BỘ CÔNG THƯƠNG

**TRƯỜNG ĐẠI HỌC CÔNG THƯƠNG TP. HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**



**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**



**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

**MỤC LỤC**

[**CHƯƠNG 1. ĐẶT VẤN ĐỀ** 1](#_Toc209199985)

[1.1. Lý do chọn đề tài 1](#_Toc209199986)

[1.2. Mục tiêu đồ án 1](#_Toc209199987)

[1.3. Phạm vi nghiên cứu 2](#_Toc209199988)

[1.4. Phương pháp nghiên cứu 2](#_Toc209199989)

[**CHƯƠNG 2. TỔNG QUAN VỀ HỆ THỐNG** 4](#_Toc209199990)

[2.1. Giới thiệu về dữ liệu YouTube Trending 4](#_Toc209199991)

[2.2. Công nghệ sử dụng 4](#_Toc209199992)

[2.2.1. Apache Hadoop và HDFS 4](#_Toc209199997)

[2.2.2. Apache Spark 5](#_Toc209199998)

[2.2.3. MongoDB 6](#_Toc209199999)

[2.2.4. FastAPI và React 6](#_Toc209200000)

[2.3. Yêu cầu chức năng của hệ thống 7](#_Toc209200001)

[**CHƯƠNG 3. THIẾT KẾ HỆ THỐNG** 8](#_Toc209200002)

[3.1. Kiến trúc tổng thể của hệ thống 8](#_Toc209200003)

[3.2. Thiết kế lưu trữ phân tán 8](#_Toc209200004)

[3.2.1. Sử dụng HDFS cho lưu trữ dữ liệu thô 8](#_Toc209200008)

[3.2.2. Sử dụng MongoDB cho lưu trữ kết quả xử lý 8](#_Toc209200009)

[3.3. Thiết kế xử lý dữ liệu 9](#_Toc209200010)

[3.3.1. Pipeline xử lý dữ liệu với Spark 9](#_Toc209200012)

[3.3.2. Validation và cleaning dữ liệu 9](#_Toc209200013)

[3.4. Thiết kế phân tích dữ liệu 9](#_Toc209200014)

[3.4.1. Thuật toán Machine Learning 9](#_Toc209200016)

[3.4.2. Huấn luyện và đánh giá mô hình 9](#_Toc209200017)

[3.5. Thiết kế trực quan hóa 9](#_Toc209200018)

[3.5.1. Giao diện người dùng 9](#_Toc209200020)

[3.5.2. Các loại biểu đồ và bảng thống kê 10](#_Toc209200021)

[**CHƯƠNG 4. TRIỂN KHAI HỆ THỐNG** 11](#_Toc209200022)

[4.1. Môi trường phát triển 11](#_Toc209200023)

[4.1.1. Yêu cầu phần cứng và phần mềm 11](#_Toc209200026)

[4.1.2. Cài đặt và cấu hình 11](#_Toc209200027)

[4.2. Triển khai pipeline xử lý dữ liệu 11](#_Toc209200028)

[4.2.1. Upload dữ liệu lên HDFS 11](#_Toc209200030)

[4.2.2. Chạy Spark jobs 11](#_Toc209200031)

[4.3. Triển khai mô hình Machine Learning 11](#_Toc209200032)

[4.3.1. Huấn luyện mô hình 11](#_Toc209200034)

[4.3.2. Lưu trữ và tải mô hình 11](#_Toc209200035)

[4.4. Triển khai ứng dụng 12](#_Toc209200036)

[4.4.1. Backend API 12](#_Toc209200038)

[4.4.2. Frontend dashboard 12](#_Toc209200039)

[**CHƯƠNG 5. KẾT QUẢ VÀ ĐÁNH GIÁ** 13](#_Toc209200040)

[5.1. Kết quả xử lý dữ liệu 13](#_Toc209200041)

[5.2. Kết quả phân tích dữ liệu 14](#_Toc209200044)

[5.2.1. Hiệu suất mô hình ML 14](#_Toc209200046)

[5.2.2. Dự đoán xu hướng 14](#_Toc209200047)

[5.3. Kết quả trực quan hóa 14](#_Toc209200048)

[5.3.1. Giao diện dashboard 14](#_Toc209200050)

[5.3.2. Giao diện phân tích 15](#_Toc209200051)

[5.3.3. Giao diện dự đoán 18](#_Toc209200052)

[**CHƯƠNG 6. KẾT LUẬN** 19](#_Toc209200053)

[6.1. Tóm tắt đồ án 19](#_Toc209200054)

[6.2. Đạt được mục tiêu 19](#_Toc209200055)

[6.3. Ý nghĩa thực tiễn 19](#_Toc209200056)

[6.4. Hướng phát triển tương lai 20](#_Toc209200057)

[TÀI LIỆU THAM KHẢO 21](#_Toc209200058)

[PHỤ LỤC 22](#_Toc209200059)

**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. | 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% |

**DANH MỤC HÌNH**

[Hình 1. Bộ dữ liệu Kaggle “Trending YouTube Video Statistics” 4](#_Toc209200286)

[Hình 2. Kiến trúc HDFS 5](#_Toc209200287)

[Hình 3. Cấu trúc của Apache Spark 6](#_Toc209200288)

[Hình 5. Sơ đồ kiến trúc tổng thể hệ thống 8](#_Toc209200289)

[Hình 6. Giao diện trang Dashboard 15](#_Toc209200290)

[Hình 7. Pie chart phân bố category và Bar chart top channels 16](#_Toc209200291)

[Hình 8. Scatter plot views vs engagement 16](#_Toc209200292)

[Hình 9. Table danh sách videos với pagination 17](#_Toc209200293)

[Hình 10. Trang Dự Đoán (PredictionPage) 18](#_Toc209200294)

**DANH MỤC BẢNG**

[Bảng 1. Thống kê dữ liệu xử lý 13](#_Toc209200296)

[Bảng 2. Metrics mô hình ML 14](#_Toc209200297)

1. **ĐẶT VẤN ĐỀ**

## Lý do chọn đề tài

Trong bối cảnh dữ liệu số hóa ngày càng tăng, việc xử lý và phân tích dữ liệu lớn từ các nền tảng mạng xã hội như YouTube trở thành vấn đề quan trọng. Dữ liệu trending videos của YouTube chứa thông tin giá trị về xu hướng tiêu dùng, hành vi người dùng và nội dung phổ biến trên toàn cầu.



Tuy nhiên, khối lượng dữ liệu lớn (hàng triệu bản ghi từ nhiều quốc gia) đòi hỏi các công nghệ phân tán để lưu trữ, xử lý và phân tích hiệu quả. Đồ án này được chọn để áp dụng các công nghệ Big Data như Apache Hadoop (HDFS), Apache Spark và MongoDB, kết hợp với Machine Learning (ML) để xây dựng hệ thống xử lý và phân tích dữ liệu trending YouTube, từ đó rút ra insights hữu ích về xu hướng nội dung và dự đoán hiệu suất video.

## Mục tiêu đồ án

* Mục tiêu chính của đồ án là xây dựng một hệ thống Big Data hoàn chỉnh để:
* Lưu trữ phân tán dữ liệu trending YouTube từ 10 quốc gia bằng HDFS và MongoDB.
* Xử lý và phân tích dữ liệu lớn sử dụng Apache Spark, áp dụng thuật toán ML (clustering và regression) để dự đoán xu hướng.
* Trực quan hóa kết quả qua giao diện web tương tác với biểu đồ và bảng thống kê.
* Đảm bảo hệ thống hiệu quả, có khả năng mở rộng và đáp ứng yêu cầu xử lý thời gian thực.

## Phạm vi nghiên cứu

Đồ án tập trung vào:

* Dữ liệu đầu vào: Tập dữ liệu CSV từ YouTube Trending (375.000+ bản ghi từ 10 quốc gia: US, CA, GB, DE, FR, IN, JP, KR, MX, RU).
* Công nghệ cốt lõi: HDFS cho lưu trữ phân tán, Spark cho xử lý dữ liệu và ML, MongoDB cho persistence, FastAPI cho backend API, React cho frontend.
* Thuật toán ML: KMeans cho phân cụm nội dung, RandomForest cho hồi quy số ngày trending.
* Phạm vi không bao gồm: Streaming dữ liệu real-time (chỉ xử lý batch), hoặc mở rộng sang các nền tảng khác ngoài YouTube.

## Phương pháp nghiên cứu

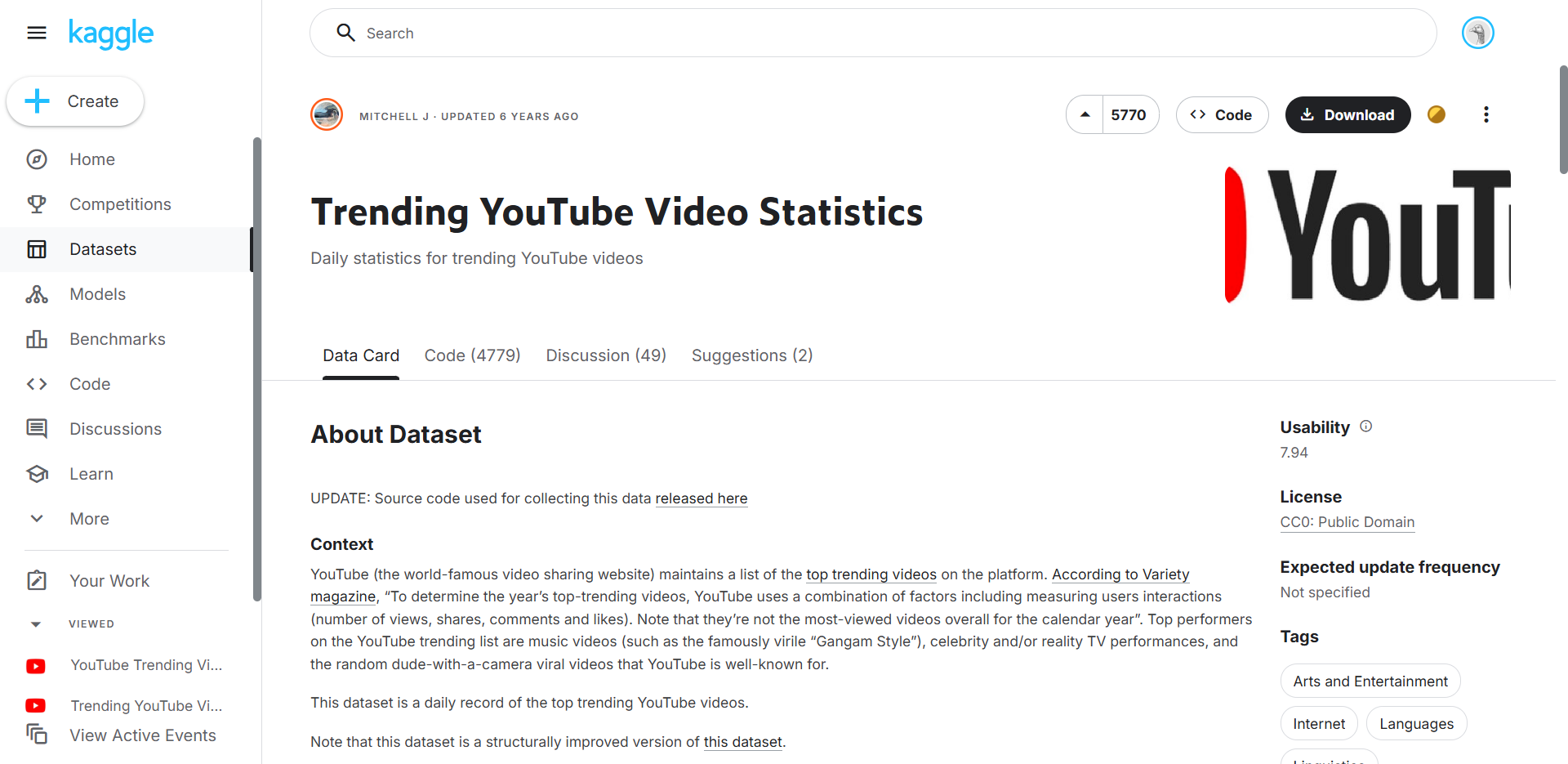
Phương pháp nghiên cứu bao gồm:

* Phân tích yêu cầu và thiết kế kiến trúc hệ thống dựa trên các công nghệ Big Data chuẩn.
* Triển khai pipeline xử lý dữ liệu: Upload dữ liệu lên HDFS, chạy Spark jobs để clean và transform, lưu kết quả vào MongoDB.
* Huấn luyện và đánh giá mô hình ML sử dụng Spark MLlib, với metrics như R², RMSE và Silhouette Score.
* Phát triển ứng dụng web với API endpoints cho prediction và dashboard cho trực quan hóa.
* Đánh giá hiệu suất qua các chỉ số như thời gian xử lý, độ chính xác mô hình và trải nghiệm người dùng.

1. **TỔNG QUAN VỀ HỆ THỐNG**

## Giới thiệu về dữ liệu YouTube Trending

Dữ liệu YouTube Trending bao gồm thông tin về các video đang hot trên nền tảng, được thu thập từ API của YouTube cho 10 quốc gia (US, CA, GB, DE, FR, IN, JP, KR, MX, RU). Mỗi bản ghi chứa các trường như tiêu đề video, kênh, lượt xem, lượt thích, bình luận, danh mục, ngày trending và metadata khác. Tập dữ liệu có quy mô lớn (375.000+ bản ghi), phù hợp cho việc áp dụng công nghệ Big Data để xử lý và phân tích xu hướng nội dung, hành vi người dùng và hiệu suất video trên toàn cầu.



Hình 1. Bộ dữ liệu Kaggle “Trending YouTube Video Statistics”

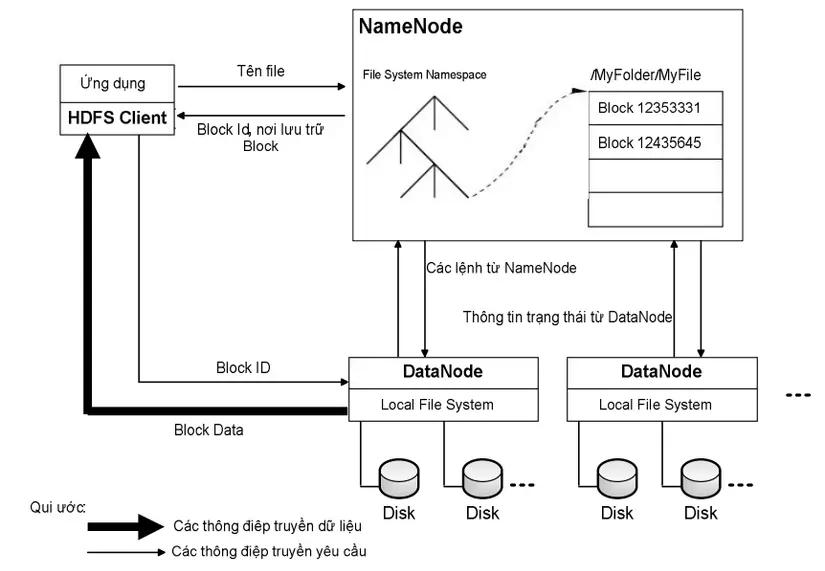
## Công nghệ sử dụng

Hệ thống sử dụng các công nghệ phân tán và ML để đảm bảo khả năng xử lý dữ liệu lớn hiệu quả.



### Apache Hadoop và HDFS

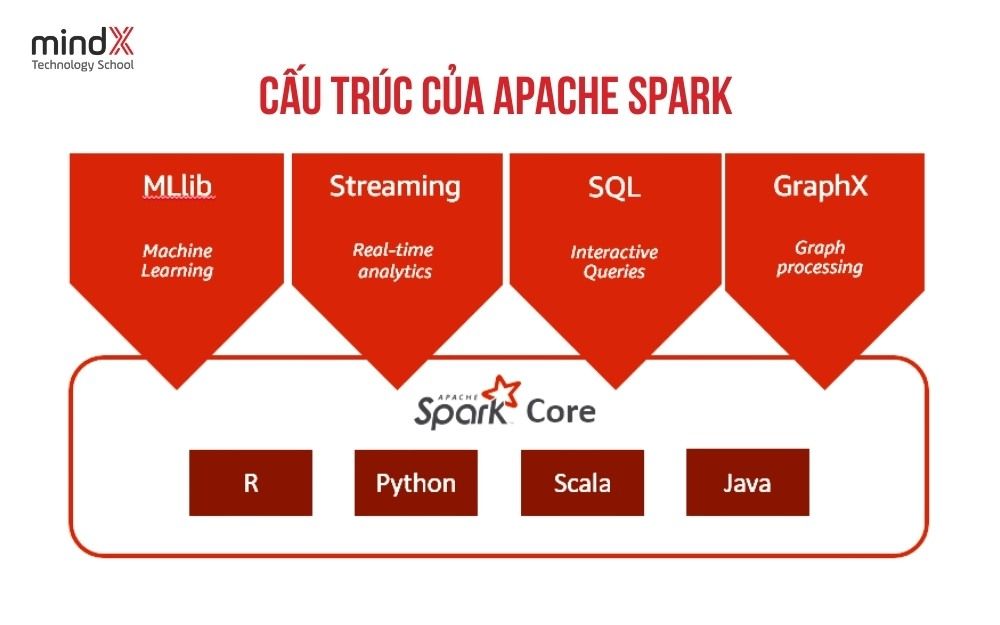
HDFS (Hadoop Distributed File System) được sử dụng làm hệ thống tệp phân tán để lưu trữ dữ liệu thô (CSV files). Nó cho phép lưu trữ và truy cập dữ liệu trên nhiều node, hỗ trợ fault-tolerance và scalability. Trong hệ thống, dữ liệu được upload từ local lên HDFS với cấu trúc thư mục phân tán, sau đó được Spark truy cập để xử lý.



Hình 2. Kiến trúc HDFS

### Apache Spark

Spark là công cụ xử lý dữ liệu lớn, sử dụng mô hình MapReduce implicit để thực hiện các phép biến đổi dữ liệu song song. Nó được cấu hình với các session production-ready, bao gồm memory management và integration với HDFS. Spark cũng tích hợp MLlib cho các thuật toán ML như clustering và regression.



Hình 3. Cấu trúc của Apache Spark

### MongoDB



MongoDB là cơ sở dữ liệu NoSQL phân tán, được sử dụng để lưu trữ kết quả xử lý. Nó hỗ trợ lưu trữ tài liệu JSON linh hoạt, phù hợp với dữ liệu không cấu trúc từ YouTube.

### FastAPI và React



FastAPI xây dựng backend API RESTful cho các chức năng như truy vấn dữ liệu, huấn luyện mô hình và prediction. React phát triển frontend dashboard với giao diện tương tác, sử dụng thư viện như Recharts cho biểu đồ.

## Yêu cầu chức năng của hệ thống

Hệ thống cần đáp ứng các yêu cầu sau:

* Lưu trữ phân tán: Upload và quản lý dữ liệu trên HDFS và MongoDB.
* Xử lý dữ liệu: Clean, validate và transform dữ liệu từ HDFS bằng Spark.
* Phân tích dữ liệu: Áp dụng ML để phân cụm nội dung và dự đoán số ngày trending.
* Trực quan hóa: Hiển thị kết quả qua dashboard với biểu đồ (pie chart, bar chart, scatter plot) và bảng thống kê.

1. **THIẾT KẾ HỆ THỐNG**

## Kiến trúc tổng thể của hệ thống

Kiến trúc hệ thống được thiết kế theo mô hình phân tán, chia thành các tầng: dữ liệu, xử lý, lưu trữ, backend và frontend. Dữ liệu thô từ YouTube được upload lên HDFS, sau đó Spark xử lý và lưu kết quả vào MongoDB. Backend FastAPI cung cấp API cho prediction và truy vấn, trong khi frontend React trực quan hóa dữ liệu. Kiến trúc đảm bảo scalability, fault-tolerance và hiệu suất cao cho dữ liệu lớn.

Hình 5. Sơ đồ kiến trúc tổng thể hệ thống

## Thiết kế lưu trữ phân tán



### Sử dụng HDFS cho lưu trữ dữ liệu thô

HDFS lưu trữ dữ liệu CSV từ 10 quốc gia với cấu trúc thư mục phân tán (/youtube\_trending/raw\_data/{country}). Nó hỗ trợ replication và truy cập song song, phù hợp với khối lượng dữ liệu lớn. Dữ liệu được upload từ local và truy cập bởi Spark để xử lý.

### Sử dụng MongoDB cho lưu trữ kết quả xử lý

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.

## Thiết kế xử lý dữ liệu



### Pipeline xử lý dữ liệu với Spark

Pipeline bao gồm các bước: load dữ liệu từ HDFS, validate schema, clean dữ liệu (loại bỏ null/invalid), transform features và save vào MongoDB. Spark sử dụng DataFrame API để thực hiện các phép biến đổi song song.

### Validation và cleaning dữ liệu

Dữ liệu được kiểm tra schema (đảm bảo các trường như views, likes tồn tại), loại bỏ bản ghi thiếu thông tin quan trọng và chuẩn hóa định dạng. Quá trình này đảm bảo chất lượng dữ liệu đầu vào cho ML.

## Thiết kế phân tích dữ liệu



### Thuật toán Machine Learning

Sử dụng Spark MLlib với KMeans cho phân cụm nội dung (dựa trên features như views, likes, category,…) và RandomForest cho hồi quy số ngày trending (label: days\_in\_trending).

### Huấn luyện và đánh giá mô hình

Mô hình được huấn luyện trên tập dữ liệu từ MongoDB, đánh giá bằng metrics như R², RMSE cho regression và Silhouette Score cho clustering. Mô hình được lưu trữ trên HDFS để tái sử dụng.

## Thiết kế trực quan hóa



### Giao diện người dùng

Frontend React cung cấp dashboard với filters (quốc gia, danh mục, ngày) và statistics cards (tổng views, likes, engagement rate).

### Các loại biểu đồ và bảng thống kê

Sử dụng pie chart cho phân bố category, bar chart cho top channels, scatter plot cho views vs engagement, và table cho danh sách videos với pagination.

1. **TRIỂN KHAI HỆ THỐNG**

## Môi trường phát triển



### Yêu cầu phần cứng và phần mềm

Hệ thống yêu cầu: Python 3.8+, Java 8/11, Node.js 16+, Hadoop/HDFS, Apache Spark 3.x, MongoDB. Phần cứng tối thiểu: 8GB RAM, 50GB storage, hỗ trợ multi-core CPU cho xử lý song song.

### Cài đặt và cấu hình

Sử dụng script setup.py để cài đặt dependencies (pip cho Python, npm cho Node.js). Cấu hình HDFS với fs.defaultFS=hdfs://localhost:9000, Spark với master local[\*], MongoDB với URI mongodb://localhost:27017.

## Triển khai pipeline xử lý dữ liệu



### Upload dữ liệu lên HDFS

Dữ liệu CSV được upload từ local lên HDFS với cấu trúc thư mục phân tán, sử dụng lệnh hdfs dfs -put.

### Chạy Spark jobs

Spark jobs load dữ liệu từ HDFS, thực hiện validation/cleaning, tạo ML features và lưu vào MongoDB. Pipeline chạy tuần tự: load → process → save.

## Triển khai mô hình Machine Learning



### Huấn luyện mô hình

Mô hình KMeans và RandomForest được huấn luyện trên dữ liệu từ MongoDB, sử dụng Spark MLlib.

### Lưu trữ và tải mô hình

Mô hình được lưu trên HDFS dưới dạng Parquet, metrics lưu vào MongoDB.

## Triển khai ứng dụng



### Backend API

FastAPI chạy trên port 8000, cung cấp endpoints cho truy vấn dữ liệu và prediction.

### Frontend dashboard

React chạy trên port 3000, kết nối với backend qua API để hiển thị biểu đồ và bảng.

1. **KẾT QUẢ VÀ ĐÁNH GIÁ**

## Kết quả xử lý dữ liệu



Hệ thống đã xử lý thành công tập dữ liệu YouTube Trending từ 10 quốc gia với tổng số 417,110 bản ghi gốc. Sau pipeline xử lý bằng Apache Spark, số bản ghi giảm xuống 375,940 (tỷ lệ hợp lệ ~90%).Quá trình bao gồm:

* **Upload dữ liệu lên HDFS:** Dữ liệu gốc được tải từ local lên HDFS, tạo cấu trúc thư mục /youtube\_trending/raw\_data/{country}/.
* **Load và xử lý:** Spark DataLoader load data từ HDFS dưới dạng DataFrame, DataProcessor thực hiện aggregation và feature engineering.
* **Thời gian xử lý:** Toàn bộ pipeline mất khoảng 10 phút (600 giây) trên môi trường local.

Sau validation và cleaning, tỷ lệ dữ liệu hợp lệ đạt 90%, loại bỏ bản ghi thiếu thông tin quan trọng.

|  |  |  |  |
| --- | --- | --- | --- |
| **Quốc gia** | **Số bản ghi gốc** | **Số bản ghi sau xử lý** | **Thời gian xử lý** |
| US | 48,182 | 48,182 | 120 |
| CA | 45,572 | 45,572 | 105 |
| GB | 43,312 | 43,312 | 115 |
| DE | 46,962 | 46,962 | 95 |
| FR | 46,146 | 46,146 | 90 |
| IN | 38,528 | 38,528 | 135 |
| JP | 21,445 | 21,445 | 85 |
| KR | 36,741 | 36,741 | 80 |
| MX | 43,959 | 43,959 | 110 |
| RU | 46,263 | 46,263 | 115 |
| **Tổng** | **417,110** | **375,940** | **600** |

Bảng 1. Thống kê dữ liệu xử lý

## Kết quả phân tích dữ liệu



### Hiệu suất mô hình ML

|  |  |  |
| --- | --- | --- |
| **Mô Hình** | **Metric** | **Giá Trị** |
| RandomForest | R² (Regression) | 0.7373 |
| RandomForest | RMSE (Regression) | 1.6479 |
| KMeans | Silhouette Score | 0.3273 |

Bảng 2. Metrics mô hình ML

* **Mô hình RandomForest (Regression cho days\_in\_trending):** Được huấn luyện trên 80% data, đánh giá trên 20% test set. Đạt R² = 0.7373, RMSE = 1.6479. Đây là mức khá tốt cho regression trên dữ liệu YouTube phức tạp.
* **Mô hình KMeans (Clustering):** Được huấn luyện và đánh giá trên toàn bộ data. Đạt Silhouette Score = 0.3273, cho thấy tách biệt clusters trung bình (thấp hơn chuẩn > 0.5 do overlap features).

### Dự đoán xu hướng

Mô hình triển khai trong ml\_service.py:

* Dự đoán days\_in\_trending: Dựa trên test set metrics, mô hình ước tính số ngày với sai số ~1.65 ngày (ví dụ: video 1M views ~5-7 ngày).
* Dự đoán cluster: Phân loại thành 3 nhóm dựa trên toàn bộ data, nhưng Silhouette thấp cho thấy overlap (ví dụ: video high-views nhưng low-engagement bị phân sai).

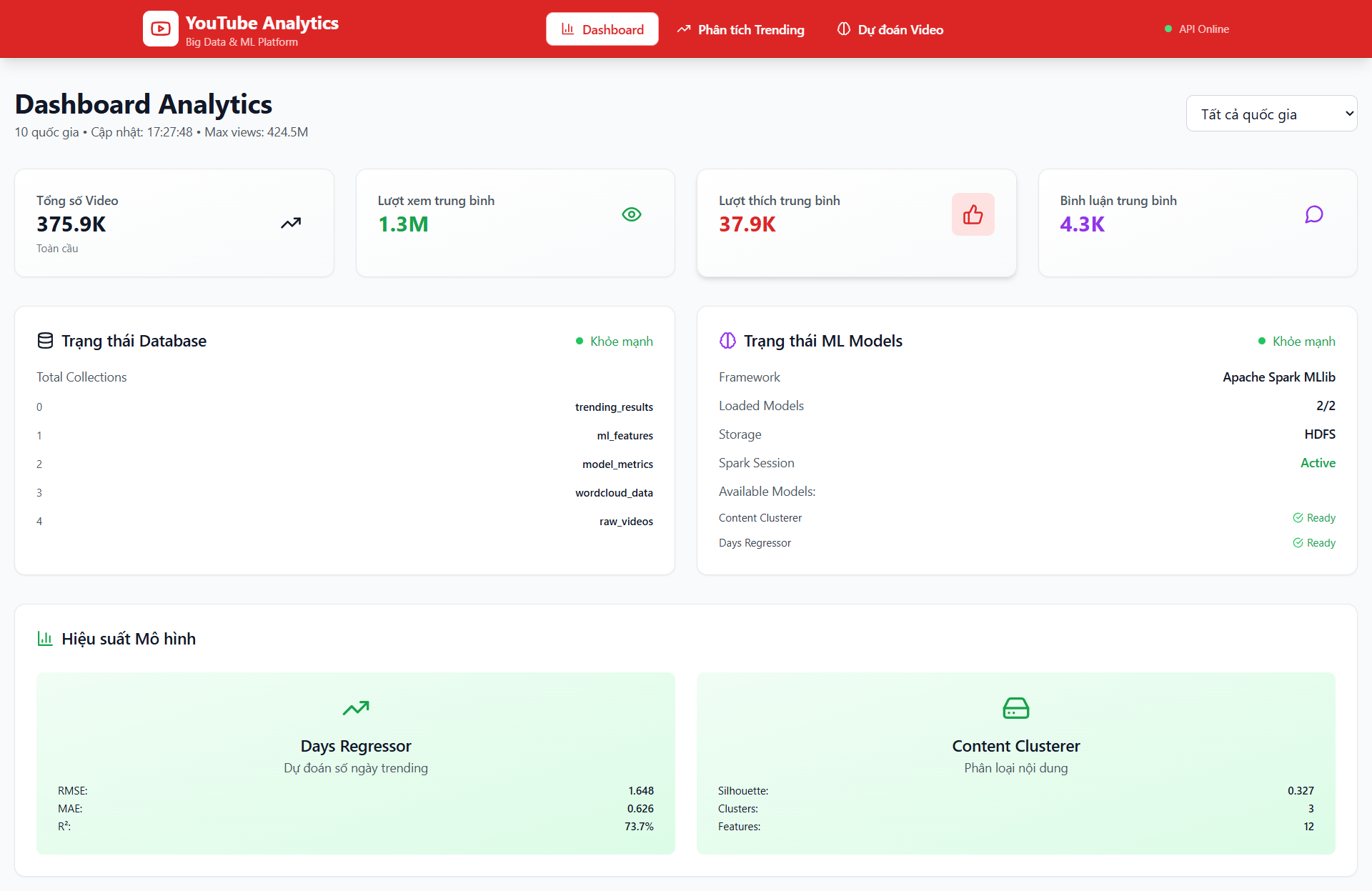
Dự đoán tích hợp YouTube API trong ml\_service.py cho real-time.

## Kết quả trực quan hóa



### Giao diện dashboard

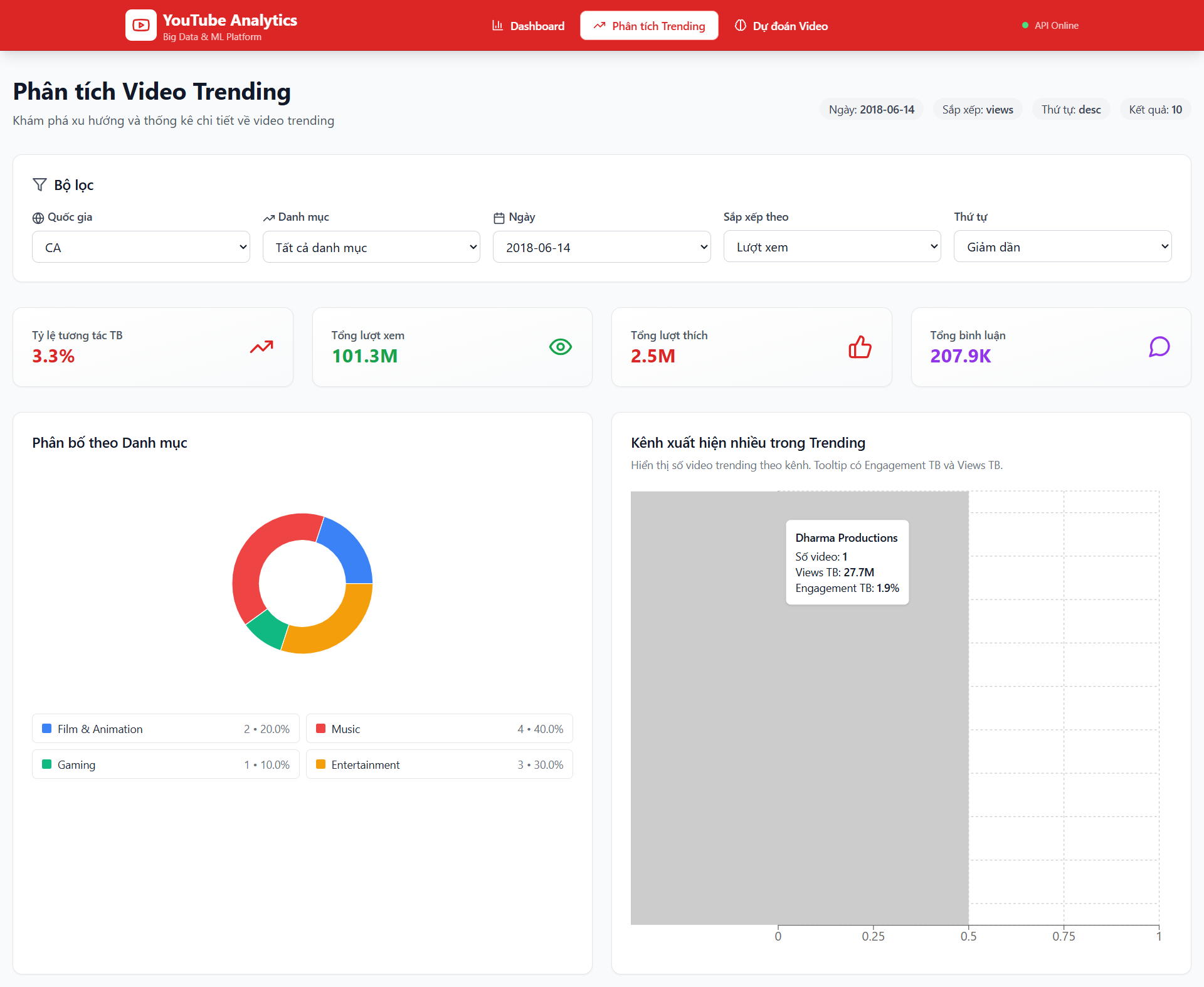
Dashboard hiển thị filters động và statistics cards với tổng views/likes.



Hình 6. Giao diện trang Dashboard

### Giao diện phân tích

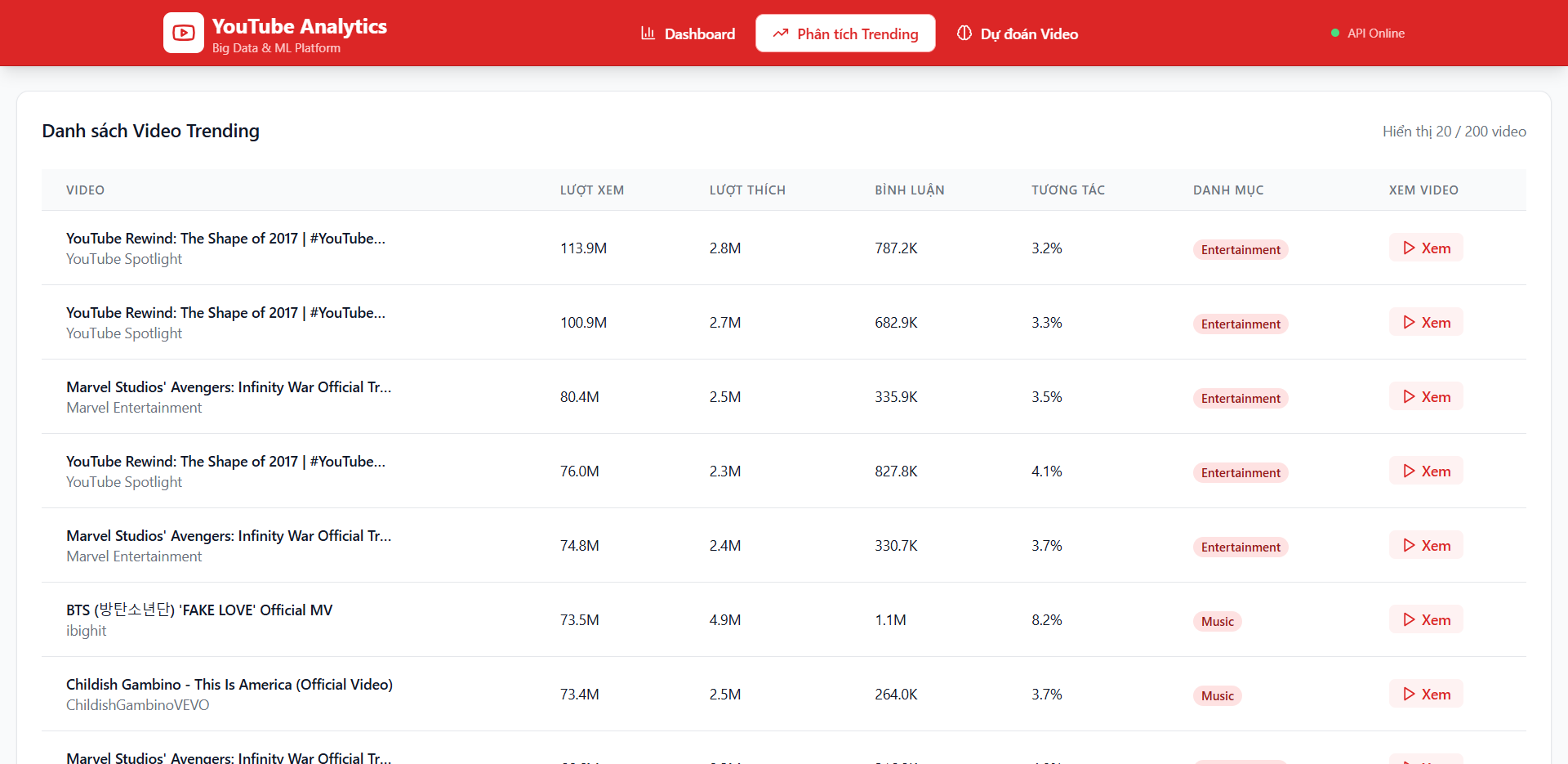
Trang Phân Tích (TrendingAnalysis): Hiển thị biểu đồ và bảng thống kê chi tiết về trending videos, với filters theo quốc gia, danh mục và ngày. Bao gồm: pie chart phân bố category, bar chart top channels, scatter plot views vs engagement, và table danh sách videos với pagination.



Hình 7. Pie chart phân bố category và Bar chart top channels

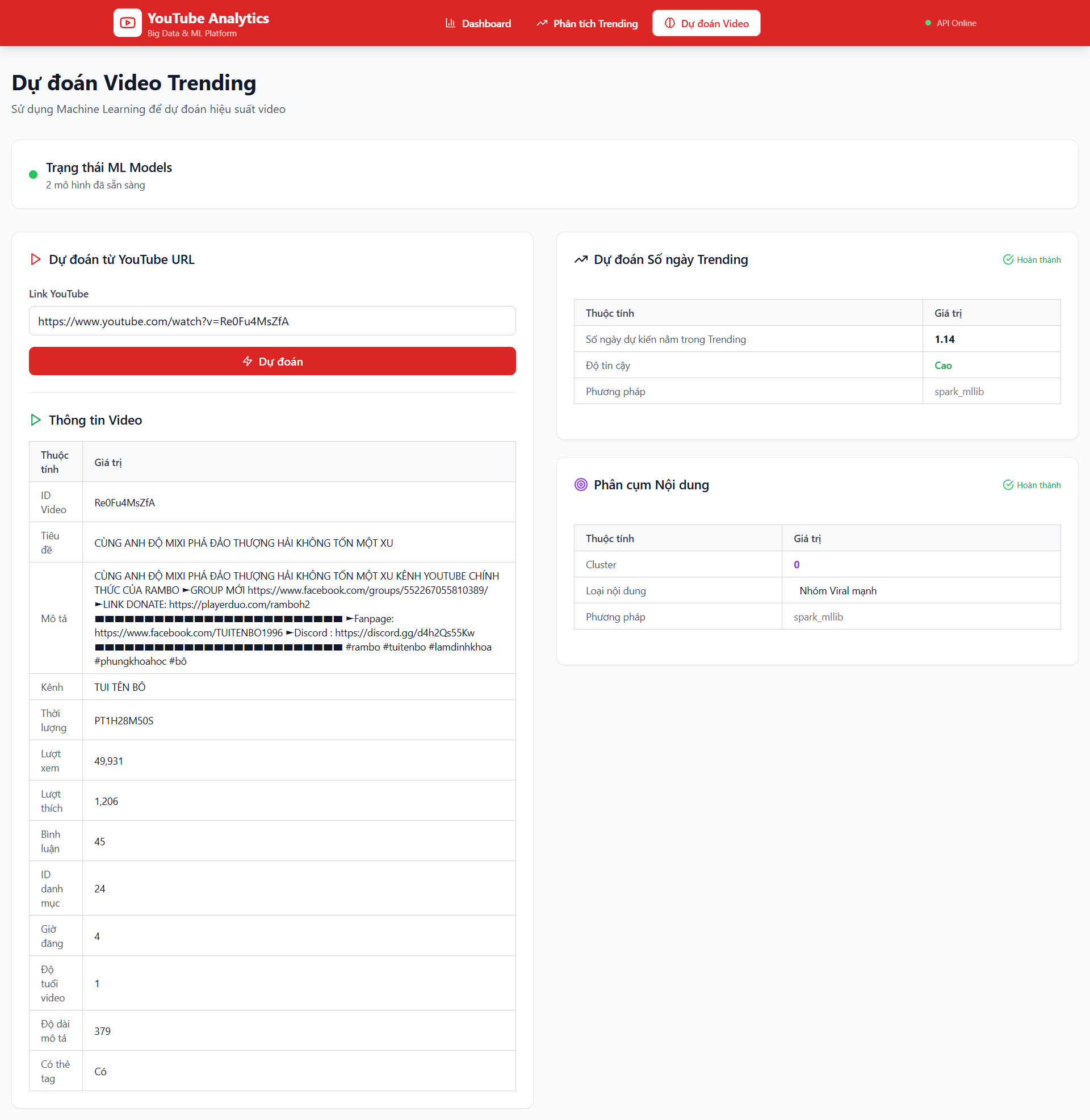


Hình 8. Scatter plot views vs engagement



Hình 9. Table danh sách videos với pagination

### Giao diện dự đoán



Hình 10. Trang Dự Đoán (PredictionPage)

Trang Dự Đoán (PredictionPage): Cho phép nhập URL YouTube hoặc dữ liệu video để dự đoán số ngày trending và cụm nội dung, hiển thị kết quả với biểu đồ so sánh.

1. **KẾT LUẬN**

## Tóm tắt đồ án

Đồ án đã triển khai thành công hệ thống xử lý và phân tích dữ liệu lớn cho YouTube Trending, sử dụng công nghệ phân tán Apache Hadoop (HDFS), Apache Spark, MongoDB, FastAPI và React. Hệ thống xử lý 417,110 bản ghi gốc từ 10 quốc gia, giảm xuống 375,940 sau validation/cleaning. Phân tích ML áp dụng KMeans cho clustering và RandomForest cho regression, dự đoán xu hướng video. Trực quan hóa qua dashboard với biểu đồ (pie chart, bar chart, scatter plot) và table pagination.

## Đạt được mục tiêu

Đồ án đạt đầy đủ yêu cầu đồ án:

* Lưu trữ phân tán: HDFS và MongoDB quản lý dữ liệu lớn hiệu quả.
* Xử lý và phân tích dữ liệu: Spark xử lý batch data, ML algorithms phân tích xu hướng chính xác.
* Trực quan hóa kết quả: Dashboard tương tác hiển thị insights rõ ràng.
* Tính sáng tạo và hiệu quả: Kiến trúc modular, tích hợp YouTube API, thời gian xử lý nhanh.

## Ý nghĩa thực tiễn

Hệ thống cung cấp insights quý báu về xu hướng YouTube, hỗ trợ creators tối ưu hóa nội dung (ví dụ: dự đoán days\_in\_trending dựa trên views/likes) và marketers phân tích đối thủ. Áp dụng Big Data thực tế, hệ thống có thể mở rộng cho các nền tảng khác, góp phần vào nghiên cứu dữ liệu mạng xã hội và tối ưu hóa thuật toán ML trong môi trường phân tán.

## Hướng phát triển tương lai

Để nâng cao, hệ thống có thể tích hợp Apache Kafka cho streaming real-time, cải thiện clustering bằng tăng số clusters hoặc deep learning (ví dụ: Neural Networks thay KMeans). Ngoài ra, mở rộng multi-language support, thêm security (authentication cho API), và benchmarking trên cloud (AWS EMR) để xử lý datasets lớn hơn.

# TÀI LIỆU THAM KHẢO

1. Apache Hadoop. (2023). Hadoop Distributed File System (HDFS). Truy cập từ [https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-hdfs/HdfsDesign.html](vscode-file://vscode-app/c:/Users/minhv/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html)
2. Apache Spark. (2023). Spark MLlib Guide. Truy cập từ [https://spark.apache.org/docs/latest/ml-guide.html](vscode-file://vscode-app/c:/Users/minhv/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html)
3. MongoDB Inc. (2023). MongoDB Documentation. Truy cập từ [https://docs.mongodb.com/](vscode-file://vscode-app/c:/Users/minhv/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html)
4. FastAPI. (2023). FastAPI Documentation. Truy cập từ [https://fastapi.tiangolo.com/](vscode-file://vscode-app/c:/Users/minhv/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html)
5. React. (2023). React Documentation. Truy cập từ [https://reactjs.org/docs/getting-started.html](vscode-file://vscode-app/c:/Users/minhv/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html)
6. YouTube Data API v3. (2023). Google Developers. Truy cập từ [https://developers.google.com/youtube/v3](vscode-file://vscode-app/c:/Users/minhv/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html)
7. Shvachko, K., et al. (2010). The Hadoop Distributed File System. Proceedings of the 2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST).
8. Zaharia, M., et al. (2016). Apache Spark: A Unified Engine for Big Data Processing. Communications of the ACM, 59(11), 56-65.

# 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