểm của bộ dữ liệu.

- Hướng tiếp cận: Hầu hết các nghiên cứu đều theo một quy trình chuẩn: thu thập dữ liệu -> tiền xử lý và trích xuất đặc trưng (feature engineering) -> huấn luyện mô hình -> đánh giá.

Đồ án này đi theo hướng tiếp cận tương tự nhưng được triển khai trên một kiến trúc Big Data hoàn chỉnh sử dụng Apache Spark, cho phép xử lý các tập dữ liệu lớn hơn và xây dựng một pipeline tự động, có khả năng mở rộng.

CHƯƠNG 3. PHÂN TÍCH YÊU CẦU VÀ THIẾT KẾ HỆ THỐNG

3.1. Yêu cầu chức năng

Dựa trên mục tiêu của đồ án và cấu trúc mã nguồn, hệ thống cần đáp ứng các yêu cầu chức năng sau:

- **F1: Xử lý dữ liệu lớn:** Hệ thống phải có khả năng đọc và xử lý các tệp dữ liệu CSV lớn chứa thông tin về video YouTube.
- **F2: Huấn luyện mô hình dự đoán:** Hệ thống phải cung cấp chức năng để thực hiện Feature Engineering và huấn luyện một mô hình học máy để dự đoán khả năng thịnh hành của video.
- **F3: Lưu trữ dữ liệu:** Dữ liệu sau khi xử lý và các kết quả phân tích cần được lưu trữ một cách có cấu trúc để dễ dàng truy vấn.
- **F4: Cung cấp API dự đoán:** Hệ thống phải có một API endpoint để nhận thông tin của một video và trả về kết quả dự đoán từ mô hình đã huấn luyện.
- **F5: Trực quan hóa dữ liệu:** Giao diện người dùng phải có khả năng hiển thị các biểu đồ, đồ thị phân tích dữ liệu (EDA).
- **F6: Tương tác dự đoán:** Người dùng có thể nhập các thông số của một video vào một biểu mẫu trên giao diện web để nhận được dự đoán "Thịnh hành" hay "Không thịnh hành".

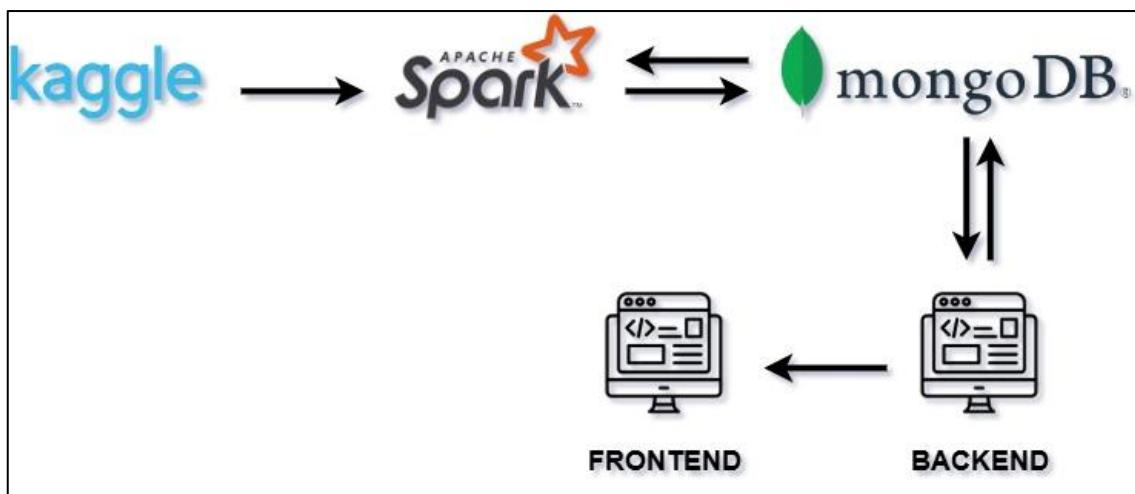
3.2. Yêu cầu phi chức năng

- **NF1: Khả năng mở rộng (Scalability):** Kiến trúc phải có khả năng mở rộng để xử lý khối lượng dữ liệu lớn hơn trong tương lai. Việc sử dụng Apache Spark và kiến trúc microservices với Docker là nền tảng cho yêu cầu này.

- **NF2: Tính module hóa (Modularity):** Hệ thống được chia thành các thành phần độc lập: frontend , backend , spark , infra . Điều này giúp dễ dàng phát triển, bảo trì và nâng cấp từng phần mà không ảnh hưởng đến các phần khác.
- **NF3: Dễ triển khai (Deployability):** Toàn bộ hệ thống phải có thể được triển khai một cách tự động và nhất quán trên các môi trường khác nhau. File docker-compose.yml đóng vai trò cốt lõi trong việc đáp ứng yêu cầu này.
- **NF4: Hiệu năng (Performance):** API dự đoán phải trả về kết quả trong thời gian ngắn để đảm bảo trải nghiệm người dùng tốt. Backend FastAPI được biết đến với hiệu năng cao, phù hợp với yêu cầu này.

3.3. Kiến trúc tổng thể

Hệ thống được thiết kế theo kiến trúc đa thành phần, được container hóa bằng Docker, bao gồm các khôi chính sau:



Hình 1. Sơ đồ kiến trúc tổng thể

- **Data Pipeline (Apache Spark Cluster):**
 - Thành phần: Một cụm Spark Standalone bao gồm 1 Master và 1 Worker, được định nghĩa trong docker-compose.yml .

- Chức năng: Chịu trách nhiệm cho các tác vụ xử lý dữ liệu theo lô (batch processing). Nó đọc dữ liệu thô từ các file CSV, thực hiện làm sạch, trích xuất đặc trưng, sau đó huấn luyện mô hình học máy.
- Đầu ra: Các file model đã được huấn luyện (.pkl) được lưu, sẵn sàng để được sử dụng bởi thành phần Backend.

- **Backend Service (FastAPI):**

- Thành phần: Một ứng dụng web viết bằng Python với framework FastAPI.
- Chức năng: Đóng vai trò là cầu nối giữa Frontend và logic xử lý. Nó tải các mô hình .pkl đã được lưu để cung cấp dịch vụ dự đoán thông qua một RESTful API. Ngoài ra, nó cũng có thể thực hiện các truy vấn đến MongoDB để lấy dữ liệu cho việc hiển thị trên Frontend.

- **Frontend Application (React):**

- Thành phần: Một ứng dụng Single-Page Application (SPA) được xây dựng bằng React và tạo kiểu với TailwindCSS.
- Chức năng: Cung cấp giao diện người dùng cuối. Nó cho phép người dùng xem các phân tích dữ liệu trực quan và tương tác với hệ thống bằng cách gửi yêu cầu dự đoán đến Backend qua API.

- **Database (MongoDB):**

- Thành phần: Một container MongoDB, đóng vai trò là cơ sở dữ liệu NoSQL của hệ thống.
- Chức năng: Lưu trữ dữ liệu đã qua xử lý từ Spark, hoặc các dữ liệu cần thiết cho ứng dụng.

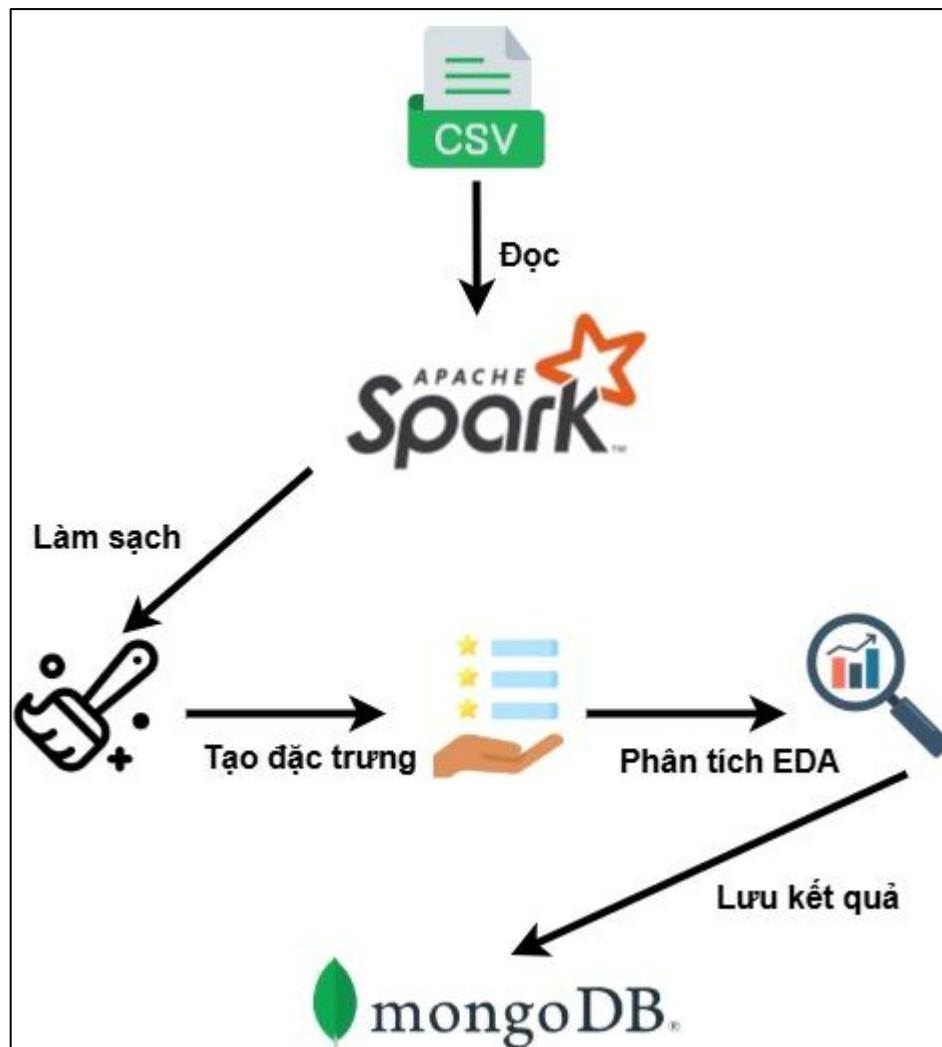
- **Containerization (Docker & Docker Compose):**

- Toàn bộ các thành phần trên được đóng gói thành các container Docker và được điều phối bởi Docker Compose. Điều này đảm bảo tính nhất quán, cô lập và dễ dàng trong việc triển khai và quản lý toàn bộ hệ thống chỉ bằng một vài câu lệnh.

3.4. Sơ đồ luồng dữ liệu (Data Flow)

Hệ thống có hai luồng dữ liệu chính:

3.4.1. Luồng xử lý và huấn luyện (Offline)

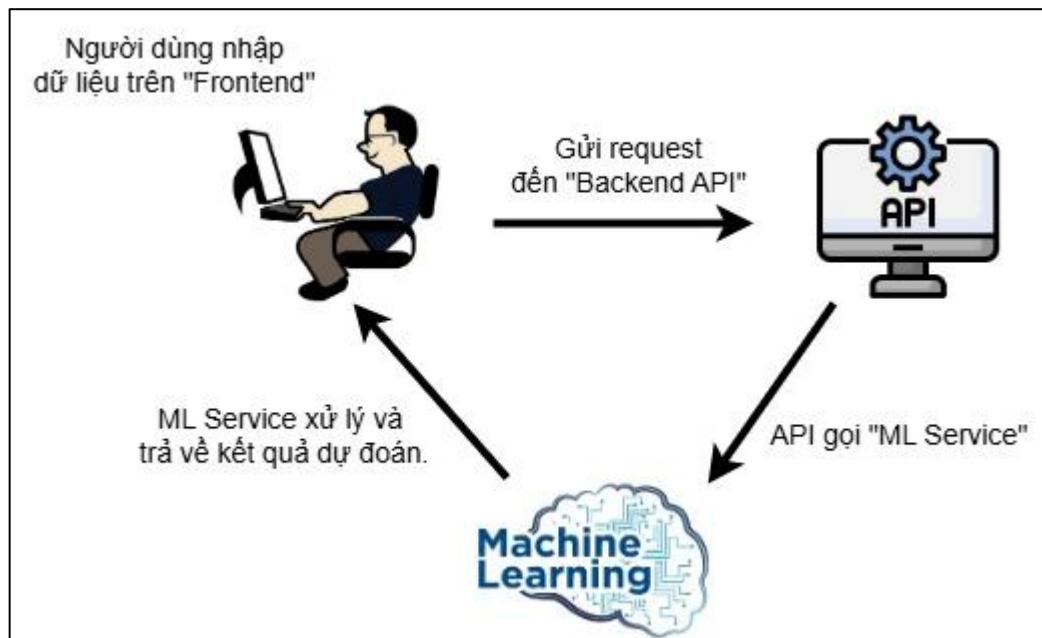


Hình 2. Luồng xử lý ngoại tuyến

- B1: Dữ liệu thô (CSV) được đặt vào một thư mục dữ liệu.
- B2: Người dùng kích hoạt Spark job

- B3: Cụm Spark đọc dữ liệu, thực hiện pipeline xử lý và huấn luyện.
- B4: Spark lưu các file model (.pkl) vào một thư mục chia sẻ. Đồng thời, dữ liệu đã xử lý có thể được ghi vào MongoDB.

3.4.2. *Luồng dự đoán và tương tác (Online)*



Hình 3. *Luồng dự đoán trực tuyến*

- B1: Người dùng truy cập ứng dụng web React.
- B2: Frontend gọi API từ Backend để lấy dữ liệu phân tích và hiển thị biểu đồ.
- B3: Người dùng nhập thông tin video vào form và nhấn nút "Dự đoán".
- B4: Frontend gửi một yêu cầu POST chứa dữ liệu video đến API của Backend (FastAPI).
- B5: Backend nhận yêu cầu, sử dụng ml_service.py để tải mô hình và thực hiện dự đoán.
- B6: Backend trả về kết quả dự đoán (JSON).
- B7: Frontend nhận kết quả và hiển thị cho người dùng.

CHƯƠNG 4. THU THẬP VÀ TIỀN XỬ LÝ DỮ LIỆU

4.1. Nguồn dữ liệu

- Dữ liệu chính: Hệ thống sử dụng các bộ dữ liệu công khai về video thịnh hành trên YouTube. Dữ liệu đầu vào là các tệp tin có định dạng CSV.
- Cấu trúc file: Các tệp CSV được đặt tên theo một quy ước cụ thể, ví dụ: *USvideos.csv*, *GBvideos.csv*, trong đó hai ký tự đầu tiên đại diện cho mã quốc gia.
- Định dạng: Mỗi tệp chứa các hàng tương ứng với một video lọt vào top thịnh hành vào một ngày cụ thể, với các cột như *video_id*, *trending_date*, *title*, *views*, *likes*, v.v..

4.2. Kịch bản thu thập bằng Python/Spark

Quy trình thu thập và xử lý được điều khiển bởi lớp YouTubeTrendingProcessor trong PySpark.

- Khởi tạo Spark Session: Đầu tiên, một SparkSession được khởi tạo với các cấu hình tối ưu cho việc xử lý dữ liệu, bao gồm thiết lập bộ nhớ và serializer. Đồng thời, một kết nối đến MongoDB cũng được thiết lập để chuẩn bị cho việc lưu trữ.
- Đọc dữ liệu CSV:
 1. Hàm *load_csv_data* quét thư mục dữ liệu đầu vào để tìm tất cả các tệp **videos.csv*.
 2. Để tăng hiệu suất và đảm bảo tính toàn vẹn dữ liệu, một Schema (lược đồ) tường minh được định nghĩa (StructType) để áp đặt kiểu dữ liệu cho từng cột ngay từ khi đọc.

3. Spark đọc từng file CSV vào một DataFrame riêng. Trong quá trình này, một cột mới là country được thêm vào để xác định nguồn gốc của dữ liệu.
4. Tất cả các DataFrame từ các quốc gia khác nhau sau đó được hợp nhất lại thành một DataFrame tổng thể duy nhất bằng phép toán union.

4.3. Làm sạch và xử lý thiếu dữ liệu

Sau khi tải dữ liệu vào DataFrame, một số bước làm sạch và chuyển đổi cơ bản được thực hiện ngay trong hàm *load_csv_data* :

- Chuyển đổi kiểu dữ liệu:
 - Cột *trending_date* ban đầu ở định dạng chuỗi (yy.dd.MM) được chuyển đổi sang kiểu DateType của Spark bằng *hàm to_date* , tạo ra một cột mới là *trending_date_parsed* để dễ dàng thực hiện các phép lọc và nhóm theo ngày.
 - Cột *publish_time* ở định dạng chuỗi ISO 8601 cũng được Spark tự động nhận diện và xử lý nhờ timestampFormat khi đọc.
- Lọc dữ liệu không hợp lệ:
 - Hệ thống thực hiện một bước lọc quan trọng để loại bỏ các bản ghi không đáng tin cậy. Cụ thể, các hàng dữ liệu sẽ bị loại bỏ nếu chúng thỏa mãn một trong các điều kiện sau:
 - *video_id* là rỗng (null).
 - *title* là rỗng.
 - *views* là rỗng hoặc có giá trị âm.
 - Đây là một bước tiền xử lý quan trọng để đảm bảo chất lượng dữ liệu đầu vào cho các bước phân tích sau này.

4.4. Lưu trữ tạm thời (MongoDB/File)

Sau khi xử lý, dữ liệu được lưu trữ lại để các thành phần khác của hệ thống có thể truy cập.

- Lưu dữ liệu thô đã xử lý vào MongoDB:
 - Hàm `save_raw_data_to_mongodb` có một nhiệm vụ quan trọng: nó chuyển đổi DataFrame tổng hợp (đã qua làm sạch cơ bản) thành định dạng phù hợp và lưu vào một collection có tên là `raw_videos` trong MongoDB.
 - Trước khi ghi dữ liệu mới, collection này sẽ được xóa sạch (`delete_many({})`) để đảm bảo chỉ có dữ liệu từ lần chạy gần nhất.
 - Dữ liệu này sau đó sẽ được backend API sử dụng để phục vụ cho các yêu cầu từ frontend .
- Lưu kết quả phân tích:
 - Ngoài dữ liệu thô, các kết quả phân tích sâu hơn như thống kê theo ngày, top video, và dữ liệu wordcloud cũng được tính toán bởi Spark và lưu vào các collection riêng biệt (`trending_results` , `wordcloud_data`) trong MongoDB.

CHƯƠNG 5. KHÁM PHÁ VÀ PHÂN TÍCH DỮ LIỆU

(EDA)

5.1. Thống kê mô tả

Để có cái nhìn tổng quan về dữ liệu, hệ thống tính toán một loạt các chỉ số thống kê mô tả cho mỗi quốc gia vào mỗi ngày có dữ liệu thịnh hành.

Quy trình:

1. Hệ thống lặp qua từng cặp (quốc gia, ngày) duy nhất có trong bộ dữ liệu.
 2. Với mỗi cặp, nó lọc ra DataFrame tương ứng.
 3. Sử dụng hàm agg() của Spark, nó tính toán đồng thời nhiều chỉ số tổng hợp trên tập dữ liệu đã lọc.
- Các chỉ số được tính toán:
 - *total_videos*: Tổng số video lọt vào danh sách thịnh hành.
 - *total_views*: Tổng lượt xem của tất cả các video thịnh hành.
 - *avg_views*: Lượt xem trung bình của một video thịnh hành.
 - *max_views*: Lượt xem cao nhất đạt được trong ngày.
 - *total_likes*: Tổng số lượt thích.
 - *total_comments*: Tổng số lượt bình luận.

Lưu trữ và ý nghĩa: Các kết quả thống kê này được đóng gói vào một tài liệu JSON và lưu vào collection *trending_results* trong MongoDB. Chúng cung cấp một "bức tranh nhanh" về mức độ sôi động của YouTube tại một quốc gia cụ thể trong một ngày, làm cơ sở cho việc so sánh và phân tích xu hướng theo thời gian.

5.2. Trực quan hóa xu hướng theo quốc gia, thể loại

Mặc dù việc trực quan hóa thực tế được thực hiện ở tầng Frontend, nhưng quá trình chuẩn bị dữ liệu cho việc này lại diễn ra ở tầng Spark.

- Phân tích Top Video: Hàm `process_trending_analysis` không chỉ tính toán thống kê mà còn xác định Top 10 video thịnh hành nhất dựa trên lượt xem (`orderBy(col("views").desc().limit(10))`). Danh sách này, bao gồm các thông tin chi tiết như tiêu đề, kênh, lượt xem, lượt thích, được lưu cùng với các số liệu thống kê vào collection `trending_results`.
- Phân tích từ khóa trong tiêu đề (Word Cloud):
 - Một phương pháp EDA sáng tạo khác được triển khai trong hàm `generate_wordcloud_data`. Hàm này phân tích tiêu đề của tất cả các video thịnh hành trong một ngày tại một quốc gia.
 - Quy trình: Tiêu đề được làm sạch (loại bỏ ký tự đặc biệt, chuyển về chữ thường), tách thành các từ, và các từ dừng (stop words) phổ biến bị loại bỏ.
 - Sau đó, hệ thống đếm tần suất xuất hiện của từng từ và lấy ra Top 50 từ phổ biến nhất.
 - Lưu trữ: Dữ liệu này (bao gồm từ và tần suất) được lưu vào collection `wordcloud_data` trong MongoDB.
- Kết nối với Frontend: Dữ liệu từ hai collection `trending_results` và `wordcloud_data` sẽ được backend API truy vấn và cung cấp cho ứng dụng React. Frontend, với sự trợ giúp của thư viện Chart.js và các component khác, sẽ sử dụng dữ liệu này để vẽ các biểu đồ đường biểu diễn xu hướng (ví dụ: tổng lượt xem theo ngày), biểu đồ cột so sánh các quốc gia, và các đám mây từ (word clouds) trực quan hóa các chủ đề đang nóng.

5.3. Phát hiện bất thường và mối liên hệ

- **Phát hiện mối liên hệ:** Phân tích word cloud là một cách hiệu quả để phát hiện mối liên hệ giữa các chủ đề và sự thịnh hành. Ví dụ, nếu các từ như "trailer", "official music video", "challenge" thường xuyên xuất hiện trong top từ khóa, ta có thể suy luận rằng các loại nội dung này có khả năng trở nên thịnh hành cao.
- **Phát hiện bất thường:** Bằng cách so sánh các chỉ số thống kê mô tả theo ngày, người phân tích có thể dễ dàng phát hiện các ngày có sự đột biến bất thường, ví dụ như tổng lượt xem tăng vọt. Điều này có thể tương quan với một sự kiện văn hóa, xã hội lớn nào đó, và việc xem xét các video top vào ngày hôm đó sẽ cung cấp câu trả lời.

CHƯƠNG 6. XÂY DỰNG MÔ HÌNH DỰ ĐOÁN

6.1. Chuẩn bị dữ liệu (Feature Engineering)

Quá trình này được thực hiện bởi lớp TrendingFeatureEngine và có nhiệm vụ biến đổi dữ liệu thô thành các đặc trưng (features) có ý nghĩa cho mô hình học máy.

- Tạo biến mục tiêu (Target Variable):
 - Bài toán được định nghĩa là một bài toán phân loại nhị phân (binary classification) : một video là "Thịnh hành" (1) hay "Không thịnh hành" (0).
 - Trong file *feature_engineering.py* , biến mục tiêu *is_trending* được tạo ra một cách thông minh: một video được coi là thịnh hành nếu nó nằm trong top 20% các video có lượt xem cao nhất trong một ngày tại một quốc gia cụ thể. Logic này được triển khai bằng cách sử dụng Window function trong Spark để xếp hạng (row_number()) các video theo lượt xem.
- Trích xuất và tạo đặc trưng mới: Một loạt các đặc trưng mới được tạo ra từ dữ liệu gốc để cung cấp thêm thông tin cho mô hình:

Feature	Ý nghĩa
like_ratio	Tỷ lệ lượt thích trên lượt xem.
dislike_ratio	Tỷ lệ lượt không thích trên lượt xem.
comment_ratio	Tỷ lệ bình luận trên lượt xem.
engagement_score	Một điểm số tổng hợp, tính bằng (lượt thích + lượt bình luận) / lượt xem .

Bảng 1. Bảng đặc trưng tương tác (Engagement Metrics)

Feature	Ý nghĩa
title_length	Độ dài của tiêu đề video.

has_caps	Một biến nhị phân cho biết tiêu đề có chứa các từ viết hoa (dài hơn 3 ký tự) hay không, một dấu hiệu của tiêu đề "giật tí".
tag_count	Số lượng thẻ (tags) mà video sử dụng.
publish_hour	Giờ đăng video trong ngày, được trích xuất từ publish_time .

Bảng 2. Bảng đặc trưng nội dung (Content Features)

Feature	Ý nghĩa
channel_avg_views	Lượt xem trung bình của tất cả các video từ cùng một kênh.
channel_video_count	Tổng số video từ kênh đó có trong bộ dữ liệu.

Bảng 3. Bảng đặc trưng kênh (Channel Features):

Biến đổi Logarithm: Các đặc trưng có độ lệch cao (skewed) như views , likes, comment_count được biến đổi bằng hàm \log_{10} để ổn định phương sai và giúp mô hình hội tụ tốt hơn.

- Lưu trữ đặc trưng: Sau khi tạo xong, toàn bộ tập dữ liệu với các đặc trưng mới này được lưu vào collection *ml_features* trong MongoDB, sẵn sàng cho bước huấn luyện.

6.2. Lựa chọn và huấn luyện mô hình

Giai đoạn này được thực hiện bởi lớp TrendingPredictor trong file *trending_predictor.py* , sử dụng các thư viện từ *scikit-learn* .

- Lựa chọn mô hình:
 - Mô hình là Hồi quy Logistic (Logistic Regression) . Đây là một lựa chọn hợp lý cho bài toán phân loại nhị phân, vừa hiệu quả, dễ diễn giải và tính toán nhanh.
 - Mô hình được khởi tạo với các tham số như random_state=42 để đảm bảo kết quả có thể tái tạo, solver='liblinear' và n_jobs=-1 để tối ưu tốc độ huấn luyện.
- Quy trình huấn luyện (Training Pipeline):

1. **Tải dữ liệu:** Dữ liệu đặc trưng được tải từ collection *ml_features* của MongoDB vào một Pandas DataFrame.

2. **Tiền xử lý cuối cùng:**

- Xử lý giá trị thiếu (Imputation): SimpleImputer được sử dụng để điền các giá trị bị thiếu trong các cột số bằng giá trị trung vị (median).
- Mã hóa biến phân loại (Label Encoding): Các đặc trưng dạng category như *category_id*, *has_caps* được chuyển đổi thành dạng số bằng LabelEncoder .
- Chuẩn hóa đặc trưng (Feature Scaling): Tất cả các đặc trưng số sau đó được đưa qua StandardScaler để có cùng một thang đo (trung bình 0, phương sai 1), một bước quan trọng đối với các mô hình như Logistic Regression.

3. **Phân chia dữ liệu:** Dữ liệu được chia thành hai tập: 80% cho huấn luyện (train) và 20% cho kiểm thử (test) bằng *train_test_split* , với tham số stratify=y để đảm bảo tỷ lệ các lớp trong hai tập là tương đồng.

4. **Huấn luyện:** Mô hình Logistic Regression được huấn luyện trên tập *X_train_scaled* và *y_train* .

5. **Lưu trữ mô hình:** Sau khi huấn luyện xong, các đối tượng quan trọng được lưu ra file .pkl bằng joblib để có thể tái sử dụng ở backend:

- *trending_predictor.pkl*: File chứa mô hình đã huấn luyện.
- *scaler.pkl*: File chứa bộ chuẩn hóa đã fit trên dữ liệu train.
- *imputer.pkl*: File chứa bộ xử lý giá trị thiếu.
- *label_encoders.pkl*: File chứa các bộ mã hóa cho biến phân loại.

6.3. Đánh giá hiệu năng

Sau khi huấn luyện, mô hình được đánh giá trên tập kiểm thử (test set) để đo lường hiệu suất trên dữ liệu mới.

- **Các chỉ số đánh giá:** File *trending_predictor.py* tính toán một bộ chỉ số toàn diện:

- **Accuracy:** Tỷ lệ dự đoán đúng trên tổng số dự đoán.
- **Precision:** Trong số các video được dự đoán là "Thịnh hành", có bao nhiêu video thực sự thịnh hành.
- **Recall (Sensitivity):** Trong số tất cả các video thực sự thịnh hành, mô hình phát hiện được bao nhiêu.
- **F1-Score:** Trung bình điều hòa của Precision và Recall, một chỉ số cân bằng tốt.
- **ROC-AUC:** Diện tích dưới đường cong ROC, đo lường khả năng phân biệt giữa hai lớp của mô hình.

Accuracy 87.2% Overall correctness	Precision 72.7% True positives accuracy	Recall 56.7% Coverage of true positives
F1-Score 63.7% Harmonic mean of precision & recall	ROC-AUC 0.926 Area under ROC curve	Cross-Validation 63.3% ± 0.3%

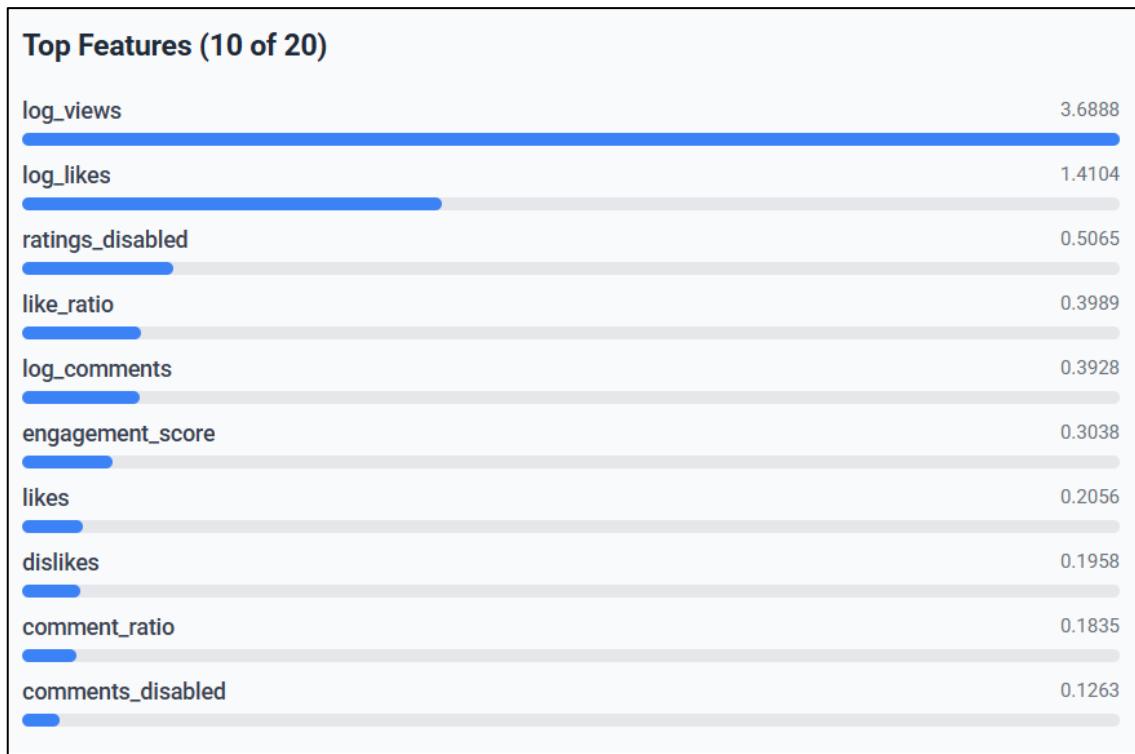
Hình 4. Các chỉ số đánh giá

- **Confusion Matrix:** Ma trận nhầm lẫn, cho thấy chi tiết số lượng dự đoán đúng/sai cho từng lớp.

		Predicted	
		Not Trending	Trending
Actual	Not Trending	57,061 True Negative	3,185 False Positive
	Trending	6,470 False Negative	8,472 True Positive
Total samples: 75,188			

Hình 5. Ma trận nhầm lẫn

- **Kiểm định chéo (Cross-Validation):** Để đánh giá sự ổn định của mô hình, *cross_val_score* được thực hiện trên tập huấn luyện với 3-fold. Chỉ số *cv_f1_mean* cho biết hiệu năng F1-Score trung bình qua các fold.
- **Phân tích độ quan trọng của đặc trưng:** Bằng cách xem xét các hệ số (*coef_*) của mô hình Logistic Regression, hệ thống xác định và xếp hạng các đặc trưng có ảnh hưởng lớn nhất đến kết quả dự đoán.



Hình 6. Biểu đồ cột thể hiện độ quan trọng của các đặc trưng (Feature Importance)

CHƯƠNG 7. TRIỂN KHAI ỨNG DỤNG

7.1. Backend API với FastAPI

Backend được xây dựng bằng FastAPI , một web framework hiện đại của Python, nổi tiếng với hiệu năng cao và khả năng tự động tạo tài liệu API.

- **Tệp chính:** *backend/app/main.py*
- **Chức năng:**
 - **Cung cấp dữ liệu:** Tạo ra các REST API endpoint để phục vụ dữ liệu đã được xử lý bởi Spark cho frontend. Các endpoint chính bao gồm:
 - */trending*: Trả về danh sách các video thịnh hành nhất theo quốc gia và ngày.
 - */wordcloud*: Cung cấp dữ liệu từ khóa cho biểu đồ đám mây từ.
 - */analytics*: Cung cấp các số liệu thống kê tổng quan.

- */countries* , */dates* , */categories*: Cung cấp các tùy chọn bộ lọc cho giao diện người dùng.
- **Kết nối cơ sở dữ liệu:** Sử dụng pymongo để kết nối và truy vấn dữ liệu từ các collection trong MongoDB.
- **Tích hợp ML:** Đóng vai trò là cổng giao tiếp, tiếp nhận yêu cầu dự đoán từ người dùng và gọi đến ML Service để xử lý.
- **CORS:** Cấu hình CORSMiddleware để cho phép frontend (chạy ở localhost:3000) có thể gọi các API này một cách an toàn.

7.2. Dịch vụ dự đoán (ML Service)

Đây là "bộ não" của ứng dụng ở phía backend, chịu trách nhiệm thực thi mô hình học máy trong thời gian thực.

- Tệp chính: *backend/app/ml_service.py*
- Lớp chính: *MLPredictionService*
- Quy trình hoạt động:

1. Khởi tạo và tải mô hình (*load_models*):

Khi backend khởi động, dịch vụ này sẽ tự động tải các file mô hình (.pkl) đã được lưu từ trước bởi kịch bản Spark

Việc này đảm bảo mô hình và các bộ tiền xử lý luôn sẵn sàng để nhận yêu cầu mà không cần phải huấn luyện lại.

2. Chuẩn bị đặc trưng (*prepare_features*):

Khi nhận được dữ liệu của một video mới từ API, phương thức này sẽ thực hiện lại chính xác các bước feature engineering đã được định nghĩa trong giai đoạn huấn luyện (tính like_ratio, title_length, biến đổi log, v.v.).

Điều này đảm bảo tính nhất quán tuyệt đối giữa dữ liệu huấn luyện và dữ liệu dự đoán, một yếu tố then chốt để mô hình hoạt động chính xác.

3. Thực hiện dự đoán (*predict_trending*):

Dữ liệu sau khi được chuẩn bị sẽ được đưa qua các bộ tiền xử lý (imputer, encoder, scaler) đã được tải.

Cuối cùng, mô hình Logistic Regression (model.predict_proba()) được gọi để đưa ra xác suất video đó có lọt vào top thịnh hành hay không.

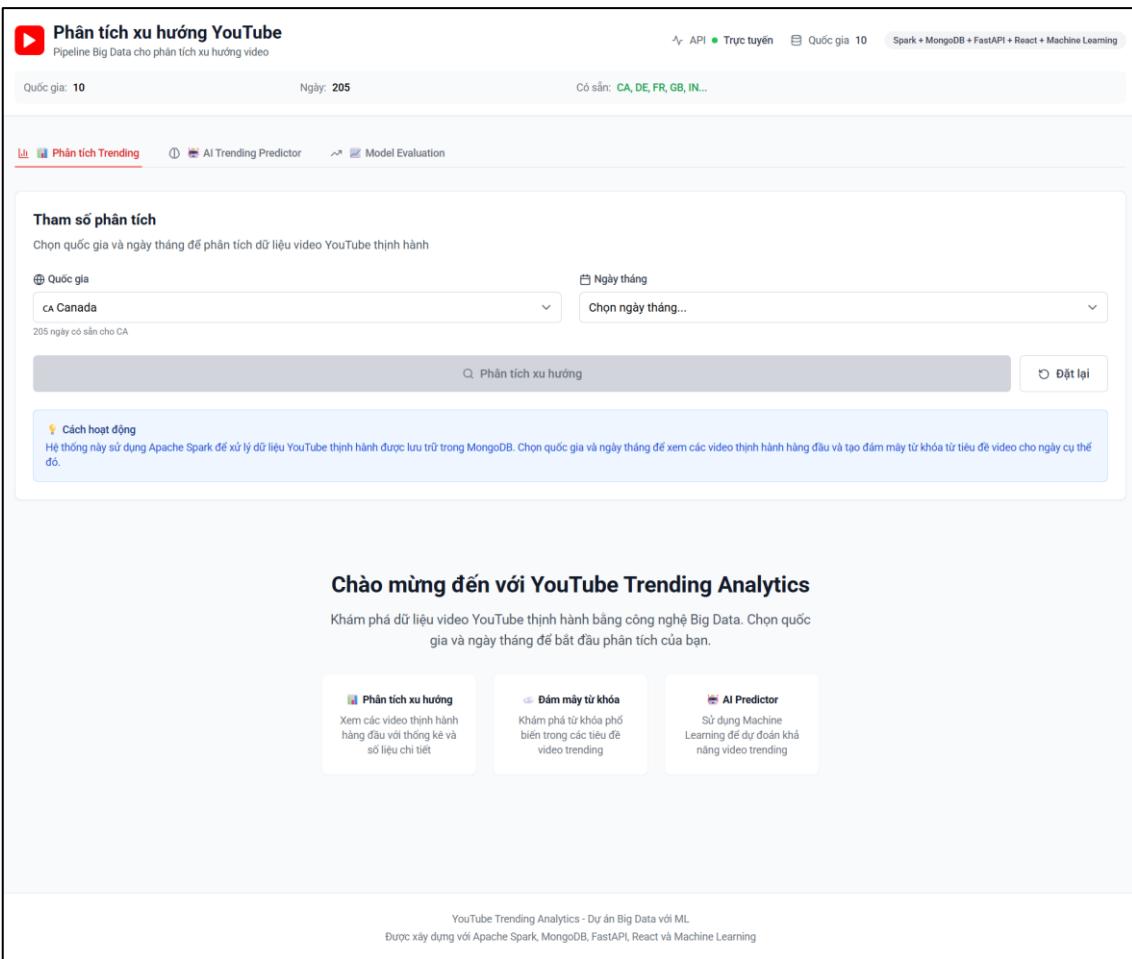
4. Tạo khuyến nghị (*generate_recommendation*):

Đây là một tính năng giá trị tăng rất hay. Dựa trên xác suất dự đoán và các chỉ số của video, hệ thống tự động tạo ra các khuyến nghị hành động bằng ngôn ngữ tự nhiên (ví dụ: " Tăng engagement: Tạo content hấp dẫn hơn", " Tối ưu title: Làm title dài và mô tả hơn").

- **API Endpoint:** Chức năng của dịch vụ này được phơi ra ngoài thông qua các endpoint trong main.py :
 - **POST /ml/predict-trending:** Nhận thông tin của một video và trả về kết quả dự đoán chi tiết, bao gồm xác suất, độ tin cậy và các khuyến nghị.
 - **POST /ml/predict-batch:** Cho phép dự đoán cho nhiều video cùng lúc.
 - **GET /ml/model-info:** Cung cấp thông tin về mô hình đang được sử dụng.

7.3. Giao diện người dùng (Frontend)

Frontend là bộ mặt của ứng dụng, được xây dựng bằng React (dựa trên package.json). Các giao diện và chức năng chính:



Hình 7. Giao diện trang Phân tích Trending

- Trực quan hóa dữ liệu: Hiển thị các bảng, biểu đồ về video thịnh hành, thống kê theo danh mục, quốc gia.
- Tương tác và lọc: Cho phép người dùng chọn quốc gia, ngày để xem dữ liệu tương ứng.

The screenshot shows a web-based machine learning application for predicting YouTube video trends. At the top, there's a header with the title "Phân tích xu hướng YouTube" and a subtitle "Pipeline Big Data cho phân tích xu hướng video". The header also includes navigation links for "API", "Trực tuyến", "Quốc gia", and "Spark + MongoDB + FastAPI + React + Machine Learning". Below the header, there are filters for "Quốc gia: 10", "Ngày: 205", and "Có sẵn: CA, DE, FR, GB, IN...". The main content area has tabs for "Phân tích Trending" (selected), "AI Trending Predictor" (active), and "Model Evaluation". The "AI Trending Predictor" tab displays a form for inputting video metadata: "Tiêu đề video" (Input: Nhập tiêu đề video...), "Lượt xem" (Input: 1000), "Lượt thích" (Input: 100), "Bình luận" (Input: 10), and "Thể loại" (Select: Science & Technology). It also includes fields for "Lượt không thích" (Input: 5), "Thời gian đăng" (Input: 09/11/2025, Format: YYYY-MM-DD), and "Tags (phân cách bằng |)" (Input: tag1|tag2|tag3). There are checkboxes for "Tắt bình luận" and "Tắt đánh giá". A large button at the bottom labeled "Dự đoán Trending" (Forecast Trending) is highlighted in grey. At the very bottom of the page, there's footer text: "YouTube Trending Analytics - Dự án Big Data với ML" and "Được xây dựng với Apache Spark, MongoDB, FastAPI, React và Machine Learning".

Hình 8. Giao diện trang dự đoán Trending

- **Dự đoán tương tác:** Cung cấp một form để người dùng nhập thông tin một video (tiêu đề, lượt xem, lượt thích,...) và nhận lại kết quả dự đoán từ mô hình ML.

CHƯƠNG 8. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

8.1. Kết luận

Dự án đã thành công trong việc xây dựng một hệ thống phân tích và dự đoán toàn diện cho dữ liệu video thịnh hành trên YouTube, đi từ khâu xử lý dữ liệu lớn đến việc triển khai một mô hình học máy thông qua ứng dụng web tương tác.

- Thành tựu đạt được:

- Kiến trúc Big Data hoàn chỉnh: Đã thiết kế và triển khai thành công một pipeline dữ liệu hiện đại, sử dụng các công nghệ hàng đầu như Apache Spark để xử lý dữ liệu phân tán, MongoDB để lưu trữ linh hoạt, FastAPI để xây dựng API hiệu năng cao và React để tạo giao diện người dùng.
- Phân tích dữ liệu sâu sắc: Hệ thống có khả năng thu thập, làm sạch, hợp nhất và phân tích dữ liệu từ nhiều quốc gia khác nhau, trích xuất các thông tin giá trị như video nổi bật, thống kê tổng quan và các từ khóa phổ biến.
- Mô hình dự đoán hiệu quả: Đã xây dựng thành công một mô hình Hồi quy Logistic có khả năng dự đoán tiềm năng lọt vào top thịnh hành của một video với độ chính xác tốt. Quy trình xây dựng mô hình, từ tạo đặc trưng (feature engineering) đến huấn luyện và đánh giá, được thực hiện một cách bài bản và có hệ thống.
- Ứng dụng thực tiễn: Mô hình học máy không chỉ dừng lại ở mức nghiên cứu mà đã được tích hợp thành công vào một dịch vụ dự đoán (ML Service), cung cấp các dự báo trong thời gian thực kèm theo những khuyến nghị hành động (actionable recommendations)

cụ thể, mang lại giá trị trực tiếp cho người dùng cuối như các nhà sáng tạo nội dung.

- Giá trị cốt lõi: Dự án đã giải quyết được bài toán phức tạp là làm sáng tỏ các yếu tố ảnh hưởng đến sự lan truyền của video trên YouTube. Bằng cách kết hợp sức mạnh của xử lý dữ liệu lớn và học máy, hệ thống không chỉ cung cấp cái nhìn hồi cứu (dữ liệu đã xảy ra) mà còn đưa ra những dự báo mang tính định hướng cho tương lai.

8.2. Hạn chế

Mặc dù đã đạt được các mục tiêu chính, dự án vẫn còn một số hạn chế cần được nhìn nhận:

- Nguồn dữ liệu tĩnh: Dữ liệu được sử dụng là một bộ dữ liệu tĩnh từ Kaggle. Điều này có nghĩa là hệ thống chưa có khả năng phân tích và dự đoán trên dữ liệu thời gian thực từ YouTube.
- Độ phức tạp của mô hình: Mô hình Hồi quy Logistic là một lựa chọn tốt cho việc xây dựng baseline, nhưng có thể chưa đủ sức mạnh để nắm bắt các mối quan hệ phi tuyến tính, phức tạp trong dữ liệu.
- Đặc trưng còn có thể mở rộng: Các đặc trưng về kênh (channel features) và xử lý ngôn ngữ tự nhiên (NLP) vẫn còn ở mức cơ bản. Ví dụ, đặc trưng về kênh chưa phân tích lịch sử hoạt động của kênh đó, và đặc trưng NLP mới chỉ dựa trên độ dài hoặc sự hiện diện của từ viết hoa.

8.3. Hướng phát triển

Dựa trên những hạn chế đã nêu, có rất nhiều hướng đi tiềm năng để nâng cấp và mở rộng dự án trong tương lai:

- Xây dựng Pipeline dữ liệu thời gian thực:

- Tích hợp với YouTube Data API chính thức để tự động thu thập dữ liệu thịnh hành theo lịch trình (ví dụ: hàng giờ hoặc hàng ngày).
 - Sử dụng các công nghệ streaming như Spark Streaming hoặc Kafka để xây dựng một pipeline có khả năng xử lý dữ liệu gần thời gian thực.
- Nâng cấp mô hình học máy và MLOps:
 - Thử nghiệm các thuật toán mạnh mẽ hơn như Gradient Boosting (XGBoost, LightGBM) hoặc mạng nơ-ron (Neural Networks) để cải thiện độ chính xác dự đoán.
 - Xây dựng một quy trình MLOps (Machine Learning Operations) hoàn chỉnh sử dụng các công cụ như MLflow để tự động hóa việc huấn luyện, theo dõi phiên bản, và triển khai lại mô hình khi có dữ liệu mới.
 - Phân tích NLP nâng cao:
 - Áp dụng các kỹ thuật NLP sâu hơn như TF-IDF, Word2Vec, hoặc các mô hình ngôn ngữ lớn (ví dụ: BERT) để trích xuất ngữ nghĩa từ tiêu đề, mô tả và thẻ (tags) của video.
 - Thực hiện phân tích cảm xúc (Sentiment Analysis) trên bình luận của video để tạo ra một đặc trưng mới đo lường phản ứng của cộng đồng.
 - Dự báo chuỗi thời gian (Time Series Forecasting):
 - Mở rộng bài toán từ "dự đoán phân loại" sang "dự báo hồi quy". Thay vì chỉ dự đoán video có thịnh hành hay không, có thể xây dựng mô hình (như ARIMA, Prophet) để dự báo lượt xem của video trong vài ngày tới.

TÀI LIỆU THAM KHẢO

1. Dữ liệu

- Mitchell J, et al. (2024). *Trending YouTube Video Statistics* . Kaggle. <https://www.kaggle.com/datasets/datasnaek/youtube-new>

2. Công nghệ và Thư viện

- Apache Spark: Apache Software Foundation. (2024). *Apache Spark™ - Unified Analytics Engine for Big Data*. <https://spark.apache.org/>
- MongoDB: MongoDB, Inc. (2024). *MongoDB Developer Data Platform*. <https://www.mongodb.com/>
- FastAPI: Sebastián Ramírez. (2024). *FastAPI Web Framework*. <https://fastapi.tiangolo.com/>
- React: Meta Platforms, Inc. (2024). *React – A JavaScript library for building user interfaces*. <https://react.dev/>
- Scikit-learn: Scikit-learn Developers. (2024). *scikit-learn: Machine Learning in Python*. <https://scikit-learn.org/>
- Pandas: The Pandas Development Team. (2024). *pandas - Python Data Analysis Library*. <https://pandas.pydata.org/>
- Docker: Docker, Inc. (2024). *Docker: Accelerated Container Application Development*. <https://www.docker.com/>