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

ỨNG DỤNG PHÂN TÍCH VÀ DỰ ĐOÁN XU HƯỚNG VIDEO 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

ỨNG DỤNG PHÂN TÍCH VÀ DỰ ĐOÁN XU HƯỚNG VIDEO 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

BẢNG PHÂN CÔNG CÔNG VIỆC

MSSV	Tên	Công việc	Đánh giá
2001222641	Trần Công Minh	• 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. • Word + PowerPoint	100%
2001225676	Lê Đức Trung	• 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.	100%

MỤC LỤC

MỞ ĐẦU 1

CHƯƠNG 1. CƠ SỞ LÝ THUYẾT VÀ TỔNG QUAN 5

1.1. YouTube Trending: Khái niệm, cơ chế hoạt động và ý nghĩa 5

1.2. Big Data: Định nghĩa và các thách thức 6

1.3. Tổng quan về các công nghệ được sử dụng 8

1.3.1. Hệ thống tệp phân tán HDFS 8

1.3.2. Apache Spark và hệ sinh thái 9

1.3.3. Cơ sở dữ liệu NoSQL – MongoDB 10

1.3.4. Kiến trúc Web (FastAPI, React) 11

1.4. Tổng quan các nghiên cứu liên quan 12

CHƯƠNG 2. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 13

2.1. Phân tích yêu cầu 13

2.1.1. Yêu cầu chức năng (Functional Requirements) 13

2.1.2. Yêu cầu phi chức năng (Non-functional Requirements) 14

2.2. Kiến trúc tổng thể của hệ thống 15

2.3. Luồng dữ liệu (Data Flow) 16

2.3.1. Luồng xử lý và huấn luyện (Offline) 16

2.3.2. Luồng dự đoán và tương tác (Online) 17

CHƯƠNG 3. THU THẬP VÀ TIỀN XỬ LÝ DỮ LIỆU 19

3.1. Nguồn và mô tả dữ liệu 19

3.2. Pipeline thu thập và xử lý dữ liệu với Apache Spark 20

3.2.1. Đọc và hợp nhất dữ liệu từ HDFS 21

3.2.2. Làm sạch và xử lý dữ liệu (Data Cleaning & Preprocessing) .. 21
3.3. Lưu trữ dữ liệu đã qua xử lý 22
3.3.1. Lưu trữ vào MongoDB cho ứng dụng web 22
3.3.2. Cấu trúc các collection trong MongoDB 22
CHƯƠNG 4. PHÂN TÍCH DỮ LIỆU KHÁM PHÁ (EDA) 24
4.1. Tổng quan các chỉ số chính 24
4.2. Trực quan hóa và Phân tích Chuyên sâu 25
4.2.1. Phân bố theo Danh mục (Category Distribution) 25
4.2.2. Phân tích từ khóa nổi bật qua Word Cloud 25
4.2.3. Mối quan hệ giữa Lượt xem và Tỷ lệ tương tác 27
CHƯƠNG 5. XÂY DỰNG MÔ HÌNH DỰ ĐOÁN 28
5.1. Kỹ thuật đặc trưng (Feature Engineering) 28
5.1.1. Tạo các đặc trưng về tương tác (Engagement Metrics) 28
5.1.2. Tạo các đặc trưng về nội dung (Content Features) 29
5.1.3. Vector hóa văn bản (Text Vectorization) 29
5.2. Lựa chọn và huấn luyện mô hình 30
5.2.1. Mô hình phân cụm nội dung (K-Means) 30
5.2.2. Mô hình dự đoán số ngày trên top thịnh hành (Random Forest Regression) 30
5.3. Đánh giá hiệu năng mô hình 31
CHƯƠNG 6. TRIỂN KHAI ỨNG DỤNG 34
6.1. Xây dựng Backend API với FastAPI 34
6.2. Xây dựng Giao diện người dùng (Frontend) với React 35

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 38

TÀI LIỆU THAM KHẢO 41

PHỤ LỤC 43

DANH MỤC HÌNH

Hình 1. Sơ đồ minh họa các yếu tố đầu vào cho thuật toán YouTube Trending 6

Hình 2. Mô hình 3Vs của Big Data..... 7

Hình 3. Kiến trúc HDFS..... 9

Hình 4. Sơ đồ minh họa hệ sinh thái Apache Spark 10

Hình 5. Sơ đồ kiến trúc tổng thể của hệ thống..... 15

Hình 6. Sơ đồ luồng xử lý và huấn luyện (Offline) 16

Hình 7. Sơ đồ luồng dự đoán và tương tác (Online)..... 17

Hình 8. Bộ dữ liệu Kaggle “Trending YouTube Video Statistics” 19

Hình 9. Biểu đồ "Top 5 Đặc Trưng Quan Trọng Nhất"..... 32

Hình 10. Màn hình trang Dashboard..... 35

Hình 11. Màn hình trang Phân tích Xu hướng..... 36

Hình 12. Màn hình trang Dự đoán 37

DANH MỤC BẢNG

Bảng 1. Mô tả các thuộc tính chính 20

Bảng 2. Mô tả cấu trúc các collection trong MongoDB 23

Bảng 3. Mô tả các đặc trưng về tương tác 28

Bảng 4. Mô tả các đặc trưng về nội dung 29

Bảng 5. Kết quả đánh giá mô hình hồi quy..... 32

MỞ ĐẦU

1. Giới thiệu

Trong kỷ nguyên số hiện nay, YouTube đã trở thành nền tảng chia sẻ video lớn nhất hành tinh, không chỉ là một phương tiện giải trí mà còn là một công cụ truyền thông mạnh mẽ, ảnh hưởng sâu sắc đến văn hóa, xã hội và kinh tế toàn cầu. Mỗi ngày, hàng tỷ giờ video được xem và hàng trăm triệu giờ nội dung được tải lên, tạo ra một kho dữ liệu khổng lồ (Big Data) chứa đựng những thông tin vô giá về hành vi, sở thích và xu hướng của người dùng.

Trong dòng chảy dữ liệu khổng lồ đó, mục "Thịnh hành" (Trending) của YouTube nổi lên như một chỉ báo quan trọng, phản ánh những nội dung đang thu hút sự quan tâm lớn nhất từ cộng đồng tại một quốc gia cụ thể. Việc một video lọt vào top thịnh hành không chỉ mang lại danh tiếng cho nhà sáng tạo mà còn mở ra nhiều cơ hội về thương mại và quảng cáo. Tuy nhiên, việc phân tích và dự đoán những yếu tố nào giúp một video trở nên thịnh hành là một bài toán phức tạp, đòi hỏi khả năng xử lý và phân tích dữ liệu ở quy mô lớn.

2. Lý do chọn đề tài

Sự bùng nổ của dữ liệu từ YouTube đặt ra một thách thức lớn nhưng cũng là cơ hội để ứng dụng các công nghệ Dữ liệu lớn (Big Data) vào việc khai phá tri thức. Việc phân tích dữ liệu video thịnh hành mang lại lợi ích cho nhiều đối tượng:

- **Đối với nhà sáng tạo nội dung:** Hiểu được các yếu tố then chốt (như tiêu đề, mô tả, thời điểm đăng, thể loại) giúp tối ưu hóa nội dung, tăng khả năng tiếp cận và cơ hội lọt vào top thịnh hành.
- **Đối với các nhà quảng cáo và tiếp thị:** Nắm bắt nhanh chóng các xu hướng mới giúp xây dựng các chiến dịch quảng cáo hiệu quả, nhắm đúng đối tượng và tối ưu hóa chi phí.
- **Đối với các nhà nghiên cứu:** Dữ liệu xu hướng là nguồn thông tin quý giá để nghiên cứu các hiện tượng xã hội, văn hóa và sự lan truyền thông tin trong cộng đồng mạng.

Xuất phát từ những nhu cầu thực tiễn đó, đề tài ***"Ứng dụng phân tích và dự đoán xu hướng video YouTube"*** được thực hiện nhằm xây dựng một hệ thống hoàn chỉnh, áp dụng các công nghệ xử lý dữ liệu phân tán để giải quyết bài toán này một cách hiệu quả.

3. Mục tiêu nghiên cứu

Đồ án tập trung vào việc thực hiện các mục tiêu chính sau đây:

- **Xây dựng hệ thống xử lý dữ liệu lớn:** Thiết kế và triển khai một hệ thống có khả năng thu thập, lưu trữ và xử lý song song tập dữ liệu lớn về các video thịnh hành trên YouTube bằng cách sử dụng các công nghệ phân tán hàng đầu như Hệ thống tệp phân tán Hadoop (HDFS) và Apache Spark.

- **Phân tích và khám phá dữ liệu:** Thực hiện phân tích dữ liệu khám phá (EDA) để tìm ra các đặc điểm, quy luật và mối tương quan ẩn sau dữ liệu, chẳng hạn như xu hướng theo thể loại, quốc gia, hoặc các yếu tố ảnh hưởng đến mức độ tương tác (lượt xem, thích, bình luận).
- **Xây dựng mô hình dự đoán:** Áp dụng các thuật toán học máy (Machine Learning) để xây dựng mô hình có khả năng dự đoán các chỉ số quan trọng, ví dụ như số ngày một video có thể duy trì trên tab thịnh hành, dựa trên các thuộc tính của nó.
- **Trực quan hóa kết quả:** Phát triển một ứng dụng web tương tác cho phép người dùng cuối dễ dàng truy cập, khám phá các kết quả phân tích và sử dụng mô hình dự đoán một cách trực quan thông qua biểu đồ, bảng biểu và giao diện thân thiện.

4. Phạm vi và đóng góp của đồ án

- **Phạm vi:**
  - **Dữ liệu:** Đồ án sử dụng bộ dữ liệu lịch sử về các video thịnh hành trên YouTube tại một số quốc gia (*ví dụ: Mỹ, Anh, Canada, Đức, Pháp,...*). Hệ thống không thực hiện thu thập và phân tích dữ liệu theo thời gian thực.
  - **Công nghệ:** Tập trung vào hệ sinh thái Hadoop/Spark để xử lý dữ liệu (batch processing) và FastAPI, React để xây dựng ứng dụng web.
  - **Phân tích:** Các phân tích và dự đoán được giới hạn trong phạm vi các thuộc tính có sẵn trong bộ dữ liệu.
- **Đóng góp của đồ án:**
  - **Về mặt kỹ thuật:** Xây dựng thành công một quy trình xử lý dữ liệu lớn end-to-end, từ khâu tiền xử lý, phân tích, huấn luyện mô hình đến triển khai thành một ứng dụng hoàn chỉnh. Đây là một

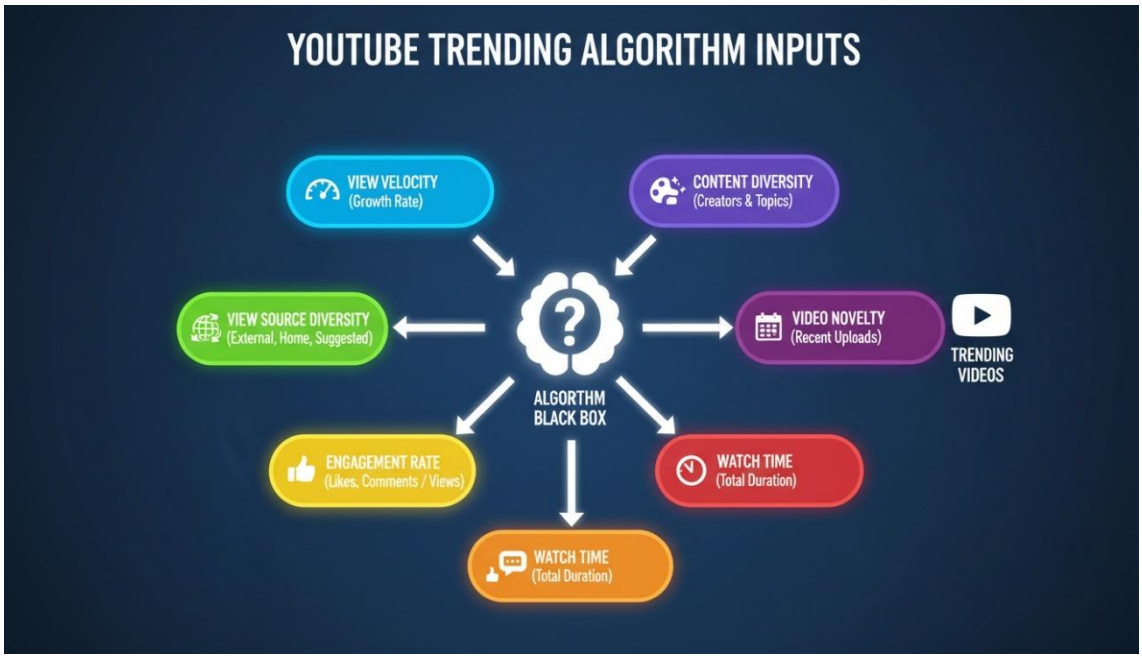
minh chứng thực tế về việc áp dụng kiến thức Big Data vào giải quyết một bài toán cụ thể.

- **Về mặt ứng dụng:** Cung cấp một công cụ hữu ích giúp người dùng có cái nhìn sâu sắc hơn về các xu hướng trên YouTube. Ứng dụng không chỉ hiển thị các thống kê mà còn đưa ra các dự đoán có giá trị, hỗ trợ việc ra quyết định cho các nhà sáng tạo nội dung và nhà tiếp thị.

CHƯƠNG 1. CƠ SỞ LÝ THUYẾT VÀ TỔNG QUAN

1.1. YouTube Trending: Khái niệm, cơ chế hoạt động và ý nghĩa

- **Khái niệm:** "YouTube Trending" (Thịnh hành) là một danh sách được cập nhật liên tục, hiển thị các video đang có sức lan tỏa và phổ biến nhất đối với người xem tại một quốc gia cụ thể. Đây không đơn thuần là danh sách các video có lượt xem cao nhất, mà là một tập hợp các video đang có tốc độ tăng trưởng lượt xem và tương tác mạnh mẽ, thu hút được sự quan tâm rộng rãi từ nhiều đối tượng khán giả.
- **Cơ chế hoạt động:** Thuật toán xác định video thịnh hành của YouTube là một "hộp đen" và không được công bố chi tiết. Tuy nhiên, dựa trên các quan sát và thông báo từ YouTube, các yếu tố chính ảnh hưởng đến việc một video có được xếp hạng hay không bao gồm:
    - **Tốc độ tăng trưởng lượt xem (View velocity):** Số lượt xem video tăng nhanh như thế nào trong một khoảng thời gian ngắn sau khi xuất bản.
    - **Nguồn gốc của lượt xem:** Lượt xem đến từ nhiều nguồn khác nhau (bên ngoài YouTube, trang chủ, video đề xuất) được đánh giá cao hơn.
    - **Tỷ lệ tương tác:** Tỷ lệ giữa lượt thích (likes), bình luận (comments) và lượt xem.
    - **Thời gian xem (Watch time):** Tổng thời gian mà người dùng đã dành để xem video.
    - **Tính mới của video (Novelty):** Các video mới xuất bản thường có nhiều cơ hội hơn.
    - **Sự đa dạng về nội dung:** Thuật toán cũng cố gắng cân bằng để hiển thị các video từ nhiều nhà sáng tạo và chủ đề khác nhau, tránh sự thống trị của một vài kênh lớn.

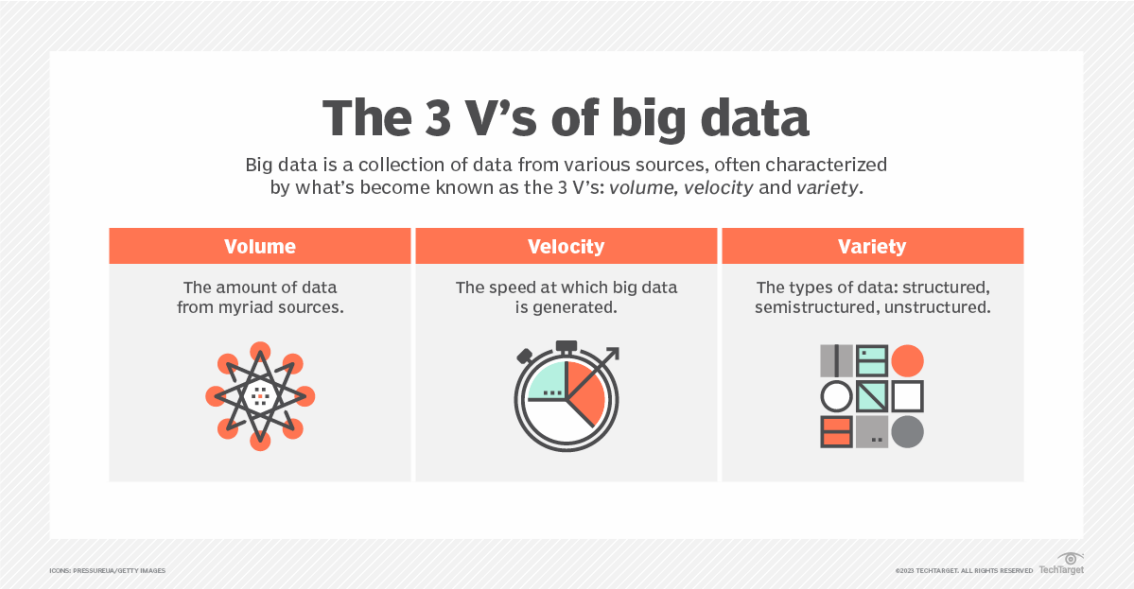


Hình 1. Sơ đồ minh họa các yếu tố đầu vào cho thuật toán YouTube Trending

- **Ý nghĩa:** Việc lọt vào danh sách thịnh hành có ý nghĩa vô cùng quan trọng. Nó giúp video và kênh được "tiếp xúc" với một lượng khán giả khổng lồ, vượt ra ngoài nhóm người đăng ký thông thường. Điều này không chỉ làm tăng vọt lượt xem, lượt đăng ký mà còn nâng cao uy tín và thương hiệu cho nhà sáng tạo, mở ra các cơ hội hợp tác quảng cáo và kinh doanh.

1.2. Big Data: Định nghĩa và các thách thức

- **Định nghĩa:** Big Data (Dữ liệu lớn) là thuật ngữ dùng để chỉ các tập dữ liệu có khối lượng cực kỳ lớn và phức tạp, đến mức các công cụ xử lý dữ liệu truyền thống không thể thu thập, quản lý và xử lý hiệu quả trong một khoảng thời gian hợp lý. Big Data thường được định nghĩa qua các đặc trưng, phổ biến nhất là mô hình 3Vs:



Hình 2. Mô hình 3Vs của Big Data

- **Volume (Khối lượng):** Đề cập đến quy mô của dữ liệu. Trong bối cảnh của đề tài, hàng triệu bản ghi về video thịnh hành từ nhiều quốc gia trong một khoảng thời gian dài tạo thành một tập dữ liệu có khối lượng lớn.
- **Velocity (Vận tốc):** Đề cập đến tốc độ dữ liệu được tạo ra và cần được xử lý. Dữ liệu trên các nền tảng mạng xã hội như YouTube được tạo ra liên tục với tốc độ chóng mặt.
- **Variety (Sự đa dạng):** Đề cập đến các loại hình dữ liệu khác nhau. Dữ liệu YouTube bao gồm cả dữ liệu có cấu trúc (lượt xem, lượt thích), bán cấu trúc (dữ liệu JSON từ API) và phi cấu trúc (tiêu đề, mô tả, bình luận).

- **Các thách thức:** Việc xử lý Big Data đặt ra nhiều thách thức, bao gồm:
  - **Lưu trữ:** Cần các hệ thống lưu trữ có khả năng mở rộng, chịu lỗi và chi phí hợp lý.
  - **Xử lý:** Cần các framework xử lý phân tán để có thể phân tích dữ liệu trong thời gian chấp nhận được.

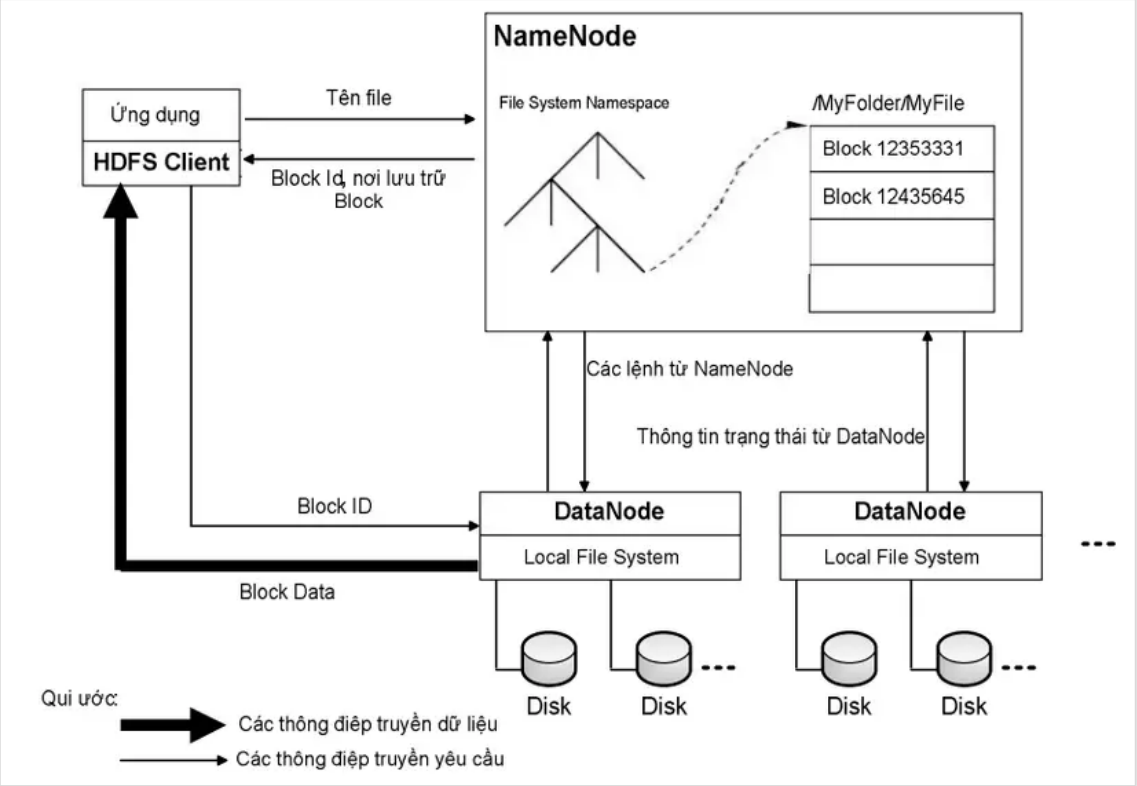
- **Đảm bảo chất lượng dữ liệu:** Dữ liệu lớn thường nhiễu, thiếu và không nhất quán, đòi hỏi các quy trình làm sạch và tiền xử lý phức tạp.
- **Bảo mật:** Đảm bảo an toàn cho dữ liệu nhạy cảm ở quy mô lớn.

1.3. Tổng quan về các công nghệ được sử dụng

1.3.1. Hệ thống tệp phân tán HDFS

Hadoop Distributed File System (HDFS) là một hệ thống tệp phân tán được thiết kế để chạy trên các phần cứng thông thường. HDFS cung cấp khả năng truy cập dữ liệu với thông lượng cao và là thành phần lưu trữ chính trong hệ sinh thái Hadoop.

- **Kiến trúc:** HDFS có kiến trúc master/slave, bao gồm một NameNode (máy chủ quản lý) và nhiều DataNode (máy chủ lưu trữ dữ liệu). NameNode chịu trách nhiệm quản lý siêu dữ liệu (metadata) của hệ thống tệp, trong khi DataNodes lưu trữ các khối dữ liệu thực tế (data blocks).

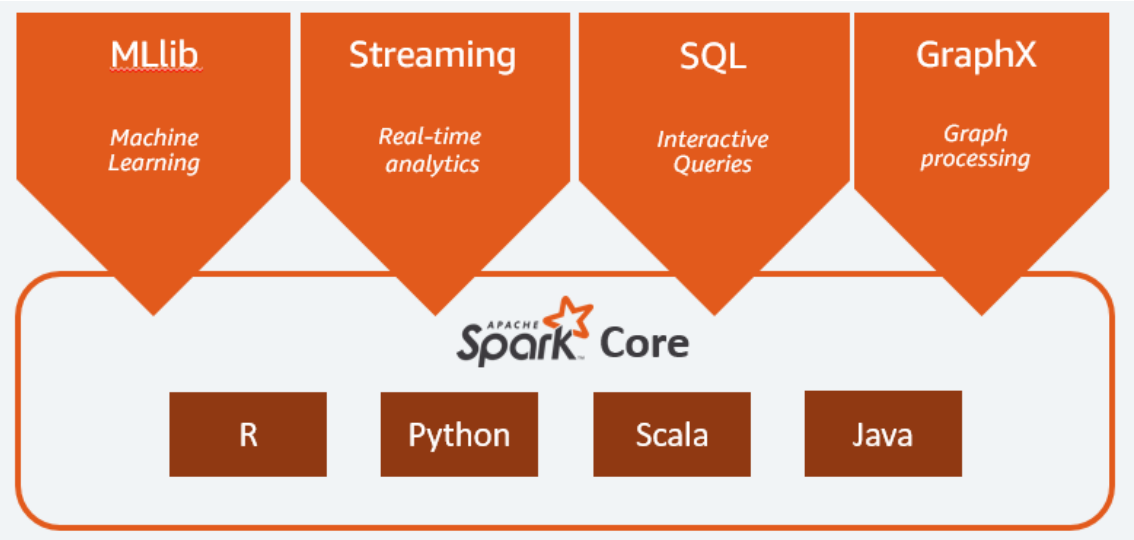


Hình 3. Kiến trúc HDFS

- **Ưu điểm:**
  - **Khả năng chịu lỗi (Fault Tolerance):** Dữ liệu được tự động nhân bản (replication) trên nhiều DataNode. Nếu một nút bị lỗi, dữ liệu vẫn có thể được truy xuất từ các bản sao khác.
  - **Khả năng mở rộng (Scalability):** Dễ dàng mở rộng dung lượng lưu trữ bằng cách thêm các DataNode mới vào cụm.
  - **Tối ưu cho tập dữ liệu lớn:** Phù hợp để lưu trữ các tệp có kích thước từ gigabyte đến terabyte.

1.3.2. Apache Spark và hệ sinh thái

Apache Spark là một framework tính toán phân tán mã nguồn mở, được phát triển để khắc phục những hạn chế của MapReduce. Ưu điểm vượt trội của Spark là khả năng xử lý dữ liệu trong bộ nhớ (in-memory processing), giúp tăng tốc độ thực thi lên gấp nhiều lần so với MapReduce.



Hình 4. Sơ đồ minh họa hệ sinh thái Apache Spark

- **Spark Core:** Là trái tim của Spark, cung cấp các chức năng cơ bản như quản lý tác vụ, quản lý bộ nhớ và điều phối I/O. Cấu trúc dữ liệu cốt lõi của Spark là Resilient Distributed Dataset (RDD), một tập hợp các phần tử không thể thay đổi, được phân tán trên các nút của cụm.
- **Spark SQL:** Là một module của Spark để làm việc với dữ liệu có cấu trúc. Nó cho phép người dùng truy vấn dữ liệu thông qua câu lệnh SQL hoặc thông qua API DataFrame/Dataset, cung cấp một tầng trừu tượng hóa cao hơn so với RDD.
- **Spark MLlib:** Là thư viện học máy của Spark, cung cấp một bộ các thuật toán và công cụ phổ biến (phân loại, hồi quy, phân cụm) được tối ưu hóa để chạy song song trên cụm. Trong đồ án này, MLlib được sử dụng để xây dựng các mô hình dự đoán.

1.3.3. Cơ sở dữ liệu NoSQL – MongoDB

MongoDB là một hệ quản trị cơ sở dữ liệu NoSQL hướng tài liệu (document-oriented), mã nguồn mở. Thay vì lưu dữ liệu trong các bảng và hàng như cơ sở dữ liệu quan hệ, MongoDB lưu trữ dữ liệu dưới dạng các tài liệu BSON (một dạng nhị phân của JSON).

- **Lý do lựa chọn:**
    - **Lược đồ linh hoạt (Flexible Schema):** Rất phù hợp với dữ liệu bán cấu trúc từ YouTube, nơi các thuộc tính có thể thay đổi hoặc không phải lúc nào cũng đầy đủ.
    - **Khả năng mở rộng ngang (Horizontal Scaling):** Dễ dàng mở rộng bằng cách thêm các máy chủ vào cụm (sharding).
    - **Hiệu năng cao:** Hiệu năng đọc/ghi tốt, phù hợp cho các ứng dụng web cần phản hồi nhanh.

Trong ứng dụng, MongoDB được sử dụng để lưu trữ dữ liệu đã qua xử lý, sẵn sàng cho việc truy vấn và hiển thị trên ứng dụng web.

1.3.4. Kiến trúc Web (FastAPI, React)

- **FastAPI:** Là một web framework hiện đại, hiệu năng cao cho việc xây dựng các API bằng Python. FastAPI được xây dựng dựa trên Starlette và Pydantic, hỗ trợ lập trình bất đồng bộ (asynchronous), giúp xử lý đồng thời nhiều yêu cầu với hiệu suất cao. Nó được sử dụng để xây dựng backend, cung cấp các endpoint cho frontend truy xuất dữ liệu và kết quả dự đoán.
- **React:** Là một thư viện JavaScript mã nguồn mở, được phát triển bởi Facebook, dùng để xây dựng giao diện người dùng (UI). React cho phép

xây dựng các thành phần UI (components) có thể tái sử dụng và quản lý trạng thái của ứng dụng một cách hiệu quả. Nó được sử dụng để xây dựng frontend của ứng dụng, tạo ra một trải nghiệm người dùng tương tác và linh hoạt.

1.4. Tổng quan các nghiên cứu liên quan

Việc phân tích dữ liệu mạng xã hội, đặc biệt là YouTube, đã là chủ đề của nhiều công trình nghiên cứu trước đây. Các nghiên cứu này thường tập trung vào các khía cạnh như:

- Phân tích các yếu tố ảnh hưởng đến sự lan truyền (viral) của video.
- Xây dựng các mô hình đề xuất nội dung.
- Phân tích cảm xúc (sentiment analysis) dựa trên bình luận của người xem.
- Phát hiện các cộng đồng và người có ảnh hưởng (influencers) trên nền tảng.

Đồ án này kế thừa các hướng tiếp cận đó nhưng tập trung vào việc xây dựng một hệ thống Big Data hoàn chỉnh, không chỉ phân tích mà còn cung cấp khả năng dự đoán và trực quan hóa cho người dùng cuối, giải quyết một bài toán thực tiễn với bộ công nghệ xử lý phân tán hiện đại.

CHƯƠNG 2. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

Sau khi đã xác định các cơ sở lý thuyết, chương này sẽ đi sâu vào việc phân tích các yêu cầu cụ thể và trình bày chi tiết về kiến trúc thiết kế của hệ thống. Đây là bước bản lề, chuyển từ "cái gì" (what) sang "như thế nào" (how) để hiện thực hóa các mục tiêu đã đề ra.

2.1. Phân tích yêu cầu

2.1.1. Yêu cầu chức năng (Functional Requirements)

Yêu cầu chức năng mô tả các tác vụ mà hệ thống phải thực hiện. Dựa trên mục tiêu của đồ án, các chức năng chính bao gồm:

- **FN-1: Xử lý và Tích hợp Dữ liệu:** Hệ thống phải có khả năng đọc nhiều file dữ liệu thô (CSV) từ các quốc gia khác nhau, hợp nhất chúng thành một bộ dữ liệu duy nhất.
- **FN-2: Làm sạch Dữ liệu:** Hệ thống phải thực hiện các bước tiền xử lý cần thiết như xử lý giá trị thiếu, chuẩn hóa kiểu dữ liệu, và loại bỏ các bản ghi không hợp lệ.
- **FN-3: Phân tích Khám phá (EDA):** Hệ thống phải cung cấp các chức năng thống kê và tổng hợp dữ liệu để người dùng có thể khám phá các xu hướng, ví dụ:
    - Top các kênh có nhiều video thịnh hành nhất.
    - Phân bố video thịnh hành theo thể loại.
    - Mối tương quan giữa lượt xem, lượt thích và bình luận.
- **FN-4: Huấn luyện Mô hình Học máy:** Hệ thống phải có khả năng huấn luyện các mô hình học máy từ dữ liệu đã xử lý để:
    - Phân cụm các video có nội dung tương tự.
    - Dự đoán số ngày một video có thể duy trì trên tab thịnh hành.

- **FN-5: Cung cấp API:** Hệ thống phải cung cấp các API endpoint để ứng dụng web có thể truy xuất kết quả phân tích và thực hiện dự đoán.
- **FN-6: Trực quan hóa Dữ liệu:** Giao diện người dùng phải hiển thị các kết quả phân tích một cách trực quan thông qua biểu đồ, bảng biểu.
- **FN-7: Giao diện Dự đoán:** Người dùng có thể nhập các thông số của một video và nhận lại kết quả dự đoán từ mô hình.

2.1.2. Yêu cầu phi chức năng (Non-functional Requirements)

Yêu cầu phi chức năng mô tả các tiêu chí về chất lượng và hiệu suất của hệ thống.

- **NFN-1: Hiệu năng (Performance):** Tác vụ xử lý dữ liệu lớn (sử dụng Spark) cần được hoàn thành trong thời gian hợp lý. Các truy vấn từ ứng dụng web đến API phải có thời gian phản hồi nhanh (dưới 2 giây).
- **NFN-2: Khả năng mở rộng (Scalability):** Kiến trúc hệ thống, đặc biệt là tầng xử lý dữ liệu và lưu trữ, phải có khả năng mở rộng để xử lý khối lượng dữ liệu lớn hơn trong tương lai bằng cách thêm tài nguyên phần cứng.
- **NFN-3: Tính sẵn sàng (Availability):** Ứng dụng web phải luôn sẵn sàng để người dùng truy cập.
- **NFN-4: Dễ sử dụng (Usability):** Giao diện người dùng phải thân thiện, dễ hiểu và dễ thao tác, ngay cả với người dùng không có chuyên môn về kỹ thuật.
- **NFN-5: Khả năng bảo trì (Maintainability):** Mã nguồn của dự án cần được tổ chức rõ ràng, có module hóa để dễ dàng sửa lỗi, nâng cấp và phát triển các tính năng mới.

2.2. Kiến trúc tổng thể của hệ thống

Hệ thống được thiết kế theo kiến trúc nhiều lớp (multi-layered architecture), phân tách rõ ràng các nhiệm vụ xử lý và phục vụ, bao gồm các thành phần chính sau:

Hình 5. Sơ đồ kiến trúc tổng thể của hệ thống

- **Tầng Dữ liệu (Data Layer):**
  - HDFS (Hadoop Distributed File System): Đóng vai trò là nơi lưu trữ dữ liệu thô (raw data) dưới dạng các file CSV. Đây là nguồn dữ liệu đầu vào cho quá trình xử lý.
  - MongoDB: Đóng vai trò là cơ sở dữ liệu phục vụ (serving database). Dữ liệu sau khi đã được xử lý, làm sạch và tổng hợp bởi Spark sẽ được lưu trữ tại đây để cung cấp cho tầng backend một cách nhanh chóng.
- **Tầng Xử lý (Processing Layer):**
  - Apache Spark: Là "bộ não" của hệ thống, chịu trách nhiệm cho tất cả các tác vụ xử lý dữ liệu nặng. Các job Spark sẽ đọc dữ liệu từ HDFS, thực hiện tiền xử lý, phân tích, huấn luyện mô hình ML và cuối cùng là lưu kết quả vào MongoDB.
- **Tầng Ứng dụng (Application Layer):**
  - Backend (FastAPI): Xây dựng các API RESTful để làm cầu nối giữa tầng dữ liệu (MongoDB) và tầng trình bày. Backend sẽ xử lý các logic nghiệp vụ, truy vấn dữ liệu từ MongoDB, và gọi các mô hình ML đã được lưu để thực hiện dự đoán.

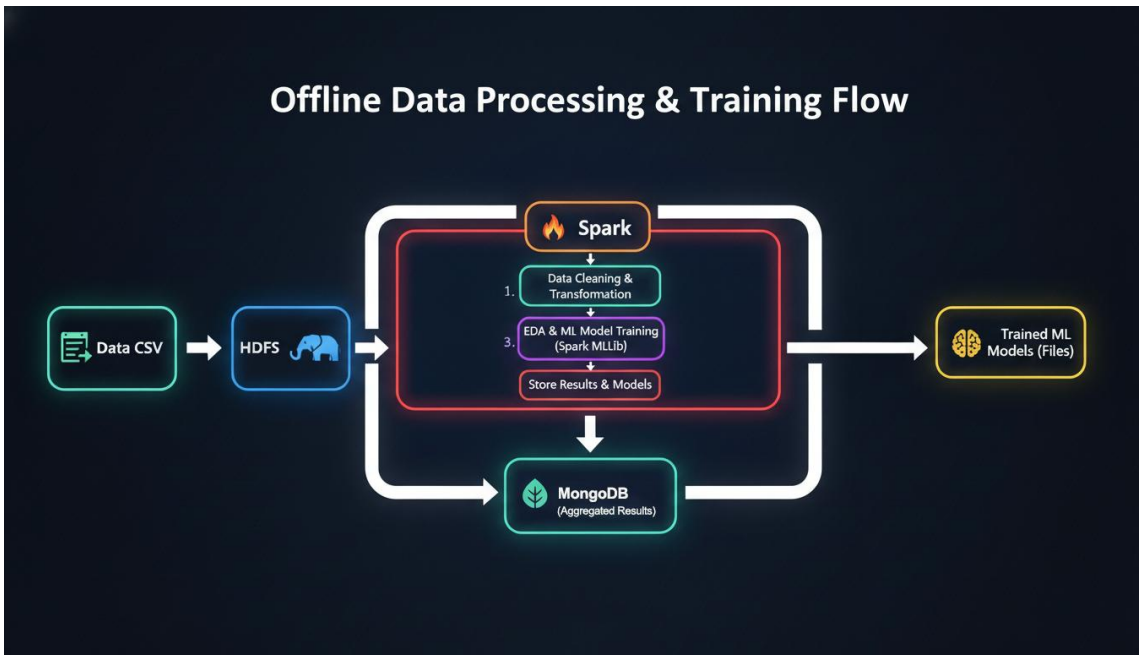
- Frontend (React): Là giao diện người dùng cuối, được xây dựng dưới dạng một ứng dụng trang đơn (Single Page Application). Frontend sẽ gọi các API từ backend để lấy dữ liệu và hiển thị cho người dùng dưới dạng các dashboard, biểu đồ và form tương tác.

2.3. Luồng dữ liệu (Data Flow)

Dựa trên kiến trúc đã thiết kế, luồng di chuyển của dữ liệu trong hệ thống có thể được chia thành hai luồng chính:

2.3.1. Luồng xử lý và huấn luyện (Offline)

Đây là luồng xử lý dữ liệu theo lô, không yêu cầu tương tác thời gian thực từ người dùng.



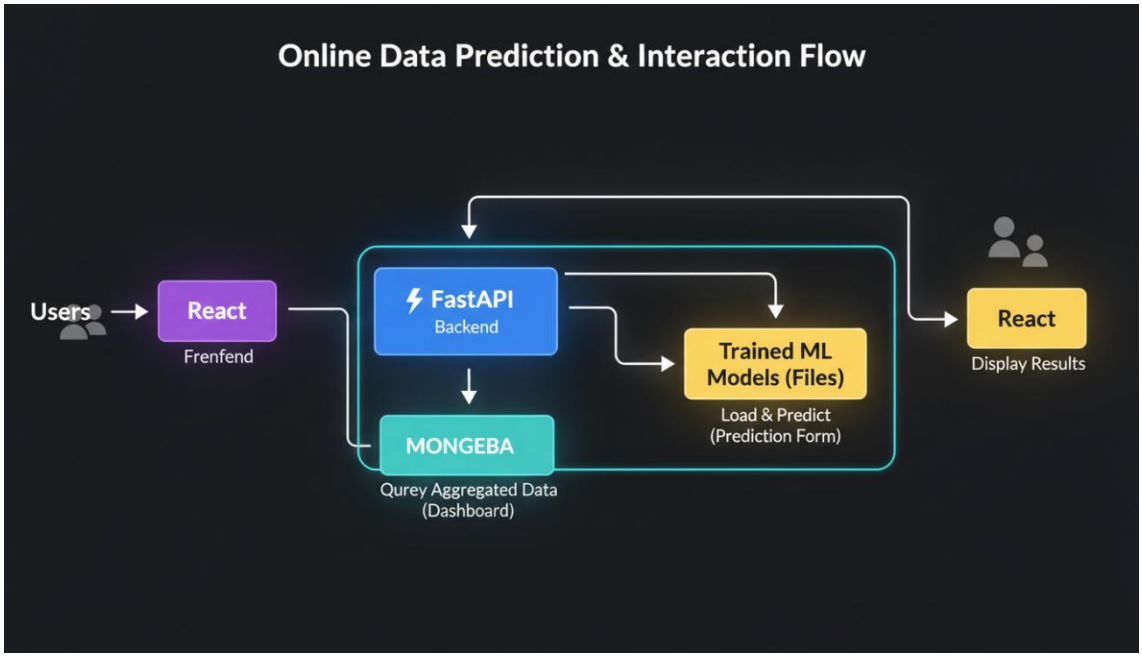
Hình 6. Sơ đồ luồng xử lý và huấn luyện (Offline)

1. **Thu thập:** Dữ liệu thô (các file CSV, JSON về video trending) được thu thập và đưa vào lưu trữ trên ***HDFS***.
2. **Xử lý:** Một ***Spark Job*** được kích hoạt để đọc toàn bộ dữ liệu từ HDFS.
3. **Làm sạch & Chuyển đổi:** Spark thực hiện các phép biến đổi như hợp nhất dữ liệu, xử lý giá trị null, thay đổi kiểu dữ liệu.

4. **Phân tích & Huấn luyện:** Dữ liệu sạch được sử dụng để thực hiện các phép tính toán thống kê (EDA) và huấn luyện các mô hình Machine Learning (Clustering, Regression) bằng ***Spark MLlib***.
5. **Lưu trữ:**
    - Các kết quả phân tích tổng hợp (ví dụ: top kênh, thống kê theo thể loại) được lưu vào các collection trong ***MongoDB***.
    - Các mô hình ML sau khi huấn luyện xong được lưu lại dưới dạng file để tầng backend có thể tải và sử dụng sau này.

2.3.2. Luồng dự đoán và tương tác (Online)

Đây là luồng dữ liệu khi người dùng tương tác trực tiếp với ứng dụng web.



Hình 7. Sơ đồ luồng dự đoán và tương tác (Online)

1. **Yêu cầu (Request):** Người dùng truy cập vào ứng dụng web (***React Frontend***) và thực hiện một hành động (ví dụ: xem dashboard hoặc gửi form dự đoán).
2. **Gọi API:** Frontend gửi một HTTP request đến ***Backend (FastAPI)***.
3. **Truy vấn & Xử lý:**

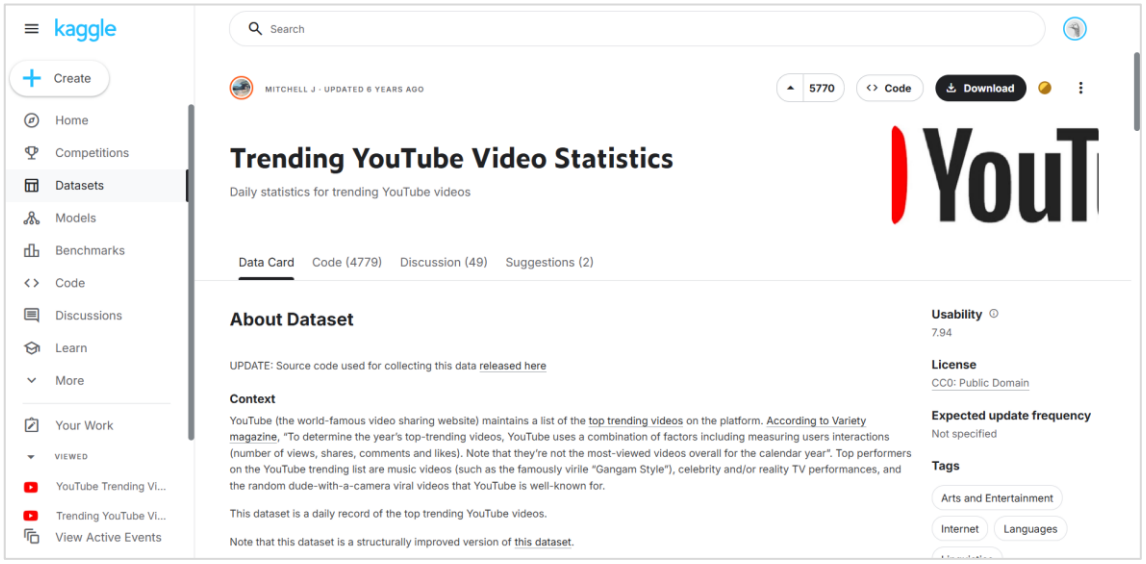
- Nếu là yêu cầu xem dashboard, Backend sẽ truy vấn dữ liệu phân tích đã được chuẩn bị sẵn từ ***MongoDB***.
- Nếu là yêu cầu dự đoán, Backend sẽ tải mô hình ML đã được huấn luyện, xử lý dữ liệu đầu vào từ người dùng và dùng mô hình để đưa ra kết quả.

4. **Phản hồi (Response):** Backend trả về kết quả cho Frontend dưới dạng JSON.
5. **Hiển thị:** Frontend nhận dữ liệu JSON và render ra giao diện cho người dùng xem (hiển thị biểu đồ, kết quả dự đoán).

CHƯƠNG 3. THU THẬP VÀ TIỀN XỬ LÝ DỮ LIỆU

Dữ liệu là tài sản cốt lõi của bất kỳ hệ thống phân tích nào. Chất lượng của dữ liệu đầu vào ảnh hưởng trực tiếp đến độ chính xác và giá trị của kết quả phân tích cũng như mô hình dự đoán. Do đó, giai đoạn thu thập, làm sạch và tiền xử lý dữ liệu đóng một vai trò vô cùng quan trọng trong quy trình của đồ án.

3.1. Nguồn và mô tả dữ liệu



Hình 8. Bộ dữ liệu Kaggle "Trending YouTube Video Statistics"

- **Nguồn dữ liệu:** Đồ án sử dụng bộ dữ liệu công khai "YouTube Trending Video Dataset", được thu thập và chia sẻ trên nền tảng Kaggle. Bộ dữ liệu này ghi lại thông tin hàng ngày về các video lọt vào top thịnh hành tại nhiều quốc gia khác nhau trong giai đoạn từ năm 2017 đến 2018.
- **Cấu trúc dữ liệu:** Dữ liệu được cung cấp dưới dạng các file CSV cho mỗi quốc gia (*ví dụ: USvideos.csv, GBvideos.csv*) và các file JSON chứa thông tin về danh mục (category). Mỗi bản ghi trong file CSV tương ứng với một video tại một thời điểm nó xuất hiện trên tab thịnh hành.

Tên thuộc tính	Kiểu dữ liệu	Mô tả
video_id	String	ID định danh duy nhất cho mỗi video.
trending_date	Date	Ngày video xuất hiện trên tab thịnh hành.
title	String	Tiêu đề của video.
channel_title	String	Tên kênh YouTube đã đăng tải video.
category_id	Integer	ID của thể loại video (ví dụ: 10 cho Âm nhạc, 24 cho Giải trí).
publish_time	Datetime	Thời điểm video được đăng tải công khai.
tags	String	Danh sách các thẻ (tags) liên quan đến video, phân tách bởi dấu `
views	Integer	Tổng số lượt xem của video tại thời điểm ghi nhận.
likes	Integer	Tổng số lượt thích của video.
dislikes	Integer	Tổng số lượt không thích của video.
comment_count	Integer	Tổng số bình luận của video.
description	String	Phần mô tả của video.

Bảng 1. Mô tả các thuộc tính chính

3.2. Pipeline thu thập và xử lý dữ liệu với Apache Spark

Toàn bộ quá trình xử lý và làm sạch dữ liệu được thực hiện thông qua một Spark Job. Quy trình này được thiết kế để xử lý hiệu quả khối lượng dữ liệu lớn một cách song song, tận dụng sức mạnh của framework Apache Spark.

3.2.1. Đọc và hợp nhất dữ liệu từ HDFS

- **Đọc dữ liệu:** Bước đầu tiên, Spark đọc đồng thời tất cả các file CSV của các quốc gia và file JSON về thể loại từ HDFS. Việc đọc nhiều file cùng lúc giúp tăng tốc độ nhập dữ liệu.
- **Thêm trường quốc gia:** Để phân biệt dữ liệu từ các nguồn khác nhau, một cột `country` được thêm vào mỗi DataFrame tương ứng với quốc gia của file dữ liệu đó (ví dụ: 'US', 'GB').
- **Hợp nhất (Union):** Tất cả các DataFrame từ các quốc gia được hợp nhất thành một DataFrame duy nhất.
- **Nối (Join) thông tin thể loại:** DataFrame hợp nhất sau đó được nối (join) với dữ liệu từ file JSON dựa trên `category_id` để có được tên thể loại (`category_name`) tương ứng.

3.2.2. Làm sạch và xử lý dữ liệu (Data Cleaning & Preprocessing)

Đây là bước quan trọng nhất để đảm bảo chất lượng dữ liệu. Dựa trên việc khảo sát dữ liệu ban đầu, các tác vụ sau được thực hiện:

- **Chuẩn hóa kiểu dữ liệu:** Các cột ngày tháng như `trending_date` và `publish_time` được chuyển đổi sang đúng kiểu dữ liệu `DateType` và `TimestampType` để có thể thực hiện các phép tính toán về thời gian. Các cột số liệu (`views, likes, dislikes, comment_count`) được chuyển thành kiểu số nguyên (`IntegerType`).
- **Xử lý giá trị thiếu (Handling Missing Values):** Các bản ghi có giá trị thiếu ở các cột quan trọng như `title` hoặc `channel_title` sẽ bị loại bỏ. Đối với cột `description`, giá trị thiếu được thay thế bằng một chuỗi rỗng.

- **Loại bỏ bản ghi trùng lặp:** Dữ liệu có thể chứa các bản ghi bị trùng lặp hoàn toàn. Hệ thống sử dụng phương thức `dropDuplicates()` để loại bỏ chúng.
- **Trích xuất thông tin thời gian:** Từ cột `publish_time`, các thông tin hữu ích hơn được trích xuất và tạo thành các cột mới như `publish_date`, `publish_hour` để phục vụ cho việc phân tích xu hướng theo thời gian.

3.3. Lưu trữ dữ liệu đã qua xử lý

Sau khi hoàn tất quá trình làm sạch và xử lý, dữ liệu được chuẩn bị để lưu trữ cho các mục đích khác nhau.

3.3.1. Lưu trữ vào MongoDB cho ứng dụng web

Để ứng dụng web có thể truy vấn và hiển thị thông tin một cách nhanh chóng, các kết quả phân tích và tổng hợp từ Spark sẽ được tính toán trước và lưu vào MongoDB. Các quy trình này bao gồm:

- Tính toán các chỉ số tổng hợp như top 10 kênh có nhiều video trending nhất, tổng lượt xem theo từng thể loại, v.v.
- Chuyển đổi DataFrame kết quả của Spark thành định dạng phù hợp và ghi vào các collection tương ứng trong MongoDB. Việc này giúp giảm tải cho backend, vì nó chỉ cần đọc dữ liệu đã được tính toán sẵn thay vì phải tính toán lại mỗi khi có yêu cầu.

3.3.2. Cấu trúc các collection trong MongoDB

Tên Collection	Mục đích	Ví dụ tài liệu (Document)
`top_channels`	Lưu trữ top các kênh có nhiều video thịnh hành nhất.	`{ "channel_title": "...", "video_count": 120 }`

`category_stats`	Lưu trữ thống kê tổng hợp theo từng thể loại.	`{ "category": "Music", "total_views": 500000000, "avg_likes": 150000 }`
`trending_overview`	Lưu trữ các chỉ số tổng quan về dữ liệu.	`{ "total_videos": 40000, "total_channels": 6000 }`

Bảng 2. Mô tả cấu trúc các collection trong MongoDB

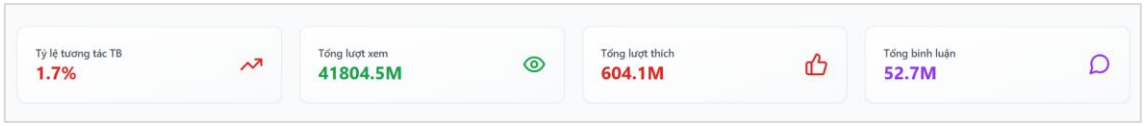
MongoDB lưu trữ kết quả sau xử lý Spark, bao gồm raw data, ML features và trending analysis. Nó sử dụng tài liệu JSON linh hoạt, hỗ trợ truy vấn nhanh cho API backend.

CHƯƠNG 4. PHÂN TÍCH DỮ LIỆU KHÁM PHÁ (EDA)

Phân tích dữ liệu khám phá (EDA) là quá trình trực quan hóa và tóm tắt dữ liệu để rút ra những hiểu biết ban đầu. Chương này tập trung vào việc mô tả các kết quả phân tích đã được chắt lọc và tích hợp vào giao diện người dùng của ứng dụng, phản ánh chính xác các chức năng mà người dùng cuối có thể tương tác.

4.1. Tổng quan các chỉ số chính

Ngay trên trang "Phân tích Video Trending", ứng dụng hiển thị một bộ bốn thẻ thống kê (Stat Cards) cung cấp cái nhìn nhanh về hiệu suất của các video trong bộ lọc hiện tại. Các chỉ số này được tính toán động ở phía frontend dựa trên dữ liệu được API trả về.



- **Tỷ lệ tương tác TB (Trung bình):** Đây là chỉ số quan trọng đo lường mức độ tương tác trung bình. Nó được tính bằng tổng (lượt thích + bình luận) / lượt xem của tất cả video, sau đó chia cho tổng số video.
- **Tổng lượt xem:** Tổng cộng tất cả lượt xem của các video đang được hiển thị.
- **Tổng lượt thích:** Tổng cộng tất cả lượt thích của các video đang được hiển thị.
- **Tổng bình luận:** Tổng cộng tất cả bình luận của các video đang được hiển thị.

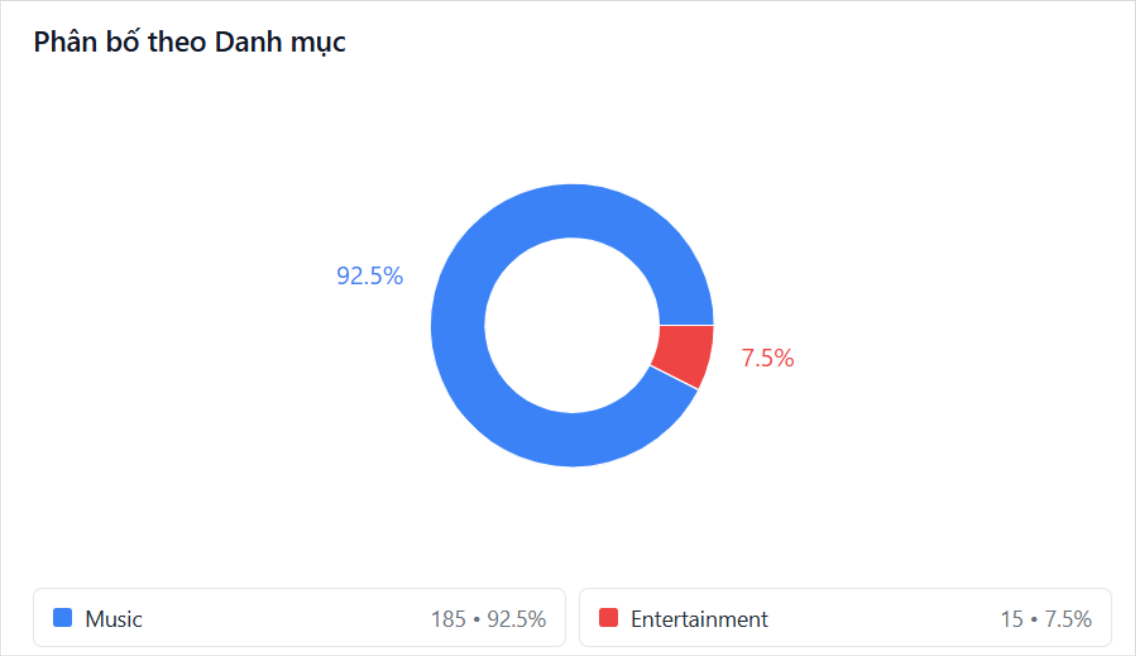
Các chỉ số này giúp người dùng nhanh chóng đánh giá quy mô và mức độ tương tác chung của tập video mà họ đang phân tích.

4.2. Trực quan hóa và Phân tích Chuyên sâu

4.2.1. Phân bố theo Danh mục (Category Distribution)

Để hiểu thể loại nội dung nào chiếm ưu thế trên tab thịnh hành, ứng dụng sử dụng biểu đồ tròn (Donut Chart) để thể hiện tỷ lệ phần trăm của mỗi thể loại.

- **Mục đích:** Giúp người dùng nhanh chóng nhận diện các thể loại video phổ biến nhất.
- **Phân tích:** Dựa trên biểu đồ, có thể thấy rõ ràng hai thể loại ***"Entertainment" (Giải trí)*** và ***"Music" (Âm nhạc)*** thường chiếm tỷ trọng lớn nhất trong các video thịnh hành. Giao diện cũng cho phép người dùng nhấp vào một danh mục trong chú thích để lọc toàn bộ trang theo danh mục đó.



4.2.2. Phân tích từ khóa nổi bật qua Word Cloud

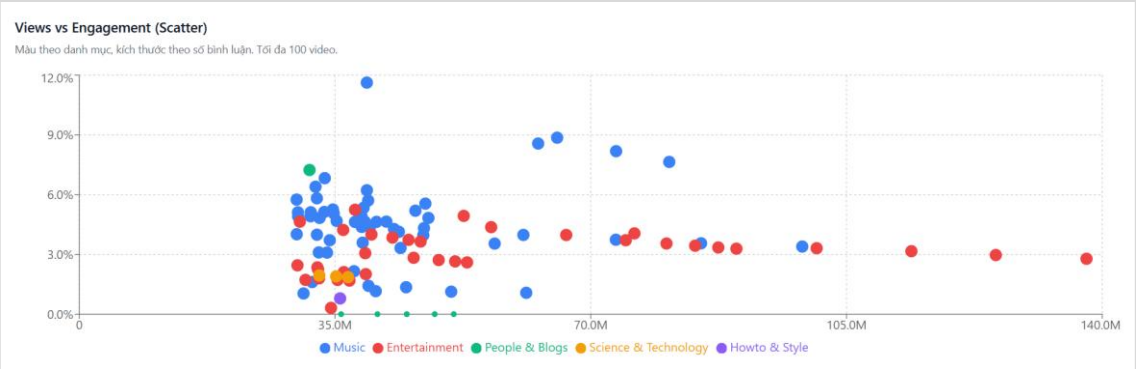
Ứng dụng cung cấp một công cụ trực quan mạnh mẽ là Word Cloud (Đám mây từ) được tạo ra từ tiêu đề của hàng ngàn video thịnh hành, thay thế cho việc chỉ liệt kê các kênh.

- **Mục đích:** Word Cloud giúp nhận diện nhanh chóng các chủ đề, từ khóa và xu hướng đang được quan tâm nhiều nhất. Kích thước của mỗi từ tỷ lệ thuận với tần suất xuất hiện của nó trong các tiêu đề.
- **Phân tích:** Các từ có kích thước lớn như ***"Official", "Video", "Trailer", "Music"*** cho thấy người dùng có xu hướng xem các nội dung chính thức từ nhà sản xuất. Đây là một công cụ hữu ích để nắm bắt nhanh các chủ đề nóng tại một thời điểm.

4.2.3. Mối quan hệ giữa Lượt xem và Tỷ lệ tương tác

Để đánh giá sâu hơn chất lượng của một video, ứng dụng cung cấp một biểu đồ phân tán (Scatter Plot) để khám phá mối quan hệ giữa ***Lượt xem (Views)*** và ***Tỷ lệ tương tác (Engagement Rate)***.

- **Mục đích:** Biểu đồ này giúp xác định xem liệu một video có nhiều lượt xem có thực sự tạo ra được sự tương tác mạnh mẽ từ cộng đồng hay không.
- **Phân tích:** Các điểm dữ liệu trên biểu đồ, được tô màu theo thể loại, cho thấy không phải lúc nào video có nhiều lượt xem nhất cũng có tỷ lệ tương tác cao nhất. Có những video dù có lượt xem ở mức vừa phải nhưng lại tạo ra tỷ lệ tương tác rất cao, cho thấy nội dung có chiều sâu và kết nối tốt với khán giả. Kích thước của mỗi điểm trên biểu đồ cũng đại diện cho số lượng bình luận, cung cấp thêm một chiều thông tin.



CHƯƠNG 5. XÂY DỰNG MÔ HÌNH DỰ ĐOÁN

Sau khi đã hiểu sâu hơn về dữ liệu qua giai đoạn phân tích khám phá, bước tiếp theo là xây dựng các mô hình học máy để tự động hóa việc trích xuất thông tin và đưa ra các dự đoán có giá trị. Chương này trình bày chi tiết quy trình từ kỹ thuật đặc trưng, lựa chọn mô hình, huấn luyện cho đến đánh giá hiệu năng. Toàn bộ quy trình được thực hiện trên Apache Spark bằng thư viện MLlib.

5.1. Kỹ thuật đặc trưng (Feature Engineering)

Kỹ thuật đặc trưng là quá trình sử dụng kiến thức miền (domain knowledge) để tạo ra các biến đầu vào (features) mới từ dữ liệu thô, giúp các thuật toán học máy hoạt động hiệu quả hơn. Thay vì chỉ sử dụng các số liệu gốc, chúng ta sẽ tạo ra các đặc trưng mang nhiều ý nghĩa hơn.

5.1.1. Tạo các đặc trưng về tương tác (Engagement Metrics)

Các chỉ số này đo lường mức độ khán giả tương tác với nội dung, thay vì chỉ xem một cách thụ động.

Tên đặc trưng	**Công thức tính**	**Ý nghĩa**
`like_ratio`	`likes / views`	Tỷ lệ lượt thích trên mỗi lượt xem. Chỉ số này thể hiện mức độ yêu thích của khán giả.
`comment_ratio`	`comment_count / views`	Tỷ lệ bình luận trên mỗi lượt xem. Chỉ số này cho thấy mức độ thảo luận mà video tạo ra.
`engagement_score`	`(likes + comment_count) / views`	Một điểm số tổng hợp đo lường mức độ tương tác chung của video.

Bảng 3. Mô tả các đặc trưng về tương tác

5.1.2. Tạo các đặc trưng về nội dung (Content Features)

Các đặc trưng này mô tả bản chất của nội dung video dựa trên siêu dữ liệu (metadata).

Tên đặc trưng	Cách tạo	Ý nghĩa
`title_length`	Độ dài của chuỗi `title`.	Độ dài tiêu đề có thể ảnh hưởng đến tỷ lệ nhấp chuột (CTR).
`tag_count`	Số lượng thẻ trong cột `tags`.	Số lượng tags có thể liên quan đến khả năng khám phá của video.
`days_to_trend`	`trending_date - publish_date`	Số ngày kể từ khi video được đăng tải cho đến khi lọt top thịnh hành.

Bảng 4. Mô tả các đặc trưng về nội dung

5.1.3. Vector hóa văn bản (Text Vectorization)

Để các thuật toán có thể hiểu được dữ liệu văn bản (như tiêu đề và tags), chúng ta cần chuyển đổi chúng thành dạng số. Quy trình này bao gồm các bước:

1. **Tokenizer:** Tách chuỗi văn bản thành các từ (token) riêng lẻ.
2. **StopWordsRemover:** Loại bỏ các từ phổ biến nhưng không mang nhiều ý nghĩa (ví dụ: "the", "a", "is").
3. **TF-IDF (Term Frequency-Inverse Document Frequency):** Tính toán trọng số cho mỗi từ, phản ánh mức độ quan trọng của từ đó trong một tiêu đề cụ thể so với toàn bộ kho dữ liệu. Kết quả là mỗi tiêu đề được biểu diễn bằng một vector số.

5.2. Lựa chọn và huấn luyện mô hình

Dựa trên mục tiêu của đồ án, hai bài toán học máy chính được xác định: phân cụm nội dung và dự đoán mức độ duy trì trên top thịnh hành.

5.2.1. Mô hình phân cụm nội dung (K-Means)

- **Mục tiêu:** Tự động nhóm các video có nội dung tương tự nhau vào cùng một cụm dựa trên vector TF-IDF của tiêu đề.
- **Thuật toán:** K-Means là một thuật toán học không giám sát, cố gắng phân chia N điểm dữ liệu vào K cụm khác nhau sao cho tổng bình phương khoảng cách từ mỗi điểm đến tâm cụm của nó là nhỏ nhất.
- **Huấn luyện:** Mô hình được huấn luyện trên ma trận TF-IDF của toàn bộ tiêu đề video. Số lượng cụm (K) được lựa chọn dựa trên thực nghiệm để cho ra các nhóm có ý nghĩa nhất.

5.2.2. Mô hình dự đoán số ngày trên top thịnh hành (Random Forest Regression)

- **Mục tiêu:** Dự đoán một video sẽ duy trì được bao nhiêu ngày trong danh sách thịnh hành. Đây là một bài toán hồi quy (regression).
- **Biến mục tiêu (Target Variable):** days_in_trending.
- **Các biến đầu vào (Features):** Bao gồm tất cả các đặc trưng đã tạo ở mục 5.1 (đặc trưng tương tác, nội dung) và các thuộc tính gốc đã được chuẩn hóa (lượt xem, thể loại, quốc gia).
- **Thuật toán:** Random Forest (Rừng Ngẫu nhiên) được lựa chọn vì những ưu điểm:
    - Hoạt động tốt với cả dữ liệu số và dữ liệu phân loại.
    - Có khả năng xử lý lượng lớn đặc trưng đầu vào.
    - Ít bị ảnh hưởng bởi hiện tượng quá khớp (overfitting) so với một cây quyết định đơn lẻ.

- Cung cấp thông tin về mức độ quan trọng của các đặc trưng (feature importance).

- **Huấn luyện:** Dữ liệu được chia thành hai tập: tập huấn luyện (80%) và tập kiểm thử (20%). Mô hình được huấn luyện trên tập huấn luyện.

5.3. Đánh giá hiệu năng mô hình

Sau khi huấn luyện, các mô hình cần được đánh giá trên tập dữ liệu kiểm thử để đo lường hiệu suất.

- **Đánh giá mô hình phân cụm (Clustering):**
  - **Mô hình:** K-Means**.**
  - **Chỉ số Silhouette:** Được sử dụng để đo lường mức độ cô đọng và tách biệt của các cụm. Kết quả thực tế cho thấy mô hình đạt chỉ số Silhouette là 0.327. Một giá trị dương cho thấy các điểm dữ liệu nằm trong cụm của chúng gần hơn so với các cụm lân cận, chứng tỏ mô hình đã phân chia được các cụm có cấu trúc nhất định. Tuy nhiên, giá trị này ở mức trung bình, cho thấy có thể có sự chồng chéo giữa các cụm.
- **Đánh giá mô hình hồi quy (Regression):**
  - **Mô hình:** Random Forest Regression**.**
  - **Phân tích:** Hiệu suất của mô hình dự đoán số ngày trên top thịnh hành là rất tốt, thể hiện qua các chỉ số sau**:**

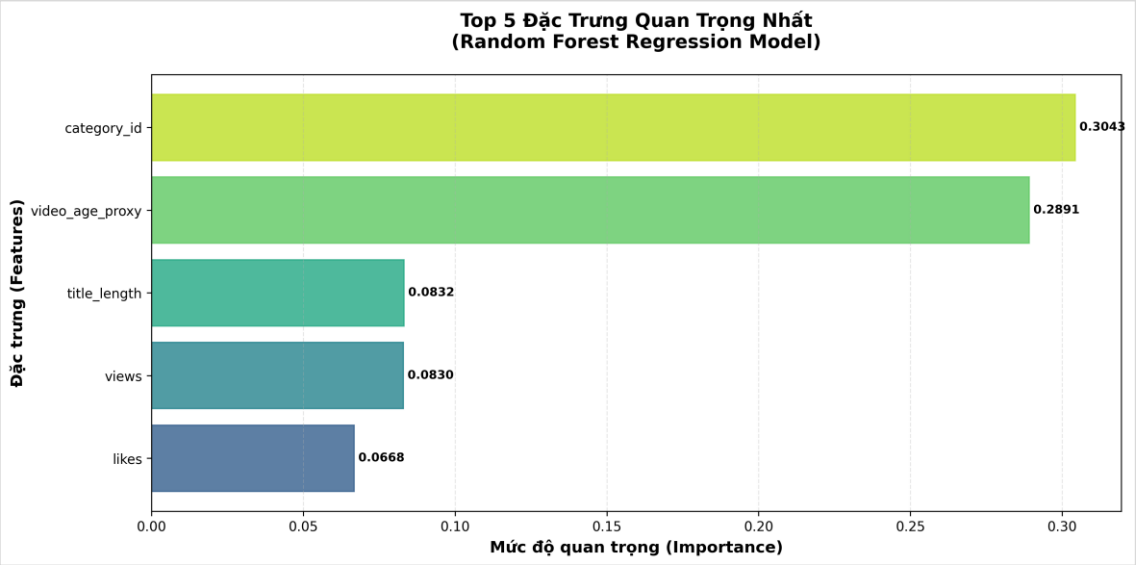
Chỉ số đánh giá	**Giá trị thực tế**	**Ý nghĩa**
R^2 (R-squared)	0.737	Hệ số xác định. Kết quả này rất ấn tượng, cho thấy mô hình có thể giải thích được khoảng 73.7% sự biến thiên của số ngày trên top thịnh hành. Điều này chứng tỏ các đặc trưng đầu vào có khả năng dự đoán mạnh mẽ.
RMSE (Root Mean	1.648	Căn bậc hai của sai số bình phương trung bình. Kết quả này cho thấy, về trung bình, dự đoán của mô hình có thể chênh lệch (cao hơn hoặc thấp hơn) khoảng 1.65 ngày so

Squared Error)		với giá trị thực tế. Đây là một mức sai số thấp và chấp nhận được.
MAE (Mean Absolute Error)	0.626	Sai số tuyệt đối trung bình. Cho biết trung bình, dự đoán của mô hình lệch khỏi giá trị thực tế là khoảng 0.63 ngày, củng cố thêm độ chính xác của mô hình.

Bảng 5. Kết quả đánh giá mô hình hồi quy

- **Phân tích độ quan trọng của đặc trưng (Feature Importance):**

Mô hình Random Forest không chỉ đưa ra dự đoán mà còn cho phép chúng ta đo lường mức độ ảnh hưởng của từng đặc trưng (feature) đến kết quả cuối cùng. Bằng cách phân tích "độ quan trọng" của các đặc trưng, chúng ta có thể hiểu được yếu tố nào là then chốt nhất.



Hình 9. Biểu đồ "Top 5 Đặc Trưng Quan Trọng Nhất"

Dựa trên biểu đồ, kết quả phân tích cho thấy:

- **`category_id` (Thể loại video)** là đặc trưng có sức ảnh hưởng lớn nhất với điểm quan trọng là 0.3043. Điều này cho thấy việc video thuộc về một thể loại cụ thể (ví dụ: Âm nhạc, Giải trí) là yếu tố dự báo mạnh mẽ nhất về khả năng duy trì trên top thịnh hành.
- **`video_age_proxy` ("Tuổi" của video)** là yếu tố quan trọng thứ hai với điểm số 0.2891. Đặc trưng này phản ánh tính mới của video. Kết

quả này củng cố giả thuyết rằng các video mới hơn có xu hướng được ưu tiên và có khả năng trending lâu hơn.

- Các đặc trưng còn lại như **`title_length` (độ dài tiêu đề)**, **`views` (lượt xem)**, và **`likes` (lượt thích)** cũng có đóng góp vào mô hình, nhưng với mức độ ảnh hưởng thấp hơn đáng kể.

Kết luận quan trọng: Phân tích này mang lại một insight giá trị: **Thể loại** và **độ mới** của video là hai yếu tố có tác động quyết định đến khả năng trending, thậm chí còn quan trọng hơn cả các chỉ số tương tác trực tiếp như lượt xem và lượt thích.

CHƯƠNG 6. TRIỂN KHAI ỨNG DỤNG

Giai đoạn triển khai là bước đưa các kết quả phân tích và mô hình dự đoán từ môi trường nghiên cứu ra thành một sản phẩm phần mềm hữu hình. Hệ thống được xây dựng theo kiến trúc client-server hiện đại, bao gồm một backend API hiệu năng cao và một frontend tương tác, linh hoạt.

6.1. Xây dựng Backend API với FastAPI

Backend đóng vai trò là "bộ não" của tầng ứng dụng, chịu trách nhiệm xử lý các logic nghiệp vụ, giao tiếp với cơ sở dữ liệu và phục vụ các mô hình học máy.

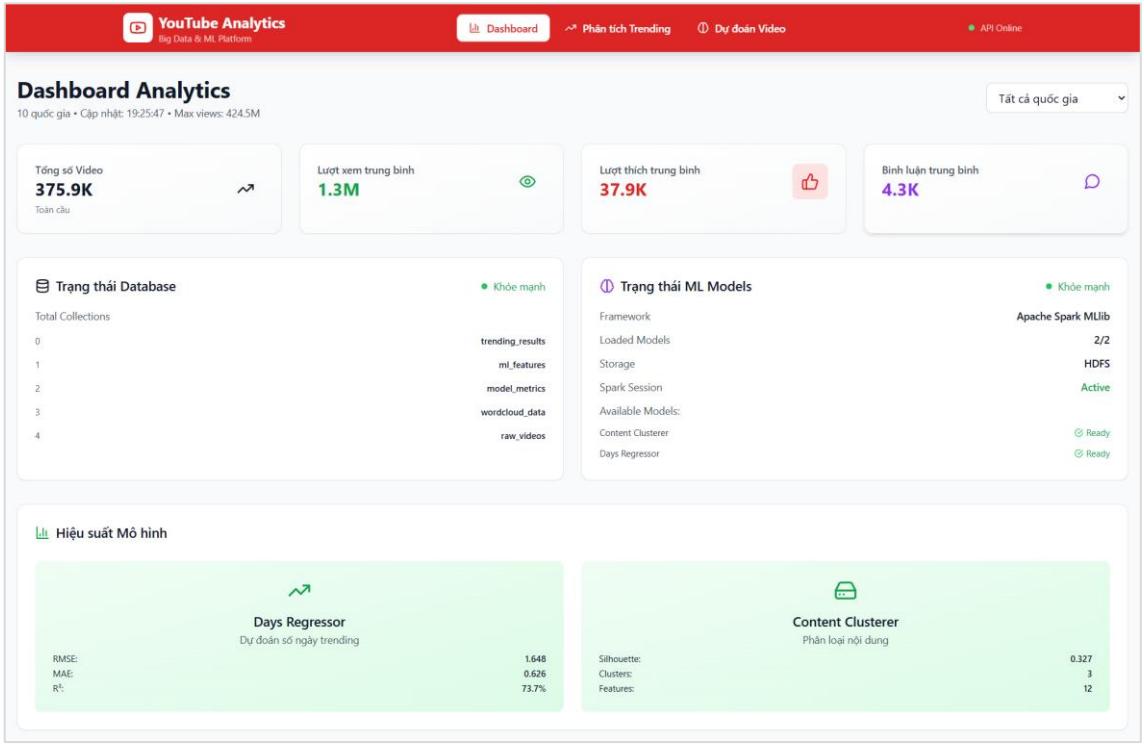
- **Lựa chọn công nghệ:** FastAPI được chọn làm framework để xây dựng backend nhờ các ưu điểm vượt trội:
    - **Hiệu năng cao:** Dựa trên Starlette và Pydantic, FastAPI cung cấp hiệu suất tương đương với NodeJS và Go.
    - **Phát triển nhanh:** Cú pháp gọn gàng, hiện đại của Python giúp giảm thời gian code.
    - **Tự động sinh tài liệu:** Tự động tạo tài liệu API tương tác (sử dụng Swagger UI và ReDoc), giúp việc kiểm thử và tích hợp trở nên dễ dàng.
- **Thiết kế các API Endpoint:** Các endpoint được thiết kế theo chuẩn RESTful để phục vụ các chức năng cụ thể từ frontend:
    - **`GET /api/trending`**: Cung cấp dữ liệu video thịnh hành đã được xử lý, hỗ trợ các tham số lọc (query parameters) theo quốc gia, ngày tháng và thể loại.
    - **`GET /api/statistics`**, **`/api/filters`**: Cung cấp các dữ liệu tổng hợp (như Word Cloud) và các tùy chọn có sẵn cho bộ lọc (danh sách quốc gia, thể loại).

- `GET /api/health`: Kiểm tra trạng thái "sức khỏe" của hệ thống, bao gồm kết nối tới database và trạng thái của các mô hình.
- `POST /api/ml/predict-trending`: Endpoint nhận dữ liệu đầu vào của một video từ người dùng và trả về kết quả dự đoán từ mô hình Random Forest.

6.2. Xây dựng Giao diện người dùng (Frontend) với React

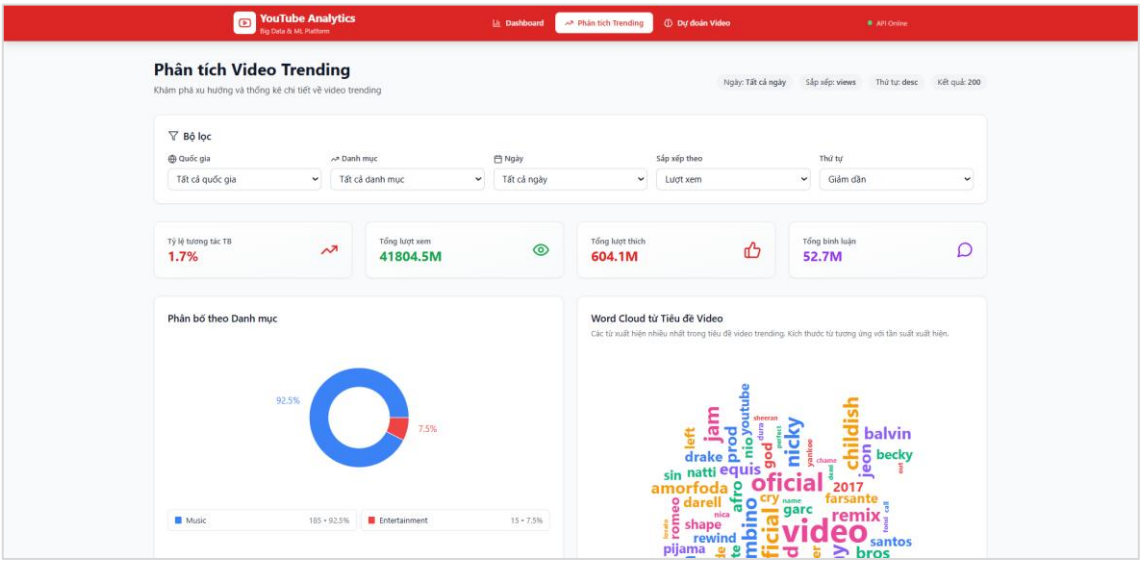
Frontend là bộ mặt của sản phẩm, được xây dựng bằng thư viện React để tạo ra một ứng dụng trang đơn (Single Page Application - SPA) mượt mà và có trải nghiệm người dùng tốt.

- **Trang Bảng điều khiển (Dashboard):** Là trang chủ của ứng dụng, cung cấp cái nhìn tổng quan về trạng thái của hệ thống. Trang này hiển thị các thông tin như trạng thái kết nối tới API, trạng thái cơ sở dữ liệu và các chỉ số hiệu suất chính của mô hình hồi quy



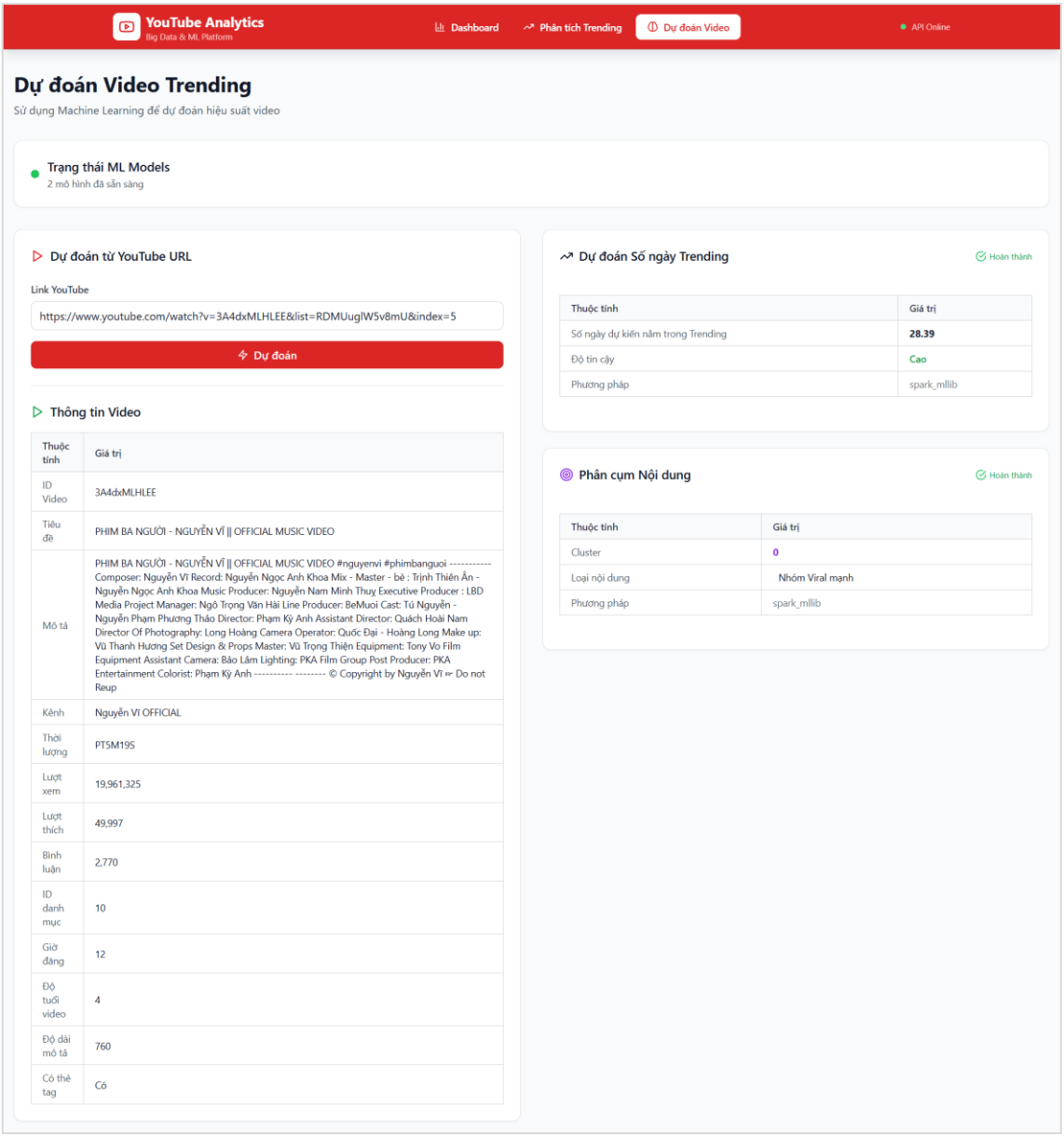
Hình 10. Màn hình trang Dashboard

- **Trang Phân tích Xu hướng (Trending Analysis):** Đây là trang chức năng chính cho việc khám phá dữ liệu.
    - **Bộ lọc đa dạng:** Người dùng có thể dễ dàng lọc và tìm kiếm dữ liệu thông qua các bộ lọc theo quốc gia, ngày bắt đầu, ngày kết thúc và thể loại.
    - **Trực quan hóa tương tác:** Dữ liệu sau khi lọc được hiển thị trực quan qua các biểu đồ (Word Cloud, Biểu đồ tròn, Biểu đồ phân tán) và một bảng dữ liệu chi tiết có hỗ trợ phân trang, giúp người dùng dễ dàng theo dõi.



Hình 11. Màn hình trang Phân tích Xu hướng

- **Trang Dự đoán (Prediction):** Trang này cho phép người dùng tương tác trực tiếp với mô hình học máy.
    - **Form nhập liệu:** Cung cấp một form để người dùng nhập các thông số của một video giả định (lượt xem, lượt thích, tiêu đề, thể loại...).
    - **Hiển thị kết quả:** Sau khi gửi, frontend sẽ gọi API dự đoán và hiển thị kết quả trả về, bao gồm số ngày dự kiến trên top thịnh hành và các gợi ý cải thiện được tạo ra bởi backend.



Hình 12. Màn hình trang Dự đoán

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

1. Kết luận và tóm tắt các kết quả đạt được

Đồ án "Ứng dụng phân tích và dự đoán xu hướng video YouTube" đã được thực hiện thành công, đáp ứng đầy đủ các mục tiêu đề ra ban đầu. Qua quá trình thực hiện, đồ án đã đạt được những kết quả quan trọng sau:

- **Về mặt kỹ thuật:**
    - Đã xây dựng thành công một quy trình xử lý dữ liệu lớn (Big Data pipeline) hoàn chỉnh, từ khâu thu thập, tiền xử lý bằng Apache Spark đến lưu trữ trên MongoDB. Hệ thống đã chứng minh được khả năng xử lý hiệu quả một tập dữ liệu lớn và phức tạp.
    - Đã xây dựng và đánh giá thành công mô hình học máy (Random Forest Regression) với hiệu suất tốt (R^2 đạt ~0.737), cho thấy khả năng dự đoán chính xác số ngày một video có thể duy trì trên top thịnh hành.
    - Đã triển khai thành công một ứng dụng web end-to-end sử dụng kiến trúc hiện đại (FastAPI cho backend, React cho frontend), cung cấp một giao diện trực quan và hữu ích cho người dùng cuối.
- **Về mặt phân tích:**
    - Qua phân tích dữ liệu khám phá, đồ án đã chỉ ra các xu hướng nổi bật trên YouTube, như sự thống trị của thể loại "Giải trí" và "Âm nhạc", cũng như các từ khóa thường xuất hiện trong tiêu đề video thịnh hành.
    - Phân tích độ quan trọng của đặc trưng đã mang lại một phát hiện giá trị: thể loại (`category_id`) và tính mới của video (`video_age_proxy`) là hai yếu tố có ảnh hưởng quyết định nhất đến khả năng trending, vượt qua cả các chỉ số tương tác trực tiếp như lượt xem hay lượt thích.

Đồ án không chỉ là một bài tập kỹ thuật mà còn là một minh chứng thực tế về việc ứng dụng các công nghệ Dữ liệu lớn để giải quyết một bài toán có ý nghĩa thực tiễn, cung cấp các insight giá trị cho các nhà sáng tạo nội dung và nhà tiếp thị.

2. Hạn chế của đồ án

Bên cạnh những kết quả đạt được, đồ án vẫn còn một số hạn chế nhất định:

- **Dữ liệu tĩnh:** Hệ thống hiện đang xử lý dữ liệu lịch sử (dữ liệu theo lô). Việc này chưa phản ánh được các xu hướng đang diễn ra theo thời gian thực.
- **Phạm vi đặc trưng:** Các mô hình dự đoán được xây dựng dựa trên các siêu dữ liệu có sẵn. Đồ án chưa khai thác các nguồn dữ liệu phi cấu trúc phức tạp hơn như phân tích nội dung hình ảnh, âm thanh của video hay phân tích cảm xúc từ bình luận.
- **Triển khai đơn giản:** Hệ thống hiện được triển khai trên môi trường cục bộ (local). Để phục vụ lượng người dùng lớn, hệ thống cần được triển khai trên một môi trường điện toán đám mây với khả năng co giãn tốt hơn.

3. Hướng phát triển trong tương lai

Từ những hạn chế trên, có thể đề xuất một số hướng phát triển và nâng cấp cho đồ án trong tương lai:

- **Xây dựng hệ thống thời gian thực:** Nâng cấp hệ thống bằng cách tích hợp các công nghệ xử lý luồng (stream processing) như Apache Kafka và Spark Streaming để có thể phân tích và cảnh báo các xu hướng ngay khi chúng xuất hiện.
- **Làm giàu đặc trưng (Feature Enrichment):**

    - Sử dụng các API xử lý ngôn ngữ tự nhiên (NLP) nâng cao để phân tích cảm xúc (sentiment analysis) từ các bình luận, từ đó có thêm đặc trưng về sự đón nhận của khán giả.
    - Tích hợp các mô hình nhận dạng hình ảnh hoặc âm thanh để phân loại nội dung video một cách chi tiết hơn.
- **Tối ưu và triển khai trên đám mây:** Đóng gói các thành phần của ứng dụng (backend, frontend, database) bằng Docker và triển khai trên các nền tảng đám mây như AWS, Google Cloud, hoặc Azure sử dụng các dịch vụ như Kubernetes để đảm bảo tính sẵn sàng và khả năng mở rộng.
- **Mở rộng mô hình dự đoán:** Xây dựng thêm các mô hình dự đoán khác như dự đoán lượt xem, lượt thích mà một video có thể đạt được trong 24 giờ đầu tiên, hoặc phân loại video "viral".

TÀI LIỆU THAM KHẢO

Sách và Giáo trình:

1. Holden Karau, Andy Konwinski, Patrick Wendell, Matei Zaharia. (2015). *Learning Spark: Lightning-Fast Big Data Analysis. O'Reilly Media*.
2. Tom White. (2015). *Hadoop: The Definitive Guide. O'Reilly Media*.
3. Aurélien Géron. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow. O'Reilly Media*.
4. Wes McKinney. (2017). *Python for Data Analysis*. O'Reilly Media.

Tài liệu kỹ thuật và Trang web:

5. **Apache Spark Documentation.** Truy cập tại: https://spark.apache.org/docs/latest/
6. **Hadoop Documentation.** Truy cập tại: https://hadoop.apache.org/docs/current/
7. **FastAPI Documentation.** Truy cập tại: https://fastapi.tiangolo.com/
8. **React Documentation.** Truy cập tại: https://reactjs.org/docs/getting-started.html
9. **MongoDB Documentation.** Truy cập tại: https://docs.mongodb.com/
10. **Kaggle - YouTube Trending Video Dataset.** Truy cập tại: https://www.kaggle.com/datasets/datasnaek/youtube-new

Bài báo khoa học:

11. Cheng, X., Dale, C., & Liu, J. (2008). *Statistics and social network of YouTube videos*. Quality of Service. IWQoS 2008. 16th International Workshop on.

12. Borghol, Y., et al. (2011). *Characterizing and modeling the dynamics of online popularity*. Proceedings of the first workshop on Online social networks.

PHỤ LỤC

Phụ lục A: Hướng dẫn cài đặt và chạy hệ thống

1. Chạy setup: python setup.py
2. Khởi động infrastructure: python run.py infrastructure
3. Chạy pipeline: python run.py pipeline
4. Khởi động app: python run.py app
5. Truy cập: Frontend http://localhost:3000, API http://localhost:8000

Phụ lục B: Mã nguồn chính

- run.py: Script chính để chạy hệ thống
- setup.py: Script cài đặt
- spark/train_models.py: Huấn luyện ML
- spark/jobs/process_trending.py: Xử lý dữ liệu
- backend/app/main.py: Backend API
- frontend/src/App.jsx: Frontend chính