TRƯỜNG ĐẠI HỌC ĐẠI NAM

KHOA CÔNG NGHỆ THÔNG TIN

BÀI TẬP LỚN

HỌC PHẦN: DỮ LIỆU LỚN

ĐỀ TÀI: DỰ ĐOÁN XU HƯỚNG VIDEO YOUTUBE THỊNH HÀNH (113 QUỐC GIA)

Giảng viên: ThS. Lê Thị Thùy Trang

ThS. Trần Quý Nam

STT	Mã sv	Họ và Tên	Ngày Sinh	Lớp
1	1671020302	Nguyễn Tất Thắng	05/12/2004	CNTT 16 - 02
2	1671020173	Bùi Tuấn Kiệt	23/05/2004	CNTT 16 - 02
3	1671020078	Nguyễn Đức Đại	26/01/2003	CNTT 16 - 02
4	1671020282	Nguyễn Văn Tân	20/08/2004	CNTT 16 - 02

Hà Nội, năm 2025

TRƯỜNG ĐẠI HỌC ĐẠI NAM

KHOA CÔNG NGHỆ THÔNG TIN

BÀI TẬP LỚN

HỌC PHẦN: DỮ LIỆU LỚN

ĐỀ TÀI: DỰ ĐOÁN XU HƯỚNG VIDEO YOUTUBE THỊNH HÀNH (113 QUỐC GIA)

STT	Mã sv	Họ và Tên	Ngày Sinh	Lớp
1	1671020302	Nguyễn Tất Thắng	05/12/2004	CNTT 16 - 02
2	1671020173	Bùi Tuấn Kiệt	23/05/2004	CNTT 16 - 02
3	1671020078	Nguyễn Đức Đại	26/01/2003	CNTT 16 - 02
4	1671020282	Nguyễn Văn Tân	20/08/2004	CNTT 16 - 02

CÁN BỘ CHẤM THI 1 **CÁN BỘ CHẤM THI 2**

Trần Quý Nam **Lê Thị Thùy Trang**

Hà Nội, năm 2025

LỜI NÓI ĐẦU

Trong thời đại công nghệ 4.0, sự phát triển mạnh mẽ của Trí tuệ nhân tạo (AI) và Học máy (Machine Learning) đã mang lại những bước tiến vượt bậc trong nhiều lĩnh vực, đặc biệt là trong xử lý và nhận dạng hình ảnh. Hệ thống nhận dạng hình ảnh ngày càng được ứng dụng rộng rãi trong thực tế, từ giám sát an ninh, y tế, giao thông, đến thương mại điện tử và công nghiệp sản xuất.

Nhằm tìm hiểu và ứng dụng các thuật toán học máy trong bài toán nhận dạng hình ảnh, nhóm chúng em đã thực hiện đồ án này với mục tiêu nghiên cứu, phân tích và triển khai mô hình nhận diện hình ảnh dựa trên các thuật toán phổ biến như CNN, SVM, và các phương pháp kết hợp khác. Đồng thời, đồ án cũng sử dụng các công cụ hỗ trợ mạnh mẽ như PySpark để tối ưu hóa việc xử lý dữ liệu lớn, từ đó đánh giá hiệu suất của các mô hình trên tập dữ liệu thực tế.

Trong quá trình thực hiện, nhóm chúng em đã tìm hiểu về các phương pháp tiền xử lý dữ liệu, các kỹ thuật trích xuất đặc trưng, và so sánh hiệu quả của các mô hình nhằm đưa ra lựa chọn tối ưu nhất cho hệ thống. Kết quả của đồ án sẽ giúp làm rõ khả năng ứng dụng của học máy trong nhận dạng hình ảnh, đồng thời là cơ sở cho các nghiên cứu và ứng dụng thực tiễn sau này.

Nhóm chúng em xin gửi lời cảm ơn chân thành đến giảng viên hướng dẫn và các thầy cô trong khoa đã tận tình giúp đỡ, hỗ trợ chúng em trong suốt quá trình thực hiện đồ án. Chúng em cũng xin cảm ơn các tài liệu tham khảo và cộng đồng nghiên cứu khoa học đã cung cấp những nguồn tài nguyên quý giá để chúng em có thể hoàn thành tốt đề tài này.

Mặc dù đã cố gắng hết sức, nhưng chắc chắn không thể tránh khỏi những thiếu sót. Chúng em mong nhận được những góp ý và phản hồi để có thể cải thiện và phát triển hơn trong tương lai.

MỤC LỤC

MỤC LỤC HÌNH ẢNH

CHƯƠNG 1. TỔNG QUAN CƠ SỞ LÝ THUYẾT LIÊN QUAN

1.1 Giới thiệu

Trong bối cảnh công nghệ thông tin phát triển mạnh mẽ, việc ứng dụng các phương pháp học máy và trí tuệ nhân tạo ngày càng trở nên phổ biến trong nhiều lĩnh vực khác nhau. Các hệ thống thông minh không chỉ giúp tự động hóa quy trình, mà còn hỗ trợ ra quyết định một cách chính xác và hiệu quả hơn.

Bài tập lớn này hướng đến việc phát triển một hệ thống xử lý dữ liệu và áp dụng các mô hình học máy để giải quyết một bài toán cụ thể. Để đạt được mục tiêu này, việc hiểu rõ các khái niệm lý thuyết liên quan là vô cùng quan trọng. Trước khi đi vào chi tiết của bài toán và cách triển khai thực tế, cần có một cái nhìn tổng quan về các phương pháp học máy, các thuật toán được sử dụng, cách tiền xử lý dữ liệu và các công cụ lập trình hỗ trợ.

Chương này sẽ trình bày những kiến thức nền tảng cần thiết, giúp làm rõ cơ sở lý thuyết của bài toán. Cụ thể, nội dung bao gồm:

- Giới thiệu tổng quan về học máy và các phương pháp chính.
- Các thuật toán học máy phổ biến và ứng dụng của chúng.
- Quy trình tiền xử lý dữ liệu, một bước quan trọng để nâng cao hiệu suất mô hình.
- Các công cụ và thư viện lập trình hỗ trợ triển khai bài toán.

Những kiến thức được trình bày trong chương này sẽ đóng vai trò làm nền tảng cho việc thiết kế, triển khai và đánh giá mô hình trong các chương tiếp theo.

1.2 Học máy và các thuật toán liên quan

1.2.1 Tổng quan về học máy

Học máy (Machine Learning - ML) là một lĩnh vực thuộc trí tuệ nhân tạo (AI), trong đó các thuật toán được phát triển để máy tính có thể học từ dữ liệu và cải thiện hiệu suất mà không cần được lập trình rõ ràng. ML có thể được chia thành ba nhóm chính:

- Học có giám sát (Supervised Learning): Dữ liệu huấn luyện có nhãn rõ ràng, mô hình học cách dự đoán đầu ra dựa trên dữ liệu đầu vào. Các thuật toán phổ biến gồm có KNN, Random Forest, SVM, và mạng nơ-ron nhân tạo.

- Học không giám sát (Unsupervised Learning): Dữ liệu huấn luyện không có nhãn, mô hình cố gắng tìm ra cấu trúc ẩn trong dữ liệu. Các thuật toán phổ biến gồm K-Means, PCA, và Hierarchical Clustering.
- Học tăng cường (Reinforcement Learning - RL): Mô hình học thông qua tương tác với môi trường và nhận phản hồi dưới dạng phần thưởng.

So sánh các phương pháp học máy

Loại học máy	Đặc điểm chính	Ứng dụng phổ biến
Học có giám sát	Dữ liệu có nhãn, mô hình học từ dữ liệu đầu vào - đầu ra	Nhận diện khuôn mặt, dự đoán giá nhà, phân loại email
Học không giám sát	Dữ liệu không có nhãn, tìm kiếm mẫu ẩn trong dữ liệu	Phân cụm khách hàng, giảm chiều dữ liệu, phân tích thị trường
Học tăng cường	Học từ phản hồi, tối ưu hóa chiến lược thông qua thử nghiệm	Trí tuệ nhân tạo chơi game, tối ưu hóa robot, giao dịch tài chính

Ứng dụng thực tế của học máy

- **Y tế**: Chẩn đoán bệnh từ hình ảnh y khoa, dự đoán khả năng tái nhập viện.
- **Tài chính**: Phát hiện gian lận thẻ tín dụng, dự báo thị trường chứng khoán.
- **Thương mại điện tử**: Hệ thống gợi ý sản phẩm, phân tích hành vi khách hàng.
- **Giao thông**: Dự đoán lưu lượng giao thông, xe tự hành.

1.2.2 Các thuật toán sử dụng trong bài toán

K-Nearest Neighbors (KNN)

K-Nearest Neighbors (KNN) là một thuật toán phân loại và hồi quy đơn giản nhưng hiệu quả, hoạt động dựa trên nguyên tắc láng giềng gần nhất. Cách thức hoạt động của KNN như sau:

- Khi nhận một điểm dữ liệu mới, thuật toán sẽ tính khoảng cách giữa điểm đó với tất cả các điểm trong tập huấn luyện.
- Chọn ra k điểm gần nhất dựa trên một tiêu chí đo khoảng cách (thường sử dụng khoảng cách Euclidean, Manhattan hoặc Minkowski).
- Với bài toán phân loại, KNN sẽ xem xét nhãn của k điểm gần nhất và chọn nhãn xuất hiện nhiều nhất làm dự đoán cho điểm dữ liệu mới.
- Với bài toán hồi quy, KNN sẽ lấy trung bình giá trị đầu ra của k điểm gần nhất để dự đoán.

Ưu điểm của KNN:

- Dễ hiểu, dễ triển khai.
- Không cần giai đoạn huấn luyện phức tạp.
- Hiệu quả đối với dữ liệu có biên tách lớp rõ ràng.

Nhược điểm của KNN:

- Tốn nhiều tài nguyên khi dự đoán do phải tính toán khoảng cách với tất cả điểm dữ liệu huấn luyện.
- Hiệu suất kém với dữ liệu có số lượng lớn hoặc có nhiều nhiễu.
- Nhạy cảm với giá trị ngoại lai.

Ứng dụng của KNN:

- Phân loại hình ảnh và văn bản.
- Nhận diện chữ viết tay.
- Hệ thống gợi ý sản phẩm.

Random Forest

Random Forest là một thuật toán thuộc nhóm học có giám sát, hoạt động dựa trên phương pháp ensemble learning (học tập tổ hợp). Thuật toán này sử dụng một tập hợp nhiều cây quyết định (Decision Trees) để tăng cường độ chính xác và giảm nguy cơ quá khớp (overfitting).

Cách hoạt động của Random Forest:

- Thuật toán tạo ra nhiều cây quyết định từ các tập con dữ liệu được chọn ngẫu nhiên (bagging).
- Mỗi cây quyết định sẽ đưa ra một dự đoán.
- Với bài toán phân loại, thuật toán chọn kết quả theo phương pháp bỏ phiếu đa số (majority voting).
- Với bài toán hồi quy, thuật toán lấy trung bình dự đoán từ tất cả các cây.

Ưu điểm của Random Forest:

- Chống quá khớp tốt: Nhờ việc kết hợp nhiều cây quyết định, mô hình ít bị ảnh hưởng bởi nhiễu.
- Độ chính xác cao: Hoạt động hiệu quả trên nhiều loại dữ liệu khác nhau.
- Tính linh hoạt cao: Áp dụng được cho cả bài toán phân loại và hồi quy.

Nhược điểm của Random Forest:

- Chi phí tính toán cao: Khi số lượng cây trong rừng lớn, việc tính toán có thể chậm.
- Khó diễn giải: Không dễ dàng hiểu được cách thuật toán đưa ra quyết định như trong cây quyết định đơn giản.

Ứng dụng của Random Forest:

- Y tế: Chẩn đoán bệnh từ dữ liệu y khoa.
- Tài chính: Dự báo rủi ro tín dụng, phát hiện gian lận.
- Thương mại điện tử: Phân loại khách hàng, cá nhân hóa đề xuất sản phẩm.
- Môi trường: Dự báo chất lượng không khí, phát hiện cháy rừng.

1.3 Tiền xử lý dữ liệu

1.3.1 Vai trò của tiền xử lý dữ liệu

Dữ liệu trong thực tế thường không hoàn hảo, có thể chứa giá trị thiếu, giá trị ngoại lai hoặc định dạng không đồng nhất. Việc tiền xử lý dữ liệu đóng vai trò quan trọng trong việc chuẩn bị dữ liệu để mô hình học máy có thể hoạt động tốt nhất. Nếu dữ liệu không được làm sạch và chuẩn hóa đúng cách, hiệu suất của mô hình có thể bị ảnh hưởng đáng kể.

Tiền xử lý dữ liệu giúp:

Loại bỏ nhiễu: Dữ liệu chứa các giá trị không hợp lệ có thể làm giảm độ chính xác của mô hình.

Chuẩn hóa dữ liệu: Giúp các đặc trưng có cùng thang đo, giúp thuật toán hoạt động hiệu quả hơn.

Giảm chiều dữ liệu: Giúp loại bỏ những đặc trưng không quan trọng, giảm tải tính toán.

Xử lý giá trị thiếu: Giúp dữ liệu đầy đủ hơn, tránh làm sai lệch mô hình học máy.

1.3.2 Các phương pháp tiền xử lý dữ liệu

1. Xử lý giá trị thiếu

Giá trị thiếu (missing values) có thể gây ra lỗi hoặc làm sai lệch kết quả của mô hình học máy. Một số phương pháp xử lý giá trị thiếu phổ biến:

- Xóa bỏ dữ liệu có giá trị thiếu: Phương pháp này chỉ nên áp dụng khi số lượng dữ liệu bị thiếu là rất nhỏ.
- Điền giá trị trung bình/median/mod: Với dữ liệu số, có thể điền giá trị trung bình (mean) hoặc trung vị (median). Với dữ liệu phân loại, có thể điền giá trị xuất hiện nhiều nhất (mode).
- Sử dụng thuật toán để dự đoán giá trị thiếu: Áp dụng mô hình hồi quy hoặc các thuật toán học máy để dự đoán giá trị bị thiếu dựa trên các đặc trưng khác.

2 Chuẩn hóa và chuẩn chỉnh dữ liệu

Dữ liệu có thể có các đơn vị đo lường khác nhau, gây ảnh hưởng đến hiệu suất của mô hình. Các phương pháp chuẩn hóa phổ biến:

- Min-Max Scaling: Đưa dữ liệu về khoảng [0,1].
- Z-score Normalization: Biến đổi dữ liệu thành phân phối chuẩn với giá trị trung bình bằng 0 và độ lệch chuẩn bằng 1.
- Log Transformation: Sử dụng log để biến đổi dữ liệu có phân phối lệch.

3 Xử lý giá trị ngoại lai

Giá trị ngoại lai có thể ảnh hưởng lớn đến mô hình. Một số phương pháp xử lý:

- Phát hiện giá trị ngoại lai bằng IQR (Interquartile Range)
- Dùng Z-score để phát hiện điểm dữ liệu cách biệt
- Cắt bỏ hoặc thay thế giá trị ngoại lai bằng giá trị trung bình của dữ liệu

4 Giảm chiều dữ liệu

Dữ liệu có nhiều đặc trưng có thể làm tăng độ phức tạp của mô hình. Một số phương pháp giảm chiều dữ liệu:

- PCA (Principal Component Analysis): Biến đổi dữ liệu về tập các thành phần chính.
- Feature Selection: Chọn ra những đặc trưng quan trọng nhất dựa trên độ quan trọng của chúng.

1.3.3 Ứng dụng của tiền xử lý dữ liệu trong bài toán

Trong bài toán cụ thể, tiền xử lý dữ liệu giúp:

- Loại bỏ dữ liệu bị thiếu và không hợp lệ.
- Chuẩn hóa các giá trị cảm biến để đảm bảo tính đồng nhất.
- Phát hiện và loại bỏ giá trị bất thường từ dữ liệu cảm biến.
- Chọn ra những đặc trưng quan trọng nhất để đưa vào mô hình.

1.4 Công cụ lập trình hỗ trợ

1.4.1 Ngôn ngữ lập trình

Ngôn ngữ lập trình đóng vai trò quan trọng trong việc triển khai các mô hình học máy và xử lý dữ liệu. Một số ngôn ngữ phổ biến trong lĩnh vực này bao gồm:

- **Python**: Là ngôn ngữ lập trình phổ biến nhất trong lĩnh vực học máy và trí tuệ nhân tạo. Python có cú pháp đơn giản, dễ học, cộng đồng hỗ trợ mạnh mẽ và hệ sinh thái thư viện phong phú như NumPy, Pandas, Scikit-learn, TensorFlow, PyTorch.
- **R**: Là một ngôn ngữ mạnh về phân tích và trực quan hóa dữ liệu, thường được sử dụng trong thống kê và khoa học dữ liệu.
- **Java và C++**: Được sử dụng trong các ứng dụng yêu cầu hiệu suất cao, đặc biệt trong môi trường sản xuất.

Trong bài toán này, Python được chọn làm ngôn ngữ chính vì sự linh hoạt và thư viện hỗ trợ phong phú trong lĩnh vực học máy.

1.4.2 Các thư viện phổ biến

Các thư viện lập trình đóng vai trò quan trọng trong việc hỗ trợ xử lý dữ liệu và triển khai mô hình học máy. Một số thư viện quan trọng bao gồm:

- **NumPy**: Cung cấp các công cụ làm việc với mảng đa chiều và hỗ trợ các phép toán ma trận.
- **Pandas**: Hỗ trợ thao tác, xử lý dữ liệu dạng bảng, giúp tổ chức dữ liệu hiệu quả.
- **Matplotlib & Seaborn**: Hai thư viện chính trong việc trực quan hóa dữ liệu, giúp phân tích xu hướng và mô hình hóa thông tin.
- **Scikit-learn**: Một thư viện mạnh mẽ cho học máy, cung cấp các công cụ để huấn luyện, đánh giá mô hình, tiền xử lý dữ liệu.
- **TensorFlow & PyTorch**: Hai nền tảng quan trọng trong học sâu (Deep Learning), hỗ trợ xây dựng các mô hình mạng nơ-ron phức tạp.
- **OpenCV**: Dùng trong xử lý ảnh và thị giác máy tính, hỗ trợ các tác vụ như phát hiện đối tượng, nhận diện khuôn mặt.

1.4.3 Môi trường lập trình

Bên cạnh ngôn ngữ và thư viện, môi trường lập trình cũng đóng vai trò quan trọng trong việc triển khai và thực nghiệm mô hình. Một số công cụ phổ biến gồm:

- **Google Colab**: Một môi trường lập trình dựa trên cloud, hỗ trợ GPU miễn phí, phù hợp cho các bài toán học máy.
- **Jupyter Notebook**: Công cụ mạnh mẽ giúp thực hiện mã nguồn Python một cách tương tác, dễ dàng trình bày kết quả.
- **PyCharm**: Một IDE chuyên dụng cho Python, hỗ trợ phát triển các dự án lớn.
- **VS Code**: Môi trường lập trình nhẹ, hỗ trợ nhiều ngôn ngữ, tích hợp tốt với Python.

Việc lựa chọn môi trường lập trình phù hợp giúp quá trình phát triển mô hình hiệu quả hơn, giảm thời gian xử lý và nâng cao khả năng tái sử dụng mã nguồn.

1.5 Tổng kết

Chương này đã trình bày tổng quan về các lý thuyết liên quan đến bài toán, bao gồm học máy, các thuật toán được sử dụng, các phương pháp tiền xử lý dữ liệu và các công cụ lập trình hỗ trợ. Những kiến thức này giúp xây dựng nền tảng quan trọng để tiếp cận và giải quyết bài toán một cách có hệ thống.

Học máy đóng vai trò cốt lõi trong việc xây dựng các hệ thống thông minh. Việc hiểu rõ về học có giám sát, học không giám sát và học tăng cường giúp lựa chọn phương pháp phù hợp với bài toán cụ thể. Các thuật toán như KNN và Random Forest được sử dụng trong bài toán này mang lại những ưu điểm nhất định trong việc phân loại và dự đoán.

Bên cạnh đó, tiền xử lý dữ liệu là một bước không thể thiếu để đảm bảo dữ liệu sạch, phù hợp và có thể sử dụng hiệu quả trong mô hình học máy. Các công cụ lập trình như Python cùng với các thư viện mạnh mẽ như Scikit-learn, Pandas, và TensorFlow giúp triển khai các mô hình học máy một cách dễ dàng và tối ưu.

Trong các chương tiếp theo, bài báo cáo sẽ đi sâu vào việc thiết kế mô hình, triển khai thực nghiệm và đánh giá kết quả nhằm kiểm chứng hiệu quả của các phương pháp được đề xuất

CHƯƠNG 2. MÔ TẢ TẬP DỮ LIỆU VÀ CÔNG NGHỆ SỬ DỤNG

2.1. Mô tả tập dữ liệu

2.1.1. Giới thiệu tập dữ liệu

Tập dữ liệu được sử dụng trong bài tập lớn là **Trending YouTube Videos** được lấy từ Kaggle. Đây là một tập dữ liệu lớn chứa thông tin về các video thịnh hành trên YouTube tại 113 quốc gia, giúp phân tích xu hướng nội dung và các yếu tố ảnh hưởng đến sự phổ biến của video.

1. Nguồn gốc và cách thu thập dữ liệu

Tập dữ liệu này được tổng hợp từ API của YouTube, cụ thể là YouTube Data API v3. Các dữ liệu về video thịnh hành được thu thập hàng ngày từ danh sách **Trending** của YouTube tại mỗi quốc gia. Điều này giúp phản ánh sự thay đổi xu hướng của nội dung theo thời gian và địa lý.

2. Phạm vi dữ liệu

- **Quốc gia**: Dữ liệu bao gồm thông tin từ 113 quốc gia trên toàn thế giới, tạo ra một tập dữ liệu phong phú để phân tích xu hướng nội dung theo vùng miền.
- **Thời gian thu thập**: Tập dữ liệu được cập nhật định kỳ, giúp theo dõi sự thay đổi về xu hướng video theo từng ngày, tuần hoặc tháng.
- **Danh mục nội dung**: Dữ liệu bao gồm nhiều danh mục khác nhau như:
    - Giải trí
    - Âm nhạc
    - Giáo dục
    - Công nghệ
    - Tin tức
    - Thể thao
    - Chương trình truyền hình

3. Đặc điểm của tập dữ liệu

Tập dữ liệu **Trending YouTube Videos** có một số đặc điểm quan trọng như sau:

- **Dữ liệu thời gian thực**: Dữ liệu được thu thập hàng ngày, cho phép theo dõi và phân tích xu hướng video theo từng ngày hoặc từng giai đoạn.
- **Mức độ phong phú**: Với hơn 113 quốc gia, tập dữ liệu phản ánh sự khác biệt về sở thích nội dung giữa các vùng lãnh thổ.

- **Phân loại rõ ràng**: Các video được gán nhãn theo danh mục, giúp dễ dàng phân tích mức độ phổ biến của từng loại nội dung.
- **Chứa thông tin tương tác**: Bao gồm số lượt xem, lượt thích, lượt bình luận, giúp đánh giá mức độ ảnh hưởng của video trên YouTube.
- **Dữ liệu có thể bị thiếu hoặc thay đổi**: Do YouTube có thể ẩn hoặc xóa video, tập dữ liệu có thể gặp tình trạng mất dữ liệu theo thời gian.

4. Ứng dụng của tập dữ liệu

Tập dữ liệu này có nhiều ứng dụng quan trọng trong các lĩnh vực khác nhau:

- **Phân tích xu hướng nội dung**: Xác định những yếu tố giúp một video trở nên phổ biến.
- **So sánh sự khác biệt theo quốc gia**: Tìm hiểu xem người xem từ các khu vực khác nhau có sở thích nội dung như thế nào.
- **Ứng dụng trong học máy và AI**: Huấn luyện mô hình để dự đoán xu hướng video trong tương lai.
- **Hỗ trợ chiến lược tiếp thị**: Giúp các nhà sáng tạo nội dung và doanh nghiệp tối ưu hóa chiến lược phát triển video.

2.1.2. Cấu trúc tập dữ liệu

Tập dữ liệu này chứa thông tin về các video thịnh hành (trending) trên YouTube tại **113 quốc gia**. Dữ liệu được thu thập từ YouTube API hoặc các nguồn tổng hợp dữ liệu từ nền tảng này, bao gồm thông tin về video, kênh đăng tải, số lượt xem, lượt thích, bình luận, và các yếu tố khác.

Mỗi dòng dữ liệu trong tập thể hiện một **trạng thái của một video tại một quốc gia vào một thời điểm cụ thể**, nghĩa là một video có thể xuất hiện nhiều lần trong tập dữ liệu nếu nó tiếp tục nằm trong danh sách video thịnh hành theo ngày hoặc theo quốc gia khác nhau.

Tập dữ liệu này có thể sử dụng cho nhiều mục đích khác nhau, như:

- Phân tích xu hướng nội dung trên YouTube.
- Đánh giá mức độ lan truyền của video theo thời gian và khu vực.
- Xác định những chủ đề, thể loại video phổ biến theo từng quốc gia.
- Dự đoán sự thay đổi trong xu hướng tiêu dùng nội dung video trực tuyến.

Nhóm 1: Thông tin cơ bản về video

Các thuộc tính trong nhóm này mô tả **nội dung chính của video**, giúp người dùng hiểu rõ về thông tin mô tả và các đặc điểm liên quan đến nội dung video.

Tên thuộc tính	Kiểu dữ liệu	Mô tả
video_id	String	Mã định danh duy nhất của video trên YouTube.
title	String	Tiêu đề của video.
description	String	Phần mô tả nội dung video do người đăng tải cung cấp.
thumbnail_url	String	URL của ảnh thu nhỏ (thumbnail) của video.
video_tags	String	Danh sách các thẻ (tags) giúp phân loại nội dung video.
kind	String	Loại nội dung video (ví dụ: video, short, livestream, v.v.).

Nhóm 2: Thông tin về kênh đăng tải

Các thuộc tính này cung cấp thông tin về kênh YouTube mà video được đăng tải, giúp phân tích mức độ ảnh hưởng của kênh và nhận diện những kênh có nhiều video thịnh hành.

Tên thuộc tính	Kiểu dữ liệu	Mô tả
channel_id	String	ID duy nhất của kênh YouTube.
channel_name	String	Tên kênh YouTube đăng tải video.

Thông tin này giúp xác định những kênh có tầm ảnh hưởng lớn, cũng như so sánh giữa các kênh có nội dung tương tự.

Nhóm 3: Thống kê về hiệu suất video

Nhóm này chứa các số liệu về mức độ phổ biến và tương tác của video, giúp đánh giá mức độ thành công của nội dung.

Tên thuộc tính	Kiểu dữ liệu	Mô tả
view_count	Integer	Tổng số lượt xem của video.
like_count	Integer	Tổng số lượt thích của video.
comment_count	Integer	Tổng số bình luận của video.

Các thông số này cho phép phân tích **sự lan truyền của video theo thời gian** và giúp dự đoán sự thành công của video dựa trên mức độ tương tác.

Nhóm 4: Xếp hạng và biến động xu hướng

Nhóm này mô tả sự thay đổi vị trí của video trong danh sách thịnh hành, cho biết tốc độ tăng hoặc giảm thứ hạng của video theo thời gian.

Tên thuộc tính	Kiểu dữ liệu	Mô tả
daily_rank	Integer	Xếp hạng của video trong danh sách thịnh hành hàng ngày.
daily_movement	Integer	Mức độ thay đổi thứ hạng của video so với ngày trước đó.
weekly_movement	Integer	Mức độ thay đổi thứ hạng của video theo tuần.

Nhóm 5: Thông tin về thời gian và vị trí

Nhóm này giúp xác định thời điểm và địa điểm mà video xuất hiện trong danh sách thịnh hành.

Tên thuộc tính	Kiểu dữ liệu	Mô tả
snapshot_date	Date	Ngày mà video được ghi nhận trong danh sách thịnh hành.
publish_date	Date	Ngày video được đăng tải lên YouTube.
country	String	Quốc gia mà video đang thịnh hành.
language	String	Ngôn ngữ của video.

2.1.3. Đánh giá cấu trúc dữ liệu

1. Đặc điểm của tập dữ liệu

1.1. Kích thước dữ liệu

Tập dữ liệu có quy mô lớn, chứa hàng triệu bản ghi, do một video có thể xuất hiện nhiều lần trong danh sách thịnh hành theo ngày hoặc theo quốc gia. Cụ thể:

- Số lượng bản ghi (dòng dữ liệu): Phụ thuộc vào phạm vi thu thập dữ liệu (số quốc gia, thời gian theo dõi).
- Số lượng video duy nhất: Một video có thể xuất hiện nhiều lần do nằm trong danh sách trending của nhiều quốc gia hoặc trong nhiều ngày liên tiếp.

Ví dụ:

- Một video có thể xuất hiện 10 ngày liên tiếp trong danh sách trending của một quốc gia, tạo thành 10 bản ghi khác nhau.
- Video có thể trending ở 5 quốc gia khác nhau cùng ngày, dẫn đến 5 bản ghi riêng biệt.

Do đó, tập dữ liệu có cấu trúc lặp lại, ảnh hưởng đến phương pháp xử lý và phân tích.

1.2. Dạng dữ liệu và kiểu dữ liệu

Tập dữ liệu được tổ chức theo dạng bảng (tabular data) với các cột đại diện cho các thuộc tính (features) của video. Các kiểu dữ liệu chính gồm:

- Chuỗi ký tự (String): video_id, title, description, thumbnail_url, channel_name, country, language, video_tags, kind.
- Số nguyên (Integer): view_count, like_count, comment_count, daily_rank, daily_movement, weekly_movement.
- Ngày tháng (DateTime): snapshot_date, publish_date.

Với sự đa dạng về kiểu dữ liệu, cần có phương pháp xử lý phù hợp, đặc biệt là khi

thực hiện phân tích thống kê và trực quan hóa.

1.3. Sự lặp lại trong dữ liệu

Có hai kiểu lặp lại chính trong tập dữ liệu:

- Lặp lại theo thời gian: Một video có thể xuất hiện nhiều ngày liên tiếp trong danh sách trending.
- Lặp lại theo quốc gia: Một video có thể trending cùng ngày nhưng ở nhiều quốc gia khác nhau.

Hệ quả:

- Làm tăng đáng kể số lượng bản ghi, ảnh hưởng đến tốc độ xử lý dữ liệu.
- Khi phân tích, cần xác định rõ phương pháp xử lý (ví dụ: chỉ lấy lần xuất hiện đầu tiên của video hoặc tính trung bình theo ngày).

2. Dữ liệu bị thiếu và phương pháp xử lý

Tập dữ liệu có thể chứa giá trị thiếu ở một số cột do chính sách bảo mật của YouTube hoặc do người dùng không cung cấp thông tin. Các cột thường bị thiếu gồm:

- description (Mô tả video): Một số video không có mô tả.
- like_count, comment_count (Lượt thích và bình luận): YouTube có thể ẩn số liệu này với một số video.
- language (Ngôn ngữ video): Có thể bị nhận diện sai hoặc không đồng nhất.

2.1. Phương pháp xử lý giá trị thiếu

- Bỏ bản ghi: Nếu số lượng giá trị thiếu nhỏ và không ảnh hưởng đến tổng thể.
- Điền giá trị mặc định:
    - Nếu description bị thiếu, có thể thay bằng "No description available" để tránh lỗi xử lý.
    - Nếu like_count hoặc comment_count bị thiếu, có thể thay bằng giá trị trung bình hoặc ước tính dựa trên dữ liệu tương tự.
- Suy luận giá trị:
    - language có thể được suy luận từ title hoặc description bằng thuật toán xử lý ngôn ngữ tự nhiên (NLP).

3. Mối quan hệ giữa các thuộc tính

Việc hiểu rõ mối quan hệ giữa các cột giúp phân tích dữ liệu chính xác hơn. Dưới đây là một số mối quan hệ quan trọng:

3.1. Mối quan hệ giữa lượt xem, lượt thích và bình luận

- Tương quan dương mạnh:
    - Video có nhiều view_count thường có nhiều like_count và comment_count.
    - Công thức ước tính:

$$like\_count \approx 0.04 \times view\_count$$

Ảnh 1: công thức tính tương quan

- Ngoại lệ:
    - Một số video có lượt xem cao nhưng ít lượt thích/bình luận, có thể do nội dung gây tranh cãi hoặc không thu hút tương tác.

3.2. Mối quan hệ giữa xếp hạng thịnh hành và số liệu tương tác

- Video có daily_rank thấp (hạng cao) thường có số lượt xem lớn hơn.
- Một số video có thể đạt thứ hạng cao mà không cần nhiều lượt xem, do thuật toán YouTube xem xét nhiều yếu tố khác như tỷ lệ giữ chân người xem.

3.3. Mối quan hệ giữa quốc gia và xu hướng nội dung

- Một số quốc gia có đặc điểm xu hướng riêng biệt, ví dụ:
    - Mỹ: Nội dung giải trí, công nghệ, Vlog phổ biến.
    - Nhật Bản: Anime, J-Pop, game streaming nổi bật.
    - Việt Nam: Nhạc trẻ, vlog đời sống, tin tức thời sự.
- Sự lan truyền giữa các quốc gia:
    - Một video có thể bắt đầu trending tại một nước rồi lan sang nước khác.
    - Ví dụ: Video từ Mỹ có thể trending tại Canada, Anh trước khi lan sang các nước khác.

4. Đánh giá khả năng mở rộng và ứng dụng

4.1. Tính khả dụng của dữ liệu

- Dữ liệu có thể mở rộng bằng cách thu thập thêm từ nhiều quốc gia hoặc tăng thời gian theo dõi.
- Có thể kết hợp với dữ liệu ngoài (Google Trends, Social Media) để tăng độ chính xác khi phân tích xu hướng.
- 4.2. Khả năng ứng dụng
- Dự đoán xu hướng video: Dùng học máy để dự đoán video nào có khả năng trending trong tương lai.

- Tối ưu hóa nội dung kênh YouTube: Giúp nhà sáng tạo nội dung xác định yếu tố thu hút người xem.
- Phân tích thị trường: Xác định sự khác biệt về sở thích người xem theo khu vực.

5. Kết luận

Tập dữ liệu có cấu trúc phức tạp với nhiều mối quan hệ quan trọng giữa các thuộc tính. Khi phân tích, cần loại bỏ dữ liệu lặp lại, xử lý giá trị thiếu, và hiểu rõ các yếu tố ảnh hưởng đến sự thịnh hành của video. Việc khai thác dữ liệu đúng cách có thể mang lại nhiều ứng dụng hữu ích trong nghiên cứu xu hướng nội dung số.

2.2. Công nghệ sử dụng

Trong bài toán phân tích xu hướng video YouTube thịnh hành, nhóm đã sử dụng nhiều công nghệ và thư viện lập trình hiện đại để xử lý dữ liệu, bao gồm các công cụ hỗ trợ xử lý dữ liệu lớn, trực quan hóa, và môi trường thực thi phù hợp.

2.2.1. Apache Spark (PySpark)

Mô tả

Apache Spark là một nền tảng xử lý dữ liệu lớn (Big Data), hỗ trợ xử lý phân tán trên nhiều máy tính, giúp tăng tốc độ tính toán so với các công cụ truyền thống như Pandas. PySpark là thư viện của Spark dành riêng cho Python, cho phép làm việc với dữ liệu lớn một cách hiệu quả.

Ứng dụng trong bài toán

- Tạo SparkSession: SparkSession là điểm vào để làm việc với Spark. Trong đoạn code, nhóm đã tạo một phiên làm việc Spark để quản lý dữ liệu.
- Đọc dữ liệu từ CSV: PySpark hỗ trợ đọc dữ liệu từ nhiều nguồn khác nhau, trong bài toán này nhóm đã sử dụng phương thức spark.read.csv() để đọc dữ liệu từ tập tin CSV.
- Xử lý dữ liệu:
    - Lọc dữ liệu: Dữ liệu có thể chứa các giá trị bị thiếu hoặc không hợp lệ, nhóm đã sử dụng các chức năng của Spark như dropna() và fillna() để làm sạch dữ liệu.
    - Chuyển đổi dữ liệu: Một số cột dữ liệu cần được chuyển đổi kiểu dữ liệu, nhóm đã sử dụng cast() để đảm bảo dữ liệu đúng định dạng.
    - Tính toán thống kê: Sử dụng các hàm như groupBy(), agg() để tính toán tổng số lượt xem, lượt thích, số lần xuất hiện của video trên bảng xếp hạng.

Lợi ích của Spark

- Xử lý dữ liệu nhanh hơn so với Pandas, đặc biệt với tập dữ liệu lớn.

- Hỗ trợ xử lý phân tán trên nhiều máy, tối ưu tài nguyên.
- Có thể tích hợp với nhiều hệ thống lưu trữ dữ liệu khác nhau.

2.2.2. Pandas

Mô tả

Pandas là một thư viện phân tích dữ liệu mạnh mẽ trong Python, giúp thao tác với dữ liệu dạng bảng (DataFrame). Dù không nhanh như Spark khi làm việc với dữ liệu lớn, Pandas vẫn rất hữu ích trong quá trình tiền xử lý dữ liệu nhỏ và kiểm tra dữ liệu.

Ứng dụng trong bài toán

- Đọc và xử lý dữ liệu ban đầu: Pandas được sử dụng để đọc tập tin CSV, kiểm tra dữ liệu ban đầu trước khi đưa vào xử lý bằng Spark.
- Chuyển đổi giữa Pandas DataFrame và Spark DataFrame: Vì Spark hoạt động với Spark DataFrame, trong khi một số thư viện khác yêu cầu Pandas DataFrame, nhóm đã sử dụng .toPandas() để chuyển đổi dữ liệu khi cần.
- Xử lý dữ liệu nhỏ: Một số thao tác như kiểm tra dữ liệu bị thiếu (isnull().sum()), xem thông tin dữ liệu (info(), describe()) được thực hiện bằng Pandas trước khi chuyển sang Spark.

Lợi ích của Pandas

- Thao tác dễ dàng, cú pháp đơn giản.
- Hỗ trợ nhiều hàm xử lý dữ liệu mạnh mẽ.
- Kết hợp tốt với các thư viện khác như Matplotlib và Seaborn để trực quan hóa dữ liệu.

2.2.3. Matplotlib và Seaborn

Mô tả

Matplotlib và Seaborn là hai thư viện trực quan hóa dữ liệu phổ biến trong Python. Matplotlib hỗ trợ vẽ biểu đồ cơ bản, trong khi Seaborn cung cấp giao diện trực quan đẹp hơn, phù hợp cho việc phân tích dữ liệu.

Ứng dụng trong bài toán

- Vẽ biểu đồ phân bố dữ liệu: Biểu đồ histogram giúp hiểu rõ hơn về phân bố lượt xem, lượt thích của các video thịnh hành.
- Vẽ biểu đồ xu hướng theo thời gian: Sử dụng line plot để phân tích xu hướng video trending theo từng ngày.
- Biểu đồ cột so sánh: So sánh số lượt xem, lượt thích giữa các quốc gia khác nhau.

Lợi ích

- Hỗ trợ nhiều loại biểu đồ trực quan.
- Giúp phân tích dữ liệu trực quan hơn, dễ dàng phát hiện xu hướng.

2.2.4. Môi trường thực thi

Google Colab

- Mô tả: Google Colab là một môi trường lập trình trên nền tảng đám mây, hỗ trợ Python và có thể chạy Spark.
- Lợi ích:
    - Không cần cài đặt phần mềm trên máy tính.
    - Hỗ trợ GPU miễn phí, giúp tăng tốc xử lý dữ liệu.
    - Hỗ trợ làm việc nhóm dễ dàng.

Jupyter Notebook

- Mô tả: Jupyter Notebook là một công cụ cho phép chạy mã Python từng bước và hiển thị kết quả ngay lập tức.
- Lợi ích:
    - Giao diện trực quan, dễ sử dụng.
    - Hỗ trợ hiển thị kết quả dưới dạng bảng, biểu đồ.
    - Dễ dàng kết hợp code với tài liệu mô tả.

Hệ điều hành

- Bài toán có thể được triển khai trên Windows, Linux hoặc macOS với môi trường Python.
- Sử dụng các công cụ như Anaconda để quản lý thư viện dễ dàng hơn.

Tổng kết

Các công nghệ được sử dụng trong bài toán giúp tối ưu quá trình xử lý dữ liệu lớn, từ thu thập, làm sạch, phân tích cho đến trực quan hóa dữ liệu. Việc kết hợp PySpark, Pandas, Matplotlib và Seaborn mang lại hiệu quả cao, giúp rút ra được nhiều thông tin hữu ích từ tập dữ liệu xu hướng video YouTube thịnh hành.

CHƯƠNG 3: KẾT QUẢ XỬ LÝ, PHÂN TÍCH DỮ LIỆU VÀ ỨNG DỤNG

3.1. Tiền Xử Lý Dữ Liệu: Bước Chuẩn Bị Thiết Yếu

Giai đoạn tiền xử lý dữ liệu đóng vai trò then chốt trong việc đảm bảo chất lượng và tính phù hợp của dữ liệu cho các bước phân tích và xây dựng mô hình tiếp theo. Nó bao gồm một loạt các bước tỉ mỉ để làm sạch, chuyển đổi và chuẩn hóa dữ liệu.

3.1.1. Khởi Tạo SparkSession: Thiết Lập Nền Tảng Xử Lý Dữ Liệu

Mô tả chi tiết:

- Mục đích cốt lõi: Khởi tạo một SparkSession để thiết lập môi trường làm việc cho Apache Spark. SparkSession là điểm truy cập duy nhất để làm việc với Spark, cho phép ứng dụng tận dụng khả năng xử lý dữ liệu phân tán mạnh mẽ của Spark.
- Cấu hình bộ nhớ: Việc cấu hình bộ nhớ cụ thể cho cả driver và executors là rất quan trọng để đảm bảo hiệu năng và ổn định của ứng dụng.
    - Driver (8GB): Driver là tiến trình điều phối chính của ứng dụng Spark, chịu trách nhiệm lập kế hoạch, phân phối công việc và thu thập kết quả. Việc cấp phát bộ nhớ đủ lớn cho driver cho phép nó xử lý hiệu quả các phép toán phức tạp và quản lý metadata của DataFrame.
    - Executors (8GB): Executors là các tiến trình thực thi công việc trên các node của cluster. Mỗi executor được cấu hình với 8GB bộ nhớ để có thể xử lý các partition dữ liệu một cách hiệu quả và thực hiện các phép biến đổi nhanh chóng.
- Tối ưu hóa cấu hình: Việc cấu hình bộ nhớ này giúp Spark quản lý dữ liệu trong bộ nhớ tốt hơn, giảm thiểu việc sử dụng đĩa và cải thiện tốc độ xử lý tổng thể. Đồng thời, nó cũng giúp ngăn ngừa các lỗi liên quan đến bộ nhớ, chẳng hạn như OutOfMemoryError, vốn thường xảy ra khi làm việc với dữ liệu lớn.
- Kiểm tra khởi tạo: Việc in ra phiên bản SparkSession (print(spark)) cung cấp một phương tiện kiểm tra đơn giản để xác minh rằng Spark đã được khởi tạo thành công và sẵn sàng để sử dụng.

3.1.2. Định Nghĩa Schema cho Dữ Liệu: Kiểm Soát Cấu Trúc Dữ Liệu

Mô tả chi tiết:

- Vai trò của Schema: Trong Apache Spark, schema đóng vai trò như một bản thiết kế, mô tả cấu trúc và kiểu dữ liệu của từng cột trong DataFrame.
- Cấu trúc Schema:
    - Chuỗi (StringType): title, channel_name, description, video_id, channel_id, video_tags, kind, language. Các cột này chứa dữ liệu dạng văn bản, thường đòi hỏi các bước xử lý phức tạp hơn.
    - Số nguyên (IntegerType): daily_rank, daily_movement, weekly_movement, view_count, like_count, comment_count. Các cột này chứa dữ liệu số nguyên, phù hợp cho các phép toán số học và thống kê.

- Ngày (DateType/TimestampType): snapshot_date (ngày chụp snapshot), publish_date (thời điểm video được đăng tải). Các cột này ban đầu ở dạng chuỗi, nhưng được chuyển đổi sang kiểu ngày để phục vụ cho phân tích thời gian.

- Mục đích và Lợi ích:
  - Ngăn ngừa lỗi kiểu dữ liệu: Bằng cách chỉ định kiểu dữ liệu cho từng cột, chúng tôi đảm bảo rằng Spark sẽ xử lý dữ liệu một cách chính xác và tránh các lỗi do kiểu dữ liệu không tương thích.
  - Cải thiện hiệu suất: Việc khai báo schema cho phép Spark tối ưu hóa quá trình đọc và xử lý dữ liệu, vì Spark không cần phải tự động suy luận kiểu dữ liệu.
  - Tăng cường tính nhất quán: Schema giúp đảm bảo rằng dữ liệu tuân thủ một cấu trúc nhất quán, giúp đơn giản hóa quá trình phân tích và báo cáo.
  - Kiểm soát dữ liệu: Schema đóng vai trò là một công cụ kiểm soát dữ liệu, cho phép chúng tôi xác định và xử lý các giá trị không hợp lệ.

3.1.3. Đọc Dữ Liệu Từ File CSV: Tiếp Nhận Dữ Liệu Thô

Mô tả chi tiết:

- Nguồn dữ liệu: Dữ liệu được đọc từ file CSV có tên trending_yt_videos_113_countries.csv. File này chứa thông tin về các video thịnh hành trên YouTube từ 113 quốc gia khác nhau.
- Phương thức đọc: Sử dụng hàm spark.read.csv để nạp dữ liệu vào Spark DataFrame. DataFrame là một cấu trúc dữ liệu phân tán, cho phép Spark xử lý dữ liệu lớn một cách hiệu quả.
- Tùy chọn cấu hình: Quá trình đọc dữ liệu được cấu hình với các tùy chọn đặc biệt để xử lý các đặc điểm phức tạp của file CSV, bao gồm:
  - header=True: Sử dụng dòng đầu tiên của file CSV làm tên cột.
  - sep=',': Chỉ định dấu phẩy là ký tự phân tách giữa các cột.
  - encoding="ISO-8859-1": Sử dụng bảng mã ISO-8859-1 để hỗ trợ nhiều ký tự đặc biệt khác nhau.
  - quote="", escape="": Chỉ định dấu nháy kép (") làm ký tự bao quanh và ký tự thoát để xử lý các chuỗi chứa dấu nháy kép.
  - multiline=True: Cho phép các giá trị cột kéo dài trên nhiều dòng, cần thiết để đọc các mô tả video dài.
  - schema(schema): Sử dụng schema đã định nghĩa ở bước trước.
- Mục đích:
  - Đọc chính xác dữ liệu: Đảm bảo rằng dữ liệu được đọc một cách chính xác từ file CSV, bao gồm cả các ký tự đặc biệt, dấu nháy kép và dữ liệu nhiều dòng.
  - Hỗ trợ dữ liệu lớn: Cho phép Spark xử lý các tập dữ liệu lớn một cách hiệu

quả bằng cách chia nhỏ dữ liệu thành các partition và xử lý song song trên các node của cluster.

- Tuân thủ schema: Áp dụng schema đã định nghĩa để đảm bảo tính nhất quán và hiệu suất trong quá trình xử lý dữ liệu.

3.1.4. Kiểm Tra và Đánh Giá Dữ Liệu: Đảm Bảo Chất Lượng Dữ Liệu

Mô tả chi tiết:

- Mục đích: Xác minh dữ liệu đã được đọc và hiểu chính xác bởi Spark, phát hiện các vấn đề tiềm ẩn trước khi tiến hành phân tích.
- Các phương pháp kiểm tra:
  - df_spark.count(): Xác định tổng số dòng trong DataFrame.
  - len(df_spark.columns): Xác định số lượng cột trong DataFrame.
  - df_spark.show(5): Hiển thị 5 dòng đầu tiên để kiểm tra trực quan cấu trúc và nội dung dữ liệu.
  - df_spark.printSchema(): In ra cấu trúc schema của DataFrame để so sánh với schema đã định nghĩa và đảm bảo tính nhất quán.
- Giá trị của việc kiểm tra:
  - Phát hiện lỗi sớm: Việc kiểm tra ngay sau khi đọc dữ liệu giúp phát hiện các lỗi định dạng hoặc sự không khớp giữa dữ liệu và schema.
  - Đảm bảo tính toàn vẹn: Xác minh rằng tất cả các dòng và cột được đọc một cách chính xác.
  - Hiểu rõ cấu trúc dữ liệu: Giúp người phân tích hiểu rõ hơn về dữ liệu mà họ đang làm việc.

3.1.5. Chuyển Đổi Kiểu Dữ Liệu: Chuẩn Hóa Dữ Liệu

Mô tả chi tiết:

- Sự cần thiết của chuyển đổi kiểu dữ liệu:
  - Ban đầu, các cột ngày tháng được đọc dưới dạng chuỗi, điều này gây khó khăn cho việc thực hiện các phép toán và phân tích liên quan đến thời gian.
- Các cột được chuyển đổi:
  - snapshot_date: Chuyển đổi từ kiểu chuỗi (yyyy-MM-dd) sang kiểu DateType (ngày).
  - publish_date: Chuyển đổi từ kiểu chuỗi (yyyy-MM-dd HH:mm:ss+00:00) sang kiểu TimestampType (thời gian). Quá trình này bao gồm hai bước:
    1. Loại bỏ thông tin múi giờ (ví dụ: "+00:00") bằng hàm regexp_replace.
    2. Sử dụng hàm to_timestamp để chuyển đổi chuỗi sang kiểu timestamp.
- Mục đích của việc chuyển đổi:
  - Hỗ trợ phân tích thời gian: Cho phép thực hiện các phép toán so sánh, tính

toán khoảng thời gian, và phân tích xu hướng theo thời gian.

- Đảm bảo tính chính xác: Tránh các lỗi tiềm ẩn do việc so sánh chuỗi ngày tháng không đúng định dạng.

3.1.6. Trích Xuất Đặc Trưng Thời Gian: Làm Giàu Thông Tin

Mô tả chi tiết:

- Mục đích của việc trích xuất đặc trưng thời gian:
  - Tăng cường khả năng của mô hình để nắm bắt các yếu tố thời gian ảnh hưởng đến lượt xem.
  - Cung cấp các thông tin chi tiết hơn về thời điểm video được đăng tải và thời điểm snapshot được thực hiện.
- Các đặc trưng được trích xuất:
  - Từ snapshot_date:
    - snapshot_year: Năm của snapshot.
    - snapshot_month: Tháng của snapshot.
    - snapshot_weekday: Thứ trong tuần của snapshot.
  - Từ publish_date:
    - publish_year: Năm phát hành video.
    - publish_month: Tháng phát hành video.
    - publish_day: Ngày phát hành video.
- Lý do loại bỏ cột gốc: Sau khi trích xuất các đặc trưng thời gian, các cột snapshot_date và publish_date gốc không còn cần thiết và được loại bỏ để tránh dư thừa và giảm độ phức tạp của dữ liệu.

3.1.7. Kiểm Tra Giá Trị Thiếu: Tìm Kiếm Điểm Yếu

Mô tả chi tiết:

- Quan trọng của việc kiểm tra giá trị thiếu:
  - Dữ liệu thiếu có thể gây ra sai lệch trong quá trình phân tích và làm giảm độ chính xác của mô hình.
  - Việc xác định các cột có nhiều giá trị thiếu giúp tập trung nguồn lực vào việc xử lý những vấn đề quan trọng nhất.
- Phương pháp kiểm tra:
  - Sử dụng hàm isNull() và cast("int") để chuyển đổi các giá trị thiếu thành 1 và các giá trị khác thành 0.
  - Sử dụng hàm sum() để tính tổng số lượng giá trị thiếu cho từng cột.
  - Sử dụng hàm show() để hiển thị kết quả thống kê.
- Ví dụ: Một cột có quá nhiều giá trị thiếu có thể cần phải loại bỏ hoàn toàn, trong khi

các cột khác có thể được xử lý bằng cách điền giá trị mặc định hoặc sử dụng thuật toán ước tính.

3.1.8. Xử Lý Dữ Liệu Thiếu: Loại Bỏ Hoặc Điền Giá Trị

Mô tả chi tiết:

- Mục đích: Quyết định một trong những cách thức phù hợp để loại bỏ dữ liệu còn thiếu.
- Phương pháp loại bỏ được sử dụng (df_spark.dropna()):
    - Loại bỏ bất kỳ dòng nào chứa ít nhất một giá trị thiếu trong bất kỳ cột nào.
    - Đây là một phương pháp đơn giản và hiệu quả, nhưng có thể làm giảm đáng kể kích thước tập dữ liệu nếu có nhiều dòng chứa giá trị thiếu.
- Hạn chế và Cân nhắc:
    - Mất mát thông tin: Việc loại bỏ dòng có thể dẫn đến mất mát thông tin quan trọng, đặc biệt nếu số lượng dòng bị loại bỏ là lớn.
    - Sai lệch: Nếu các giá trị thiếu không phân bố ngẫu nhiên trong tập dữ liệu, việc loại bỏ có thể dẫn đến sai lệch trong quá trình phân tích.
    - Các phương pháp thay thế: Trong một số trường hợp, có thể sử dụng các phương pháp điền giá trị thiếu (imputation) thay vì loại bỏ hoàn toàn. Các phương pháp imputation có thể bao gồm:
        - Điền giá trị trung bình/trung vị: Sử dụng giá trị trung bình hoặc trung vị của cột để điền vào các giá trị thiếu.
        - Sử dụng giá trị mặc định: Thay thế giá trị thiếu bằng một giá trị mặc định (ví dụ: "Unknown" cho cột language).
        - Sử dụng thuật toán ước tính: Áp dụng các thuật toán máy học để dự đoán và điền các giá trị thiếu.

3.1.9. Kiểm Tra Dữ Liệu Trùng Lặp: Loại Bỏ Các Bản Sao

Mô tả chi tiết:

- Ảnh hưởng của dữ liệu trùng lặp: Dữ liệu trùng lặp có thể làm sai lệch kết quả phân tích và đánh giá mô hình, đặc biệt là trong các bài toán phân loại và hồi quy.
- Phương pháp phát hiện trùng lặp:
    - df_spark.count(): Đếm tổng số lượng dòng trong DataFrame.
    - df_spark.dropDuplicates().count(): Đếm số lượng dòng duy nhất sau khi loại bỏ các dòng trùng lặp.
    - Tính hiệu của hai số lượng để xác định số dòng trùng lặp.
- Biện pháp xử lý: Loại bỏ các dòng trùng lặp để đảm bảo tính chính xác của dữ liệu.

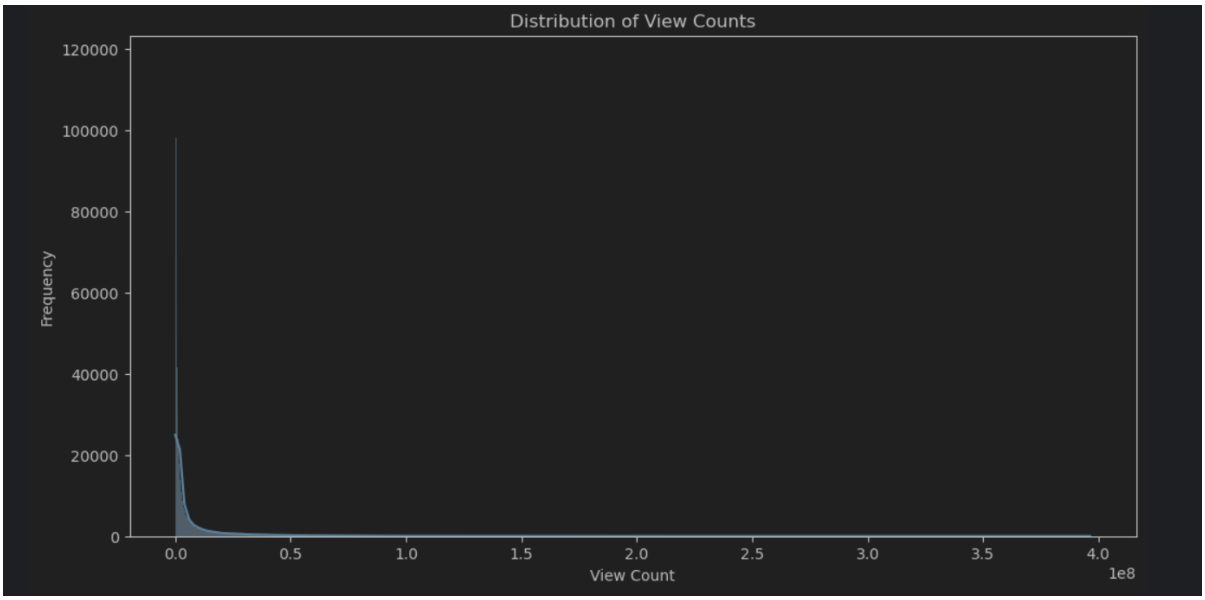
3.2. Trực Quan Hóa Dữ Liệu: Khám Phá Thông Tin Chi Tiết

Sau quá trình tiền xử lý, chúng tôi sử dụng các kỹ thuật trực quan hóa để khám phá các mẫu, xu hướng và mối quan hệ trong dữ liệu.

3.2.1. Phân Phối Của Lượt Xem (View Count)

Mô tả: Hiển thị phân phối tần suất của cột "view_count" bằng biểu đồ histogram, cùng với đường cong KDE (Kernel Density Estimate) để ước tính mật độ phân phối.
Giá trị của phân tích:

- Đặc điểm hình dạng: Cho thấy dữ liệu có phân phối chuẩn hay lệch. Nếu lệch phải, điều này cho thấy có nhiều video có lượt xem thấp, trong khi một số ít video có lượt xem cực kỳ cao.
- Xác định outlier: Giúp nhận diện các giá trị ngoại lệ (outliers) có thể ảnh hưởng đến quá trình phân tích.

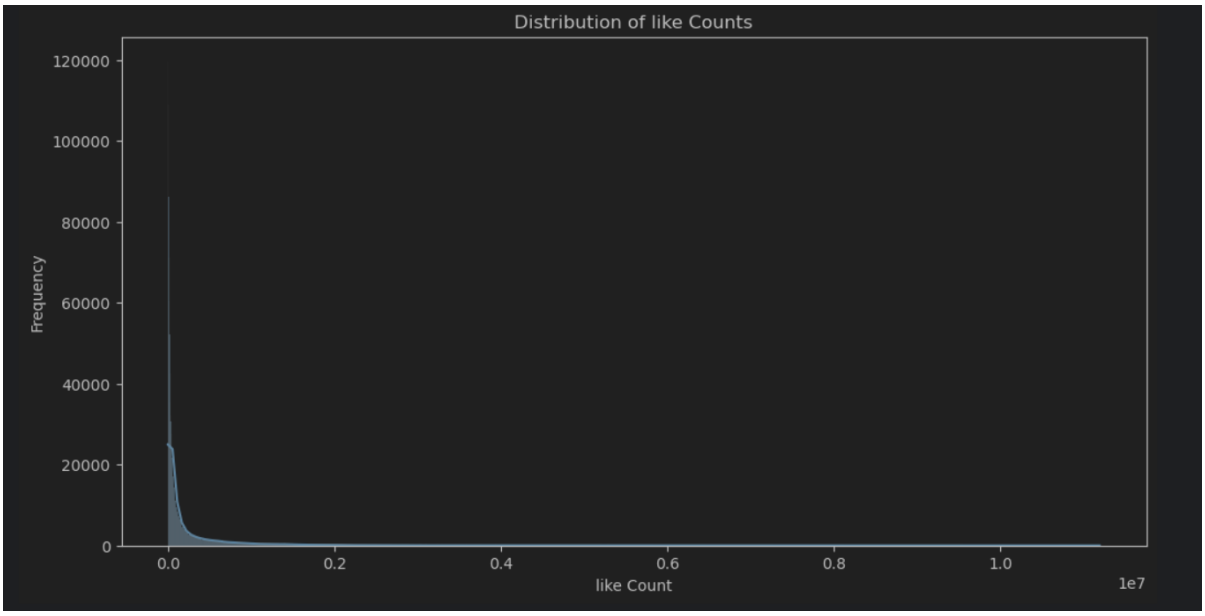


Ảnh 2: biểu đồ lượt xem

3.2.2. Phân Phối của Lượt Thích (Like Count)

Mô tả: Tương tự như lượt xem, vẽ biểu đồ histogram và đường cong KDE cho cột "like_count" để xem xét phân phối tần suất.
Giá trị của phân tích:

- Tương quan với lượt xem: So sánh hình dạng phân phối của lượt thích với lượt xem để đánh giá mức độ tương quan giữa hai biến này.

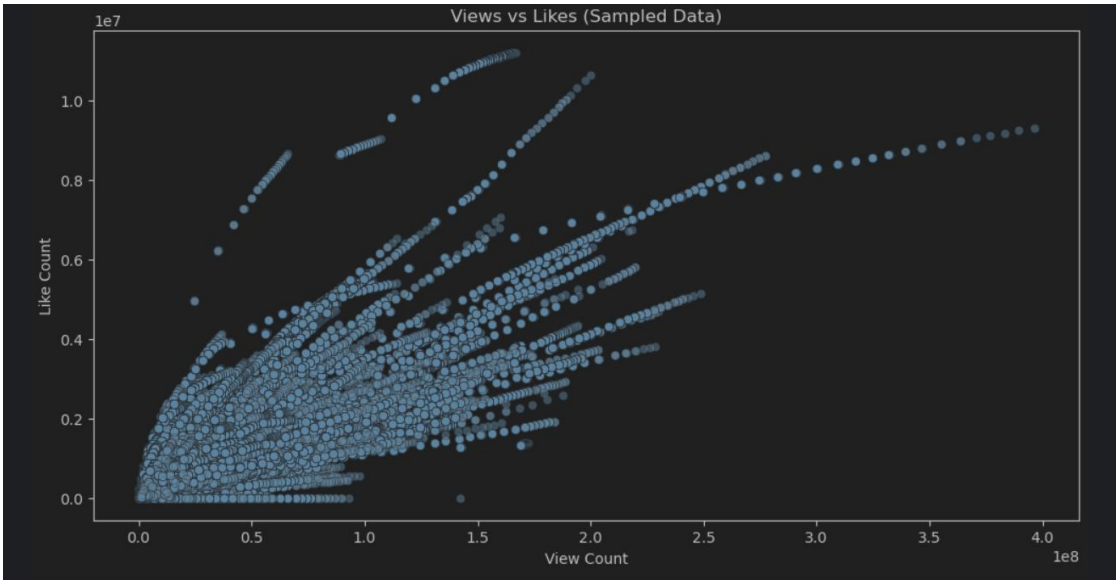


Ảnh 3: biểu đồ lượt like

3.2.3. Tương Quan Giữa Lượt Xem và Lượt Thích

Mô tả: Sử dụng biểu đồ phân tán (scatterplot) để trực quan hóa mối quan hệ giữa lượt xem và lượt thích.
Giá trị của phân tích:

- Đánh giá mối quan hệ: Biểu đồ scatterplot giúp xác định xem có mối quan hệ tuyến tính hay phi tuyến giữa hai biến hay không.
- Nhận biết các cụm điểm: Có thể có các cụm điểm trên biểu đồ, cho thấy các nhóm video có đặc điểm tương tự về lượt xem và lượt thích.

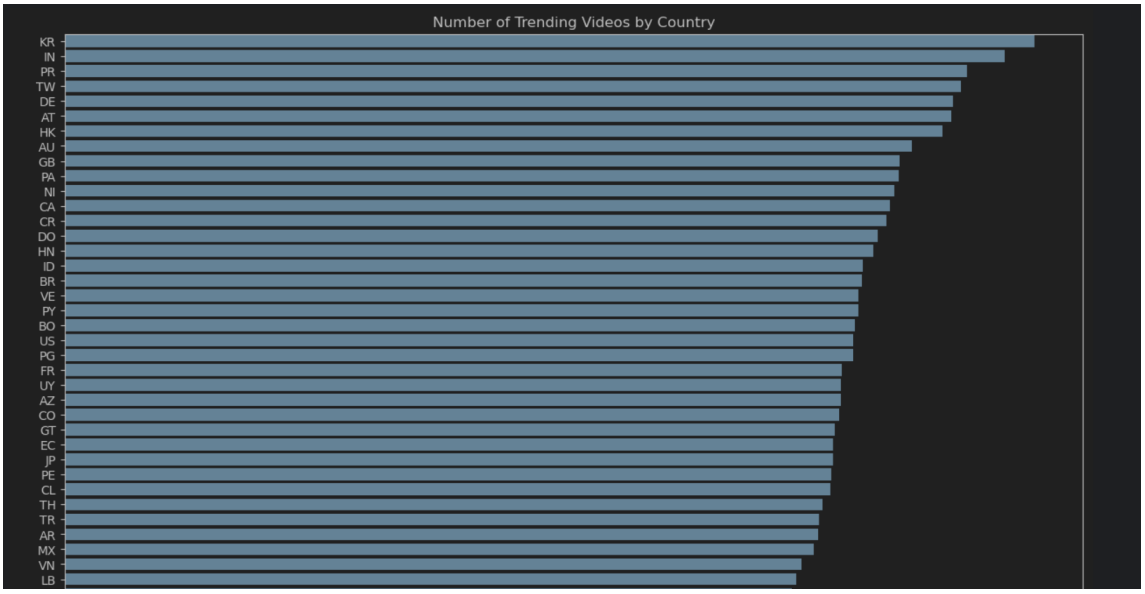


Ảnh 4: biểu đồ tương quan giữa lượt xem và lượt thích

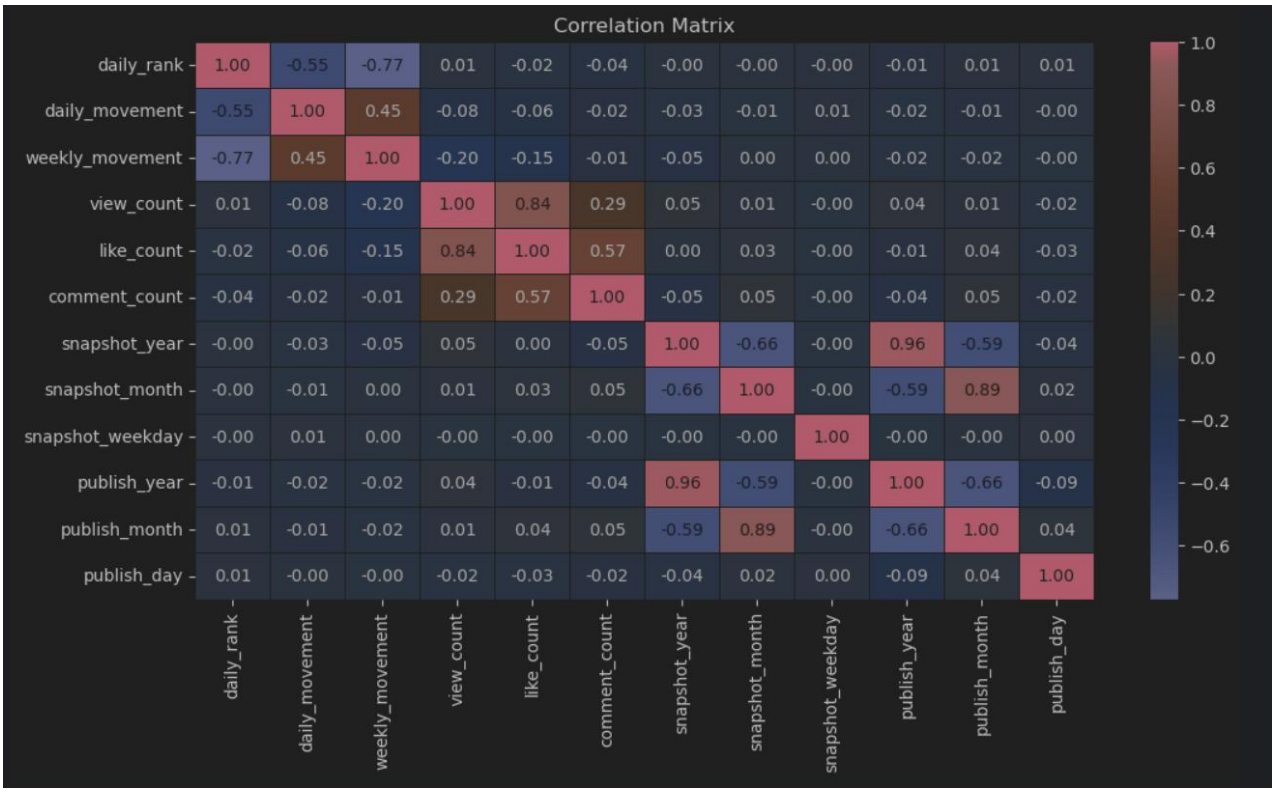
3.2.4. Số Lượng Video Theo Quốc Gia

Mô tả: Sử dụng biểu đồ cột (barplot) để hiển thị số lượng video thịnh hành từ mỗi quốc gia.
Giá trị của phân tích:

- Xác định các thị trường quan trọng: Cho thấy các quốc gia có số lượng video thịnh hành lớn, giúp tập trung các nỗ lực phân tích và marketing.
- So sánh mức độ cạnh tranh: Đánh giá mức độ cạnh tranh trên nền tảng YouTube ở các quốc gia khác nhau.



Ảnh 5: biểu đồ số lượng video theo quốc gia



Ảnh 6: Ma trận tương quan

3.3. Xây Dựng Mô Hình

Giai đoạn này tập trung vào việc lựa chọn, xây dựng, và huấn luyện mô hình máy học để dự đoán lượt xem video.

3.3.1. Chuyển Đổi Các Kiểu Dữ Liệu String

Mô tả chi tiết:

- Lý do cần mã hóa: Các thuật toán máy học thường yêu cầu dữ liệu đầu vào ở dạng số. Vì vậy, cần chuyển đổi các cột dạng chuỗi sang dạng số để sử dụng trong mô hình.
- Sử dụng StringIndexer:
    - StringIndexer: Là một công cụ trong Spark MLlib để mã hóa các cột dạng chuỗi thành các chỉ số số nguyên. Các chỉ số được gán dựa trên tần suất xuất hiện của các giá trị chuỗi (giá trị xuất hiện nhiều nhất được gán chỉ số 0).
    - Mỗi cột dạng chuỗi được gán một StringIndexer riêng biệt.
- Sử dụng Pipeline:
    - Pipeline: Cho phép kết hợp nhiều bước xử lý dữ liệu (như mã hóa chuỗi, chuẩn hóa) thành một quy trình duy nhất.
    - Pipeline giúp đơn giản hóa quá trình huấn luyện mô hình và đảm bảo rằng các bước xử lý dữ liệu được thực hiện một cách nhất quán trên cả tập huấn

luyện và tập kiểm thử.

- Quá trình mã hóa và loại bỏ cột gốc: Sau khi mã hóa, các cột chuỗi gốc được loại bỏ để tránh gây nhầm lẫn và giảm kích thước tập dữ liệu.

3.3.2. Lựa Chọn Đặc Trưng

Mô tả chi tiết:

- Mục tiêu của việc lựa chọn đặc trưng:
  - Chọn các cột dữ liệu có liên quan nhất đến biến mục tiêu (lượt xem).
  - Giảm độ phức tạp của mô hình và cải thiện hiệu suất dự đoán.
- Các cột được chọn làm đặc trưng:
  - daily_rank: Thứ hạng của video trong ngày.
  - daily_movement: Thay đổi thứ hạng hàng ngày của video.
  - weekly_movement: Thay đổi thứ hạng hàng tuần của video.
  - like_count: Số lượng lượt thích.
  - comment_count: Số lượng bình luận.
- Lý do loại bỏ các cột khác: Các cột như title, description, thumbnail_url, và video_id không dễ dàng sử dụng trực tiếp trong mô hình và có thể yêu cầu các kỹ thuật xử lý phức tạp hơn (ví dụ: xử lý ngôn ngữ tự nhiên).

3.3.3. Chuyển Đổi Đặc Trưng Sang Dạng Vector

Mô tả chi tiết:

- Yêu cầu của Spark MLlib: Hầu hết các thuật toán học máy trong Spark MLlib yêu cầu dữ liệu đầu vào là một vector duy nhất chứa tất cả các đặc trưng.
- Sử dụng VectorAssembler:
  - VectorAssembler: Một công cụ trong Spark MLlib để kết hợp nhiều cột số thành một cột duy nhất chứa một vector.
- Mục đích của việc chuyển đổi:
  - Tương thích: Đảm bảo rằng dữ liệu đầu vào phù hợp với yêu cầu của các thuật toán học máy.
  - Tối ưu hóa: Có thể cải thiện hiệu suất tính toán trong một số trường hợp.

3.3.4. Mô Hình Hồi Quy Random Forest

Mô tả chi tiết:

- Lựa chọn thuật toán Random Forest:
  - Random Forest là một thuật toán học máy phổ biến và hiệu quả cho cả bài toán phân loại và hồi quy.
  - Nó có khả năng xử lý dữ liệu có nhiều đặc trưng và giảm thiểu overfitting.

- Các bước thực hiện:

1. Khởi tạo mô hình:
    - Tạo một đối tượng RandomForestRegressor, chỉ định các tham số như:
        - featuresCol: "features" (cột chứa vector đặc trưng).
        - labelCol: "view_count" (cột chứa biến mục tiêu).
        - numTrees: 50 (số lượng cây trong rừng).
        - maxDepth: 10 (độ sâu tối đa của mỗi cây).
        - seed: 42 (để đảm bảo tính tái lặp).
2. Huấn luyện mô hình:
    - Sử dụng phương thức fit() để huấn luyện mô hình trên tập dữ liệu huấn luyện.
3. Dự đoán trên tập kiểm thử:
    - Sử dụng phương thức transform() để tạo ra các dự đoán trên tập dữ liệu kiểm thử.

3.4. Đánh Giá Mô Hình

Mô tả chi tiết:

- Sử dụng RMSE để đánh giá:
    - RMSE (Root Mean Squared Error) là một độ đo phổ biến để đánh giá hiệu suất của các mô hình hồi quy.
    - RMSE đo lường độ lệch trung bình giữa giá trị dự đoán và giá trị thực tế. Giá trị RMSE càng nhỏ thì mô hình càng chính xác.
- Phương pháp tính RMSE:
    - Sử dụng RegressionEvaluator từ thư viện pyspark.ml.evaluation.
    - Chỉ định cột chứa giá trị thực tế (labelCol) và cột chứa giá trị dự đoán (predictionCol).
    - Sử dụng phương thức evaluate() để tính RMSE.
- Thông tin bổ sung: Tính giá trị trung bình của lượt xem trong tập dữ liệu để có một cơ sở so sánh cho giá trị RMSE.

3.5. Ứng Dụng

Mô tả chi tiết:

Các ứng dụng tiềm năng của mô hình dự đoán lượt xem video là rất đa dạng.

1. Đề xuất video:
    - Gợi ý cá nhân hóa: Đề xuất các video phù hợp với sở thích và lịch sử xem của từng người dùng, dựa trên dự đoán về lượt xem.

- Tối ưu hóa đề xuất trang chủ: Sắp xếp các video hiển thị trên trang chủ của YouTube để tăng khả năng thu hút người xem.

2. Tối ưu hóa chiến lược marketing:
    - Lựa chọn thời điểm đăng tải: Xác định thời điểm nào trong ngày/tuần là tốt nhất để đăng tải video, dựa trên dự đoán về lượt xem.
    - Tối ưu hóa tiêu đề và mô tả: Sử dụng mô hình để đánh giá tiềm năng thu hút của các tiêu đề và mô tả khác nhau.
    - Chọn từ khóa: Xác định các từ khóa phù hợp để tăng khả năng hiển thị của video trong kết quả tìm kiếm.
3. Phân tích xu hướng:
    - Xác định các chủ đề thịnh hành: Phân tích các đặc điểm chung của các video có lượt xem cao để xác định các chủ đề đang được quan tâm.
    - Dự đoán xu hướng: Sử dụng mô hình để dự báo các xu hướng video trong tương lai, giúp nhà sáng tạo nội dung chủ động tạo ra các nội dung hấp dẫn.
4. Phát hiện gian lận:
    - Phát hiện lượt xem ảo: Xác định các video có lượt xem bất thường so với các đặc điểm khác (ví dụ: tỷ lệ thích/xem quá cao, số lượng bình luận thấp), có thể là dấu hiệu của hành vi gian lận.
    - Cải thiện tính công bằng: Giúp YouTube duy trì một môi trường cạnh tranh công bằng cho tất cả các nhà sáng tạo nội dung.

3.6. Kết Luận

Bài toán dự đoán lượt xem video trên YouTube là một thử thách phức tạp, đòi hỏi phải kết hợp nhiều kỹ thuật khác nhau từ phân tích dữ liệu đến máy học. Mặc dù mô hình Random Forest hiện tại có giá trị RMSE khá cao, nhưng nó cung cấp một nền tảng quan trọng cho các nghiên cứu trong tương lai. Để cải thiện hiệu suất dự đoán, có thể thử nghiệm các phương pháp sau:

- Sử dụng mô hình phức tạp hơn: Gradient Boosted Trees, mạng nơ-ron, các mô hình kết hợp.
- Thêm các đặc trưng khác: Chủ đề video, phân tích cảm xúc, thông tin về kênh YouTube.
- Tinh chỉnh siêu tham số: Tìm kiếm các giá trị siêu tham số tối ưu cho mô hình.
- Sử dụng các kỹ thuật xử lý dữ liệu tiên tiến: Biến đổi dữ liệu để xử lý sự phân phối lệch, áp dụng các phương pháp feature scaling.

KẾT LUẬN

Báo cáo đã trình bày các bước xử lý, phân tích dữ liệu và ứng dụng trong thực tế. Việc tiền xử lý giúp làm sạch và chuẩn hóa dữ liệu, đảm bảo tính chính xác trước khi phân tích. Sau đó, phân tích thống kê và khai phá dữ liệu cung cấp thông tin hữu ích về xu hướng video thịnh hành, mức độ tương tác và sự khác biệt theo quốc gia. Những kết quả này có thể hỗ trợ nhà sáng tạo nội dung, doanh nghiệp quảng cáo và thuật toán gợi ý video trên YouTube.

Ưu điểm

- Giúp chuẩn hóa và làm sạch dữ liệu, cải thiện chất lượng phân tích.
- Cung cấp thông tin chi tiết về xu hướng nội dung thịnh hành trên YouTube.
- Xác định các yếu tố ảnh hưởng đến sự tương tác của video, hỗ trợ chiến lược nội dung.

Nhược điểm

- Dữ liệu thu thập có thể bị giới hạn do API hoặc chính sách nền tảng.
- Một số dữ liệu có thể không đầy đủ, ảnh hưởng đến kết quả phân tích.

Cách cải tiến

- Kết hợp nhiều nguồn dữ liệu khác nhau để bổ sung dữ liệu bị thiếu.
- Áp dụng các mô hình AI nâng cao để phân tích chi tiết hơn.
- Tự động hóa quy trình thu thập và tiền xử lý dữ liệu để đảm bảo tính liên tục.

DANH MỤC TÀI LIỆU THAM KHẢO

1. YouTube API Documentation: https://developers.google.com/youtube/v3
2. Scikit-learn Documentation: https://scikit-learn.org/stable/
3. Pandas Library: https://pandas.pydata.org/
4. Matplotlib Visualization: https://matplotlib.org/
5. Machine Learning and Data Science References