

TRƯỜNG ĐẠI HỌC TÀI NGUYÊN VÀ MÔI TRƯỜNG
THÀNH PHỐ HỒ CHÍ MINH
Độc lập – Tự do – Hạnh Phúc



HỌC PHẦN: CÔNG NGHỆ DỮ LIỆU LỚN
Đề tài: Thống kê sự quan tâm của người dùng facebook
đối với một số sản phẩm

Giảng viên hướng dẫn : ThS. Phạm Trọng Huỳnh

Khoá : 10

Lớp : Công nghệ thông tin 3

Thành viên nhóm : Lâm Thị Phương Thảo (Nhóm trưởng)

Võ Minh Hân

Trà Ngọc Thông

TP. Hồ Chí Minh, tháng 10 năm 2023

TRƯỜNG ĐẠI HỌC TÀI NGUYÊN VÀ MÔI TRƯỜNG
THÀNH PHỐ HỒ CHÍ MINH
Độc lập – Tự do – Hạnh Phúc



BÁO CÁO CÔNG NGHỆ DỮ LIỆU LỚN
Đề tài: Thống kê sự quan tâm của người dùng facebook
đối với một số sản phẩm

Giảng viên hướng dẫn : ThS. Phạm Trọng Huỳnh

Khoá : 10

Lớp : Công nghệ thông tin 3

Thành viên nhóm : Lâm Thị Phương Thảo (Nhóm trưởng)

Võ Minh Hân



Trà Ngọc Thông

TP. Hồ Chí Minh, tháng 10 năm 2023

MỤC LỤC

THÔNG TIN NHÓM	4
I. BÁO CÁO TÓM TẮT	6
1. Lý do chọn đề tài	6
1.1 Tầm quan trọng	7
1.2 Tính ứng dụng cao của thống kê dữ liệu	7
1.3 Tích hợp Apache Spark	8
1.4 Áp dụng Machine Learning Regression (Hồi quy tuyến tính)	9
2 Đề tài phù hợp với môn học, đáp ứng các vấn đề 5V của xử lý dữ liệu lớn	10
3 Có thể ứng dụng đề tài thực tế làm dữ liệu thống kê cho lĩnh vực content marketing	11
3.1 Lợi ích của thống kê dữ liệu	11
3.2 Ứng dụng cụ thể:	12
3.3 Các nội dung và kết quả:	12
II. BÁO CÁO CHI TIẾT	13
1. Big Data	13
1.1 Định nghĩa	13
1.2 Lịch sử	13
1.3 Đặc trưng [2]	14
3.2 Các công cụ được dùng trong Big Data	17
3.3 Ứng dụng của Big Data [5]	29
2. Bài toán	39
3. Giải pháp	41
3.1 Mô hình giải pháp	42
3.2 Cài đặt thử nghiệm	44
III. TỔNG KẾT	79
1. Đề xuất	79
2. Đánh giá	79
2.1 Ưu điểm	79
2.2 Nhược điểm	79
3. Kết luận	79
TÀI LIỆU THAM KHẢO	82

THÔNG TIN NHÓM

ĐỀ TÀI	THỐNG KÊ SỰ QUAN TÂM CỦA NGƯỜI DÙNG FACEBOOK ĐỐI VỚI MỘT SỐ SẢN PHẨM	Đóng góp của các thành viên
Thành viên 1	<p>Lâm Thị Phương Thảo – 1050080118</p> 	<p>Khái quát mô hình, cài đặt thực nghiệm.</p> <p>Điểm tự đánh giá quá trình: 9</p> <p>Điểm tự đánh giá cuối kỳ: 9</p>
Thành viên 2	<p>Võ Minh Hân – 1050080096</p> 	<p>Support cài đặt thực nghiệm.</p> <p>Điểm tự đánh giá quá trình: 7</p> <p>Điểm tự đánh giá cuối kỳ: 7</p>

Thành viên 3	<p>Trà Ngọc Thông – 1050080120</p> 	<p>Support cài đặt thực nghiệm. Điểm tự đánh giá quá trình: 7 Điểm tự đánh giá cuối kỳ: 7</p>
---------------------	--	---

I. BÁO CÁO TÓM TẮT

1. Lý do chọn đề tài

Trong thời đại số ngày nay, dữ liệu xã hội đóng vai trò quan trọng trong việc hiểu biết hành vi và quan tâm của người dùng. Facebook, với hàng tỷ người dùng trên toàn thế giới, là một nguồn dữ liệu quý báu, giúp chúng ta phân tích sự tương tác của người dùng với nhiều nội dung, bao gồm cả sản phẩm và quảng cáo. Điều này đặt ra một loạt các cơ hội và thách thức trong việc tận dụng dữ liệu lớn (Big Data) để cải thiện sản phẩm và dịch vụ, cũng như tối ưu hóa chiến dịch quảng cáo.

Dữ liệu xã hội và vai trò quan trọng: Dữ liệu xã hội là một loại dữ liệu đặc biệt quan trọng trong nghiên cứu về hành vi của con người. Nó bao gồm các tương tác của người dùng trên các nền tảng mạng xã hội như Facebook, bao gồm việc chia sẻ nội dung, bình luận, thích, kết bạn, và nhiều hoạt động khác. Những dữ liệu này có giá trị lớn đối với các doanh nghiệp và tổ chức vì chúng cung cấp thông tin về sở thích, tư duy và hành vi của người dùng.

Phân tích hành vi người dùng: Một trong những ứng dụng quan trọng của dữ liệu xã hội là phân tích hành vi người dùng. Các công ty như Facebook có khả năng theo dõi sự tương tác của người dùng với nhiều nội dung trên nền tảng của họ. Điều này có thể giúp họ hiểu rõ hơn về những gì người dùng yêu thích, những vấn đề quan tâm, và cách họ tương tác với sản phẩm và dịch vụ cụ thể. Phân tích dữ liệu xã hội có thể giúp tối ưu hóa trải nghiệm người dùng và cải thiện sản phẩm thông qua việc tạo ra tính năng mới hoặc tối ưu hóa các tính năng hiện có.

Tiềm năng trong quảng cáo: Dữ liệu xã hội cũng cung cấp cơ hội tuyệt vời cho quảng cáo. Bằng cách hiểu rõ hơn về sở thích và hành vi của người dùng, các doanh nghiệp có thể tạo ra chiến dịch quảng cáo có độ chính xác cao hơn. Họ có thể định rõ mục tiêu đối tượng, tối ưu hóa quảng cáo dựa trên dữ liệu thời gian thực, và đảm bảo rằng thông điệp quảng cáo đạt được người mà nó nhắm

đến. Điều này có thể giảm thiểu lãng phí quảng cáo và tăng cường hiệu suất chiến dịch.

Thách thức về quản lý dữ liệu lớn: Tuy dữ liệu xã hội có tiềm năng lớn, nhưng cũng đặt ra nhiều thách thức về quản lý dữ liệu lớn. Hàng tỷ người dùng trên mạng xã hội tạo ra một lượng dữ liệu khổng lồ hàng ngày. Quá trình thu thập, lưu trữ, và bảo mật dữ liệu này đòi hỏi cơ sở hạ tầng mạng lớn và sự đầu tư vào công nghệ xử lý dữ liệu. Bên cạnh đó, bảo mật dữ liệu và bảo vệ quyền riêng tư của người dùng cũng là một vấn đề quan trọng đối với các doanh nghiệp xã hội.

Kết luận: Dữ liệu xã hội đã trở thành một nguồn thông tin quý báu trong thời đại số, đặc biệt đối với các mạng xã hội lớn như Facebook. Việc sử dụng dữ liệu lớn từ mạng xã hội có tiềm năng để nâng cao hiệu suất kinh doanh, cải thiện sản phẩm và dịch vụ, và tối ưu hóa chiến dịch quảng cáo. Tuy nhiên, để tận dụng được tiềm năng của dữ liệu xã hội, cần phải đối mặt với nhiều thách thức về quản lý và bảo mật dữ liệu lớn.

Chính vì những lý do trên đã đưa nhóm đi đến quyết định chọn đề tài ***Thống kê sự quan tâm của người dùng Facebook đối với một số sản phẩm.***

1.1 Tầm quan trọng

Trong thời đại ngày nay, dữ liệu trở thành một tài nguyên vô cùng quan trọng và big data đã trở thành một lĩnh vực nghiên cứu ngày càng phổ biến. Việc thực hiện đề tài ***"Thống kê Sự Quan Tâm của Người Dùng Facebook đối với Một Số Sản Phẩm"*** không chỉ là thách thức mà còn là cơ hội để áp dụng và hiểu rõ các nguyên lý và kỹ thuật liên quan đến dữ liệu lớn.

1.2 Tính ứng dụng cao của thống kê dữ liệu

Sử dụng kỹ thuật thống kê để phân tích sự quan tâm của người dùng đối với các sản phẩm không chỉ mang lại cái nhìn tổng quan về thị trường mà còn giúp dự đoán xu hướng và đưa ra chiến lược kinh doanh hiệu quả.

1.3 Tích hợp Apache Spark

Apache Spark là một nền tảng tính toán phân tán mạnh mẽ và đa năng, đóng vai trò quan trọng trong việc xử lý dữ liệu lớn từ nhiều nguồn khác nhau, bao gồm dữ liệu từ Facebook. Spark đã trở thành một công cụ quan trọng cho các dự án Big Data và có nhiều ưu điểm quan trọng:

Xử lý dữ liệu lớn hiệu quả: Spark được thiết kế để xử lý dữ liệu lớn một cách hiệu quả hơn so với các giải pháp truyền thống. Khả năng tích hợp với hệ thống phân tán mạnh mẽ như Hadoop Distributed File System (HDFS) giúp Spark chia tải công việc tính toán trên nhiều máy tính, tận dụng tài nguyên đám mây hoặc trung tâm dữ liệu để đảm bảo hiệu suất cao và thời gian đáp ứng nhanh chóng.

Đa năng và tích hợp dữ liệu đa nguồn: Spark hỗ trợ nhiều ngôn ngữ lập trình, bao gồm Scala, Java, Python và R, giúp các nhà phân tích dữ liệu lựa chọn ngôn ngữ ưa thích của họ. Điều này giúp tích hợp dễ dàng với các dự án sử dụng ngôn ngữ khác nhau và truy cập đa dạng các nguồn dữ liệu. Với tích hợp linh hoạt, Spark cho phép trích xuất, biến đổi và tải dữ liệu từ Facebook cũng như các nguồn khác một cách thuận tiện.

Đáng tin cậy và khả năng mở rộng: Spark được phát triển để đảm bảo tính đáng tin cậy trong xử lý dữ liệu lớn. Hệ thống kiến trúc và cách thức hoạt động của Spark đã được kiểm chứng trong các dự án lớn, giúp tạo ra kết quả mà bạn có thể tin tưởng. Ngoài ra, Spark có khả năng mở rộng linh hoạt, nghĩa là bạn có thể tận dụng các tài nguyên bổ sung để xử lý dữ liệu lớn hơn và đáp ứng nhu cầu thay đổi.

Tăng cường khả năng phân tích và dự đoán: Khi tích hợp Spark vào dự án, bạn có khả năng thực hiện phân tích phức tạp và dự đoán dựa trên dữ liệu Facebook một cách hiệu quả. Mô hình Machine Learning có thể được xây dựng và đào tạo trên dữ liệu lớn một cách nhanh chóng và đơn giản, giúp bạn hiểu

rõ hơn về hành vi người dùng, dự đoán xu hướng và tối ưu hóa quyết định kinh doanh.

Tóm lại, tích hợp Apache Spark vào dự án Big Data là một quyết định thông minh. Nền tảng mạnh mẽ và tích hợp đa ngôn ngữ giúp bạn xử lý dữ liệu lớn một cách hiệu quả, cung cấp kết quả nhanh chóng và đáng tin cậy, và tạo cơ hội tối ưu hóa phân tích và dự đoán dữ liệu từ Facebook và các nguồn khác.

1.4 Áp dụng Machine Learning Regression (Hồi quy tuyến tính)

Mô hình hồi quy tuyến tính là một trong những phương pháp phổ biến trong Machine Learning và phân tích dữ liệu, được sử dụng để dự đoán giá trị đầu ra dựa trên mối quan hệ tuyến tính giữa các biến đầu vào và đầu ra. Khi áp dụng vào lĩnh vực Big Data, mô hình hồi quy tuyến tính trở nên mạnh mẽ và quan trọng hơn bao giờ hết.

a. Xây dựng mô hình hồi quy tuyến tính trong Big Data

Dữ liệu lớn từ nhiều nguồn, bao gồm các dữ liệu liên quan đến sản phẩm và dữ liệu người dùng trên các nền tảng xã hội như Facebook, có thể được khai thác bằng mô hình hồi quy tuyến tính để đưa ra dự đoán, giải thích và tối ưu hóa nhiều khía cạnh khác nhau:

Dự đoán giá trị sản phẩm: Trong lĩnh vực thương mại điện tử, mô hình hồi quy tuyến tính có thể được sử dụng để dự đoán giá trị của sản phẩm dựa trên các biến đầu vào như đặc điểm sản phẩm, thời gian, mức độ quảng cáo, và phản hồi từ người dùng. Điều này có thể giúp các doanh nghiệp xác định giá sản phẩm tối ưu để tối đa hóa lợi nhuận.

Phân tích hành vi người dùng trên mạng xã hội: Mô hình hồi quy tuyến tính có thể giúp hiểu rõ mối quan hệ giữa các yếu tố như số lượt thích, bình luận, chia sẻ và sự phát triển của nội dung trên các nền tảng xã hội. Điều này

có thể giúp định rõ những nội dung nào thu hút nhiều sự quan tâm và tương tác từ người dùng, giúp các doanh nghiệp tạo ra nội dung hiệu quả hơn.

Tối ưu hóa chiến dịch quảng cáo: Bằng cách sử dụng dữ liệu lớn từ các chiến dịch quảng cáo trước đây, mô hình hồi quy tuyến tính có thể dự đoán hiệu suất của chiến dịch quảng cáo tương lai. Điều này cho phép các doanh nghiệp tối ưu hóa việc chọn đối tượng, quản lý ngân sách, và thiết kế quảng cáo để đạt được kết quả tốt nhất.

Dự đoán xu hướng và sự thay đổi trong thời gian: Mô hình hồi quy tuyến tính có thể được sử dụng để dự đoán xu hướng trong dữ liệu xã hội theo thời gian. Điều này có thể giúp các doanh nghiệp dự đoán sự thay đổi trong hành vi của người dùng và thích nghi nhanh chóng.

2 Đề tài phù hợp với môn học, đáp ứng các vấn đề 5V của xử lý dữ liệu lớn

Khối Lượng (Volume): Đề tài đề cập đến sự quan tâm của người dùng Facebook đối với sản phẩm, có thể tạo ra một lượng dữ liệu lớn từ các tương tác, bình luận, và chia sẻ. Việc xử lý lượng dữ liệu lớn này sẽ đặt ra thách thức về hiệu suất và khả năng lưu trữ, điều mà môn học Xử Lý Dữ Liệu Lớn giúp sinh viên hiểu rõ và vận dụng.

Tốc Độ (Velocity): Dữ liệu trên Facebook được tạo ra và cập nhật liên tục. Quá trình thu thập và phân tích nhanh chóng sẽ đòi hỏi kỹ thuật xử lý dữ liệu thời gian thực. Xử lý sự thay đổi nhanh chóng này là một trong những khía cạnh mà sinh viên sẽ có cơ hội nắm bắt trong môn học.

Đa Dạng (Variety): Dữ liệu trên Facebook không chỉ bao gồm văn bản mà còn hình ảnh, video, và nhiều định dạng khác nhau. Việc làm thế nào để hiệu quả xử lý và phân tích dữ liệu đa dạng là một vấn đề quan trọng mà môn học Xử Lý Dữ Liệu Lớn có thể giúp sinh viên giải quyết.

Chính Xác (Accuracy): Sự chính xác của dữ liệu là quan trọng để đảm bảo kết quả phân tích đúng đắn. Môn học có thể giúp sinh viên hiểu về các phương pháp và công cụ để kiểm soát và cải thiện chất lượng dữ liệu.

Tính Tương Tác (Veracity): Dữ liệu từ mạng xã hội có thể bao gồm thông tin không chính xác hoặc thiếu chính xác. Việc giải quyết vấn đề này đòi hỏi sự hiểu biết vững về xử lý dữ liệu không chắc chắn, một khía cạnh mà môn học có thể chú trọng.

Đề tài này không chỉ đáp ứng đầy đủ các yếu tố 5V của Big Data mà còn mở rộng kiến thức và kỹ năng của sinh viên về cách áp dụng các nguyên lý xử lý dữ liệu lớn trong bối cảnh thực tế của thị trường kỹ thuật số và mạng xã hội ngày nay.

3 Có thể ứng dụng đề tài thực tế làm dữ liệu thống kê cho lĩnh vực content marketing

Lĩnh vực Content Marketing đang trở thành một phần quan trọng trong chiến lược tiếp thị của các doanh nghiệp. Việc hiểu rõ sự quan tâm của người dùng đối với các sản phẩm qua nền tảng xã hội như Facebook sẽ giúp tối ưu hóa chiến lược Content Marketing, từ đó tăng cường tương tác và chuyển đổi.

3.1 Lợi ích của thống kê dữ liệu

3.1.1 Đối tượng khách hàng chính xác:

Thông qua phân tích dữ liệu, ta có thể xác định đối tượng khách hàng cụ thể quan tâm đến sản phẩm nào. Content Marketing có thể được tối ưu hóa để đáp ứng nhu cầu và mong muốn của đối tượng này.

3.1.2 Nghiên cứu xu hướng và sở thích:

Hiểu rõ xu hướng và sở thích của người dùng giúp xây dựng nội dung phong phú và hấp dẫn. Các chiến lược tiếp thị có thể được điều chỉnh để phản ánh đúng yêu cầu thị trường.

3.2 Ứng dụng cụ thể:

Dựa trên dữ liệu về sự quan tâm của người dùng, nội dung có thể được tối ưu hóa để chứa đựng thông điệp thuận lợi và hấp dẫn đối tượng mục tiêu. Kết hợp thông tin từ thống kê để xác định các sản phẩm hot, giúp tạo ra chiến lược quảng cáo hiệu quả. Tiếp cận người dùng với nội dung mà họ quan tâm tăng cơ hội chuyển đổi. Dữ liệu về sự quan tâm của người dùng có thể hỗ trợ việc lập kế hoạch chi tiết cho chiến dịch tiếp thị trên nền tảng Facebook. Tăng khả năng tương tác và chia sẻ từ cộng đồng người hâm mộ.

3.3 Các nội dung và kết quả:

- Mô tả bài toán.
- Đưa ra giải pháp.
- Đưa ra mô hình xử lý cho giải pháp được đề ra.
- Các công nghệ được sử dụng cho mô hình.
- Cài đặt các công nghệ.
- Demo.
- Đánh giá.

II. BÁO CÁO CHI TIẾT

1. Big Data

1.1 Định nghĩa



[1] Big Data không chỉ đánh dấu sự thay đổi trong cách chúng ta làm việc và tương tác với thông tin, mà còn mở ra một loạt cơ hội và thách thức cho nhiều lĩnh vực khác nhau, bao gồm công nghiệp, khoa học, chính trị, và xã hội học. Trong luận văn này, chúng ta sẽ khám phá sâu hơn về khái niệm Big Data, ý nghĩa của nó, và những ảnh hưởng to lớn mà nó mang lại.

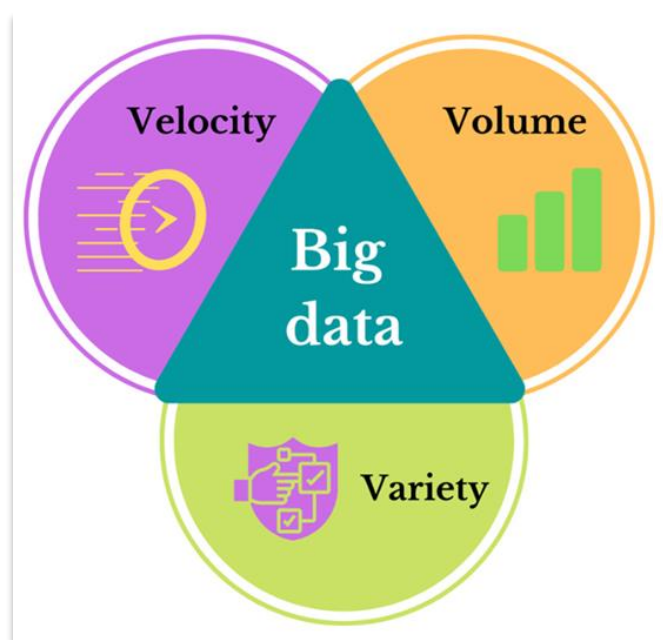
1.2 Lịch sử

Dữ án dữ liệu lớn đầu tiên được tạo ra vào năm 1937 và được theo lệnh của chính quyền Franklin D. Roosevelt trong HOA KỲ. Máy xử lý dữ liệu đầu tiên xuất hiện vào năm 1943 và được người Anh phát triển để giải mã mật mã của Đức Quốc xã trong lúc Chiến tranh Thế giới II. Thiết bị này có tên là Colossus, đã tìm kiếm mẫu trong tin nhắn bị chặn ở tỷ lệ 5.000 ký tự trên giây. Kể từ những năm 90 việc tạo ra dữ liệu được thúc đẩy khi ngày càng có nhiều thiết bị được kết nối với Internet. Sau khi siêu máy tính đầu tiên được chế tạo, nó không có thể xử lý dữ liệu có bản chất khác nhau lưu trữ, kích thước và định dạng. Dữ liệu lớn tạo ra thách thức mới trong bàn giao dữ liệu và tạo ra thông tin hữu ích từ đó. Trong năm 2005, khi phần mềm Hadoop được phát triển bởi yahoo được xây dựng dựa trên MapReduce của Google. Mục đích là

lập chỉ mục cho toàn bộ World Wide Web. Ngày nay, việc mở nguồn Hadoop được rất nhiều tổ chức sử dụng để giải quyết vấn đề lượng dữ liệu khổng lồ. Nhiều tổ chức chính phủ đang sử dụng dữ liệu lớn để tìm ra sự hỗ trợ quyết định hữu ích cho sự cải thiện xã hội từ đó. Năm 2009 Ấn Độ Chính phủ đã khởi động dự án mang tên AADHAAR để chụp quét móng mắt, lấy dấu vân tay và chụp ảnh tất cả 1.32 của nó tỷ dân. Tất cả dữ liệu này được lưu trữ trong thư mục lớn nhất cơ sở dữ liệu sinh trắc học trên thế giới. Gần đây, người Ấn Độ .Chính phủ đã bắt đầu dự án dữ liệu lớn để tìm hiểu người vi phạm thuế thu nhập sử dụng phương tiện truyền thông xã hội (dữ liệu facebook và Twitter) vào năm 2017.

1.3 Đặc trưng [2]

3.1.1 3V'S



3.4 Volume - Khối lượng dữ liệu

Big data là thuật ngữ nói về khối lượng dữ liệu lớn, kích thước lớn. Xác định giá trị của dữ liệu và kích thước dữ liệu là rất quan trọng và cần thiết, nếu khối lượng lớn, đó chính là Big data.

Volume là khối lượng dữ liệu được các doanh nghiệp thu thập từ các nguồn khác nhau, như IoT (Internet of Things), video, giao dịch kinh doanh, các phương tiện truyền thông xã hội,...

Khi công nghệ chưa có sự phát triển vượt bậc, việc lưu trữ lượng lớn dữ liệu là một thách thức lớn. Tuy nhiên ngày nay, các nền tảng lưu trữ giá thành rẻ như Hadoop và Data lake xuất hiện, việc lưu trữ đã trở nên dễ dàng hơn nhiều.

3.5 Velocity - Tốc độ xử lý

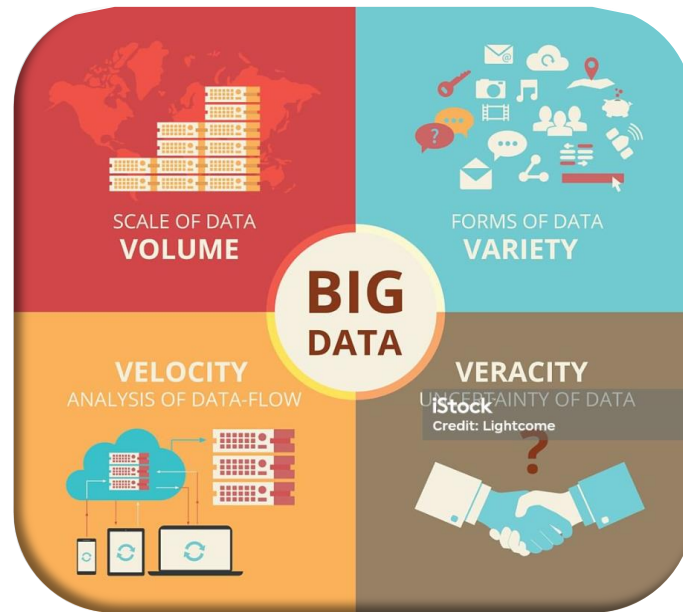
Dựa vào tốc độ xử lý của luồng dữ liệu để xác định đó có phải là Big data hay không. Thường thì tốc độ của luồng dữ liệu trực tiếp vào bộ nhớ cao hơn so với khi được ghi vào đĩa. Đặc biệt là ngày nay, với sự phát triển của IoT, các luồng dữ liệu truyền tải với tốc độ cực nhanh và chúng phải được xử lý kịp thời.

Ví dụ: Trên mạng xã hội Facebook, các thông báo như status, tweet,... đã cũ sẽ không được người dùng quan tâm và bị quên lãng nhanh chóng. Dữ liệu giờ đây được tính gần như vào thời gian thực và tốc độ cập nhật thông tin dường như giảm xuống đơn vị mili giây.

3.6 Variety - Tính đa dạng, linh hoạt

Đặc trưng tiếp theo của Big data chính là tính đa dạng, linh hoạt, ở dạng cấu trúc và phi cấu trúc, bao gồm dữ liệu số, Email, Video, âm thanh, giao dịch tài chính,... Tính đa dạng ảnh hưởng đến hiệu suất, đây là một trong những vấn đề chính mà lĩnh vực Big data cần phải giải quyết.

3.1.2 4V'S



Tương tự như 3V'S và có sự thay đổi là có thêm Veracity:

a. Veracity

Veracity sẽ đề cập đến cả chất lượng và tính sẵn có của dữ liệu. Khi nói đến phân tích kinh doanh truyền thống, nguồn dữ liệu sẽ nhỏ hơn nhiều cả về số lượng và sự đa dạng. Tuy nhiên, tổ chức sẽ có nhiều quyền kiểm soát hơn đối với chúng và tính xác thực của chúng sẽ cao hơn.

Khi chúng ta nói về Big Data sự đa dạng đồng nghĩa với việc không chắc chắn hơn về chất lượng của dữ liệu đó và tính sẵn có của nó. Nó cũng sẽ có tác động về mặt nguồn dữ liệu mà chúng tôi có thể có.

3.1.3 5V'S



a. Variability (Độ chính xác):

Vì đa dạng về các kiểu dữ liệu, nên sự không thống nhất của tập dữ liệu có thể cản trở các quy trình để xử lý và quản lý nó. Do đó, độ chính xác của công nghệ này có thể đảm bảo giúp cho việc giảm bớt sự sai lệch đáng tiếc có thể xảy ra.

b. Value (Mức độ giá trị của thông tin)

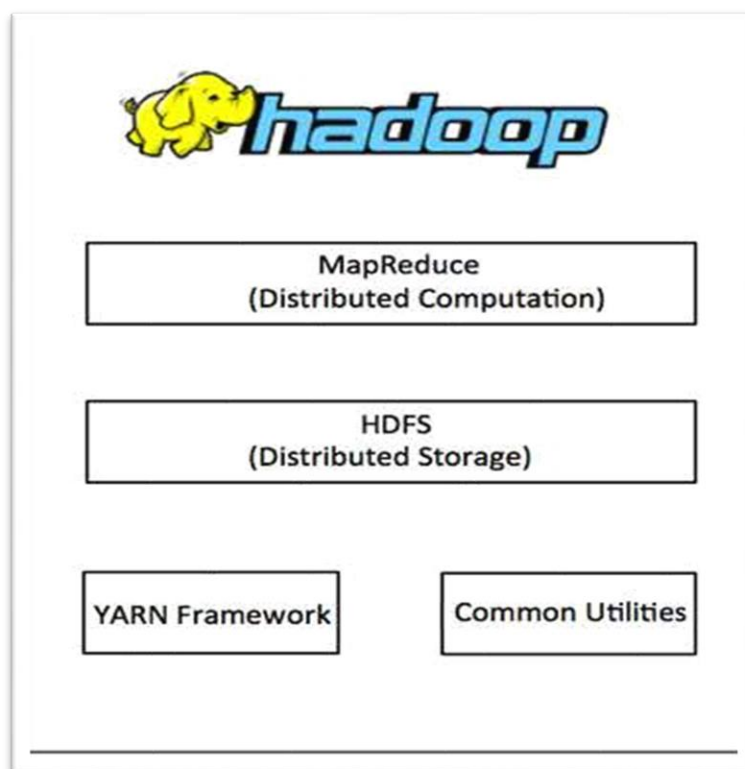
Chất lượng dữ liệu của những dữ liệu lấy được có thể thay đổi rất nhiều, điều này sẽ ảnh hưởng rất mạnh đến việc phân tích chính xác những đây. Ta có thể xem đây là tính chất cũng là khái niệm mà những doanh nghiệp hay nhà nghiên cứu muốn sử dụng và khai thác Big Data phải nắm giữ và am hiểu nó đầu tiên.

3.2 Các công cụ được dùng trong Big Data

Big Data sử dụng nhiều công cụ để xử lý, lưu trữ và phân tích dữ liệu lớn. Dưới đây là một số công nghệ quan trọng trong lĩnh vực Big Data:

3.2.1 Hadoop:

Hadoop là một framework mã nguồn mở quan trọng cho lưu trữ và xử lý dữ liệu lớn. Nó bao gồm Hadoop Distributed File System (HDFS) cho việc lưu trữ dữ liệu và MapReduce cho xử lý dữ liệu song song và Yarn giúp quản lý và phân phối tài nguyên tính toán trong cụm Hadoop.



a. Hadoop Distributed File System (HDFS):

HDFS là một phần quan trọng của Hadoop. Nó được thiết kế để lưu trữ dữ liệu lớn, phân tán trên nhiều máy tính trong một cụm. Mô hình dữ liệu của HDFS chia dữ liệu thành các khối nhỏ (thường là 128MB hoặc 256MB) và nhân bản chúng trên nhiều nút. Điều này đảm bảo sự an toàn và sẵn sàng của dữ liệu. HDFS cung cấp khả năng xử lý dữ liệu lớn mà không cần phải lo lắng về việc lưu trữ trên một máy tính duy nhất.

Phân tán và Lưu trữ dữ liệu: HDFS chia dữ liệu thành các khối nhỏ hơn, thường là 128MB hoặc 256MB, để quản lý dễ dàng và tận dụng hiệu quả các

máy tính trong cụm Hadoop. Một tệp lớn được chia thành nhiều khối và các khối này được phân phối trên nhiều nút trong cụm.

Dữ liệu trong HDFS được nhân bản (replication) trên nhiều nút. Mặc định, mỗi khối dữ liệu được nhân bản thành 3 bản sao để đảm bảo tính an toàn và sẵn sàng của dữ liệu. Nếu một nút gặp sự cố, HDFS vẫn có thể truy cập dữ liệu từ các bản sao trên các nút khác.

Phục hồi lỗi tự động: HDFS được thiết kế với khả năng tự động phục hồi lỗi. Nếu một nút gặp sự cố hoặc một bản sao dữ liệu bị hỏng, HDFS sẽ tự động thay thế nó bằng bản sao dữ liệu khác. Điều này đảm bảo tính an toàn và độ tin cậy của dữ liệu.

Giao thức dữ liệu chuyên dụng: HDFS sử dụng một giao thức dữ liệu chuyên dụng để truy cập và quản lý dữ liệu. Giao thức này được gọi là Hadoop Distributed File System Protocol (HDFS Protocol). Nó cho phép ứng dụng đọc và ghi dữ liệu trực tiếp vào HDFS.

Khả năng mở rộng: HDFS có khả năng mở rộng dễ dàng. Bạn có thể thêm nút vào cụm Hadoop để tăng khả năng lưu trữ và xử lý dữ liệu mà không cần phải ngừng hoạt động của hệ thống. Điều này làm cho HDFS phù hợp cho môi trường Big Data, nơi lưu lượng dữ liệu có thể tăng lên đáng kể theo thời gian.

Hiệu suất đọc và ghi: HDFS được tối ưu hóa để hiệu suất đọc và ghi dữ liệu lớn. Dữ liệu được lưu trữ trên nhiều nút, cho phép đọc và ghi song song, giúp cải thiện hiệu suất xử lý dữ liệu lớn.

Tích hợp với Hadoop Ecosystem: HDFS là phần quan trọng của framework Hadoop. Nó hoạt động cùng với các thành phần khác của Hadoop như MapReduce, YARN và nhiều công cụ và thư viện khác để xử lý và phân tích dữ liệu lớn.

HDFS là một phần không thể thiếu của hệ thống Hadoop và chính thức hoạt động như một hệ thống lưu trữ phân tán cho Big Data, đảm bảo tính an toàn, sẵn sàng và hiệu suất cao cho việc quản lý và xử lý dữ liệu lớn.

b. MapReduce:

MapReduce là một mô hình lập trình và một framework xử lý dữ liệu lớn được phát triển bởi Google và sau đó được triển khai trong hệ thống Hadoop. Nó là một phần quan trọng của hệ thống Hadoop, giúp xử lý dữ liệu lớn hiệu quả bằng cách chia công việc thành các phần nhỏ và thực hiện chúng trên nhiều máy tính đồng thời. Dưới đây là một số chi tiết quan trọng về MapReduce trong lĩnh vực Big Data:

Mô hình MapReduce:

- *Phần Map:* Trong phần này, dữ liệu đầu vào được chia thành các phần nhỏ hơn và các xử lý Map được áp dụng lên từng phần. Kết quả là các cặp (key, value) được tạo ra. Phần Map có thể chạy độc lập trên nhiều máy tính và cho phép phân tán xử lý dữ liệu.
- *Phần Reduce:* Phần này nhận các cặp (key, value) được tạo bởi phần Map. Các giá trị có cùng key được tổng hợp lại. Phần Reduce cũng có thể chạy độc lập trên nhiều máy tính, giúp tạo ra kết quả cuối cùng sau khi tổng hợp dữ liệu từ các phần Map.

Mô hình này dựa trên nguyên tắc chia để trị, cho phép xử lý dữ liệu lớn bằng cách phân chia nó thành các phần nhỏ hơn và sau đó tổng hợp kết quả cuối cùng.

Phân phối dữ liệu và tính toán: MapReduce cho phép phân phối dữ liệu và tính toán trên nhiều máy tính. Điều này giúp tận dụng sức mạnh tính toán và lưu trữ của một cụm máy tính, cho phép xử lý dữ liệu lớn nhanh chóng.

Tích hợp với Hadoop: MapReduce là một phần quan trọng của framework Hadoop. Nó được sử dụng để xử lý dữ liệu trên các nút trong cụm Hadoop. Dữ liệu được lưu trữ trong Hadoop Distributed File System (HDFS), và MapReduce được sử dụng để thực hiện các tác vụ xử lý dữ liệu trên dữ liệu này.

Khả năng mở rộng và sự linh hoạt: Một trong những điểm mạnh của MapReduce là khả năng mở rộng. Bạn có thể dễ dàng thêm máy tính vào cụm Hadoop để tăng khả năng lưu trữ và xử lý dữ liệu. Điều này giúp đáp ứng nhu cầu tăng lên của lưu lượng dữ liệu mà không cần phải thay đổi cấu trúc của ứng dụng MapReduce.

Sử dụng đa dạng: Mặc dù MapReduce ban đầu được thiết kế cho việc xử lý lô dữ liệu (batch processing), nó đã được sử dụng rộng rãi cho nhiều loại ứng dụng khác nhau, bao gồm xử lý luồng dữ liệu thời gian thực (real-time data processing) và xử lý dữ liệu đồ thị.

MapReduce là một công cụ quan trọng trong lĩnh vực Big Data, cho phép xử lý và phân tích dữ liệu lớn một cách hiệu quả và mở rộng.

c. Yarn

YARN (Yet Another Resource Negotiator) là một thành phần quan trọng trong hệ thống Hadoop, giúp quản lý và phân phối tài nguyên tính toán trong cụm Hadoop. YARN ra đời nhằm giải quyết nhược điểm của phiên bản trước đó của Hadoop MapReduce, nơi MapReduce Framework đảm nhiệm cả việc quản lý tài nguyên và xử lý dữ liệu. Việc tổng hợp quản lý tài nguyên và xử lý dữ liệu trong một framework duy nhất đã gây ra nhiều hạn chế, nhất là trong việc chia sẻ và quản lý tài nguyên hiệu quả.

Với YARN, kiến trúc của Hadoop được chia thành hai phần chính:

- **ResourceManager:** ResourceManager là thành phần trung tâm của YARN, chịu trách nhiệm quản lý tài nguyên trong cụm Hadoop. Nó theo dõi tài nguyên có sẵn trên các nút máy tính và phân phối chúng cho các ứng dụng. ResourceManager đảm bảo rằng tài nguyên được sử dụng hiệu quả, đáp ứng các yêu cầu về xử lý dữ liệu từ các ứng dụng khác nhau.
- **NodeManager:** NodeManager là thành phần chạy trên mỗi nút máy tính trong cụm. Nó chịu trách nhiệm quản lý tài nguyên cục bộ trên nút đó, bao gồm CPU, bộ nhớ và dung lượng lưu trữ. NodeManager thực hiện các công việc kiểm tra sức khỏe của tài nguyên và báo cáo cho ResourceManager về tình trạng tài nguyên địa phương.

YARN giúp cải thiện hiệu suất và khả năng mở rộng của Hadoop bằng cách tách quản lý tài nguyên và quá trình xử lý dữ liệu. Điều này cho phép nhiều ứng dụng chia sẻ tài nguyên cùng một lúc mà không cần phải cạnh tranh với nhau. Ví dụ, một cụm Hadoop có thể chạy cùng lúc các ứng dụng MapReduce, Spark, Hive, và HBase, mà mỗi ứng dụng sẽ được quản lý và chia sẻ tài nguyên một cách độc lập thông qua YARN.

Ngoài ra, YARN cũng hỗ trợ khả năng xử lý dữ liệu thời gian thực bằng cách cho phép chạy các ứng dụng dựa trên dữ liệu thời gian thực như Apache Storm và Apache Flink trên cùng một cụm Hadoop.

Hệ sinh thái Hadoop: Hadoop có một hệ sinh thái phong phú với nhiều công cụ và khung làm việc liên quan. Các công cụ như Hive cho truy vấn dữ liệu, Pig cho xử lý dữ liệu, HBase cho lưu trữ dữ liệu phân tán, và Spark cho xử lý dữ liệu thời gian thực, tất cả đều tích hợp dễ dàng với Hadoop. Điều này giúp doanh nghiệp tận dụng các công cụ phù hợp với nhu cầu cụ thể của họ.

Độ tin cậy và khả năng xử lý lỗi: Hadoop được thiết kế để xử lý lỗi. Nó có khả năng phát hiện và khắc phục lỗi trên các nút máy tính. Điều này đảm bảo rằng dữ liệu luôn đáng tin cậy và không bị mất, điều quan trọng trong việc lưu trữ và xử lý dữ liệu quý báu trong môi trường Big Data.

Tóm lại, Hadoop là một công cụ quan trọng trong lĩnh vực Big Data với khả năng lưu trữ và xử lý dữ liệu lớn, tích hợp với nhiều công cụ, khả năng mở rộng linh hoạt và khả năng xử lý lỗi. Điều này giúp doanh nghiệp quản lý và tận dụng hiệu quả dữ liệu lớn trong môi trường ngày càng phức tạp của họ.

3.2.2 *Spark:*

Với chủ đề mà nhóm đã và đang phân tích, chủ đề có tích hợp với Apache Spark, cho nên ở đây ta sẽ nói chi tiết về Spark.

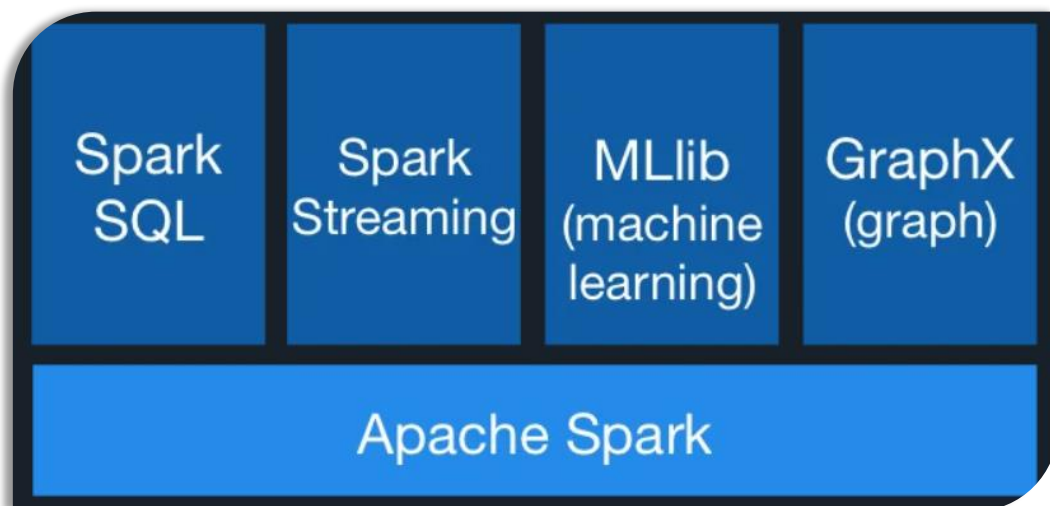
Apache Spark là một framework mã nguồn mở mạnh mẽ cho xử lý và phân tích dữ liệu lớn. Nó hỗ trợ xử lý dữ liệu thời gian thực và tính toán song song.

3.2.2.1 Lịch sử phát triển

[3] Apache Spark bắt đầu như một dự án nghiên cứu tại UC Berkeley AMPLab vào năm 2009 và có nguồn mở vào đầu năm 2010. Nhiều ý tưởng đằng sau hệ thống đã được trình bày trong nhiều tài liệu nguyên cứu khác nhau trong nhiều năm.

Sau khi được phát hành, Spark đã phát triển thành một cộng đồng nhà phát triển rộng lớn và chuyển sang Quỹ phần mềm Apache vào năm 2013. Ngày nay, dự án được cộng tác phát triển bởi một cộng đồng gồm hàng trăm nhà phát triển từ hàng trăm tổ chức.

3.2.2.2 Các thành phần [4]



[4] Apache Spark gồm các thành phần chính : Spark Core, Spark Streaming, Spark SQL, MLlib và GraphX, trong đó:

Spark Core: là nền tảng cho các thành phần còn lại và các thành phần này muốn khởi chạy được thì đều phải thông qua Spark Core do Spark Core đảm nhận vai trò thực hiện công việc tính toán và xử lý trong bộ nhớ (In-memory computing) đồng thời nó cũng tham chiếu các dữ liệu được lưu trữ tại các hệ thống lưu trữ bên ngoài.

Spark SQL: cung cấp một kiểu data abstraction mới (SchemaRDD) nhằm hỗ trợ cho cả kiểu dữ liệu có cấu trúc (structured data) và dữ liệu nửa cấu trúc (semi-structured data – thường là dữ liệu dữ liệu có cấu trúc nhưng không đồng nhất và cấu trúc của dữ liệu phụ thuộc vào chính nội dung của dữ liệu ấy). Spark SQL hỗ trợ DSL (Domain-specific language) để thực hiện các thao tác trên DataFrames bằng ngôn ngữ Scala, Java hoặc Python và nó cũng hỗ trợ cả ngôn ngữ SQL với giao diện command-line và ODBC/JDBC server.

Spark Streaming: được sử dụng để thực hiện việc phân tích stream bằng việc coi stream là các mini-batches và thực hiện kỹ thuật RDD transformation đối với các dữ liệu mini-batches này. Qua đó cho phép các đoạn code được viết cho xử lý batch có thể được tận dụng lại vào trong việc xử lý stream, làm cho việc phát triển lambda architecture được dễ dàng hơn. Tuy nhiên điều này lại tạo ra độ trễ trong xử lý dữ liệu (độ trễ chính bằng mini-batch duration) và do đó nhiều chuyên gia cho rằng Spark Streaming không thực sự là công cụ xử lý streaming giống như Storm hoặc Flink.

MLlib (Machine Learning Library): MLlib là một nền tảng học máy phân tán bên trên Spark do kiến trúc phân tán dựa trên bộ nhớ. Theo các so sánh benchmark Spark MLlib nhanh hơn 9 lần so với phiên bản chạy trên Hadoop (Apache Mahout).

GrapX: Grapx là nền tảng xử lý đồ thị dựa trên Spark. Nó cung cấp các Api để diễn tả các tính toán trong đồ thị bằng cách sử dụng Pregel Api.

3.2.2.3 Các đặc điểm

- Xử lý dữ liệu: Spark xử lý dữ liệu theo lô và thời gian thực
- Tính tương thích: Có thể tích hợp với tất cả các nguồn dữ liệu và định dạng tệp được hỗ trợ bởi cụm Hadoop.
- Hỗ trợ ngôn ngữ: hỗ trợ Java, Scala, Python và R.
- Phân tích thời gian thực: Apache Spark có thể xử lý dữ liệu thời gian thực tức là dữ liệu đến từ các luồng sự kiện thời gian thực với tốc độ hàng triệu sự kiện mỗi giây. Apache Spark có thể được sử dụng để xử lý phát hiện gian lận trong khi thực hiện các giao dịch ngân hàng. Đó là bởi vì, tất cả các khoản thanh toán trực tuyến được thực hiện trong thời gian thực và chúng ta cần ngừng giao dịch gian lận trong khi quá trình thanh toán đang diễn ra.

❖ Ví dụ: Data Twitter chẳng hạn hoặc lượt chia sẻ, đăng bài trên Facebook. Sức mạnh Spark là khả năng xử lý luồng trực tiếp hiệu quả.

Apache Spark là Framework thực thi dữ liệu dựa trên Hadoop HDFS. Apache Spark không thay thế cho Hadoop nhưng nó là một framework ứng dụng. Apache Spark tuy ra đời sau nhưng được nhiều người biết đến hơn Apache Hadoop vì khả năng xử lý hàng loạt và thời gian thực.

3.2.3 NoSQL Databases

Các hệ thống cơ sở dữ liệu NoSQL như MongoDB, Cassandra và HBase được sử dụng để lưu trữ và truy xuất dữ liệu phi cấu trúc và phân tán.

Hệ thống cơ sở dữ liệu NoSQL (không SQL) là một loại cơ sở dữ liệu không tuân theo mô hình cơ sở dữ liệu quan hệ truyền thống, chúng thường được sử dụng trong các ứng dụng big data và các tình huống đòi hỏi lưu trữ và truy xuất dữ liệu phi cấu trúc hoặc dữ liệu phân tán. Dưới đây là một số thông

tin chi tiết hơn về một số hệ thống cơ sở dữ liệu NoSQL quan trọng trong ngữ cảnh big data:

- **MongoDB:** MongoDB là một hệ thống cơ sở dữ liệu NoSQL thuộc loại cơ sở dữ liệu tài liệu (document database). Dữ liệu trong MongoDB được lưu trữ dưới dạng các tài liệu (documents) được biểu diễn bằng JSON hoặc BSON (binary JSON). MongoDB hỗ trợ dữ liệu động và có khả năng mở rộng dự án big data bằng cách chia tài liệu thành các bản sao (replica sets) và cụm (clusters).
- **Cassandra:** Cassandra là một hệ thống cơ sở dữ liệu NoSQL thuộc loại cơ sở dữ liệu cột (column-family database). Cassandra được thiết kế để lưu trữ dữ liệu phân tán trên nhiều máy chủ và hỗ trợ quy mô lớn. Nó có khả năng chịu lỗi và đảm bảo sẵn sàng cao cho các ứng dụng big data.
- **HBase:** HBase là hệ thống cơ sở dữ liệu NoSQL thuộc loại cơ sở dữ liệu cột (column-family database). Nó được xây dựng trên cơ sở Hadoop và được sử dụng để lưu trữ dữ liệu phân tán và truy xuất dữ liệu nhanh chóng. HBase thường được sử dụng trong các tình huống yêu cầu quy mô lớn và hiệu suất cao cho big data.

Các hệ thống NoSQL này thường được ưa chuộng trong big data vì khả năng mở rộng, hiệu suất cao, và khả năng làm việc với dữ liệu phi cấu trúc hoặc dữ liệu lớn. Chúng cho phép xử lý và lưu trữ dữ liệu theo cách hiệu quả hơn trong môi trường big data, nơi mà các cơ sở dữ liệu quan hệ truyền thống có thể gặp khó khăn trong việc đáp ứng yêu cầu.

3.2.4 SQL Databases:

Cơ sở dữ liệu quan hệ truyền thống như MySQL, PostgreSQL và Oracle vẫn được sử dụng trong môi trường Big Data.

3.2.5 Apache Kafka:

Kafka là một hệ thống xử lý luồng dữ liệu thời gian thực, thường được sử dụng cho việc thu thập và truyền dữ liệu trong các hệ thống Big Data.

Kiến thức cơ bản về Kafka: Kafka là một nền tảng xử lý luồng dữ liệu được thiết kế để xử lý hàng tỷ sự kiện mỗi ngày, và nó có khả năng chịu tải và mở rộng rất tốt.

Apache Kafka chia dữ liệu thành các topic (chủ đề), và dữ liệu được xuất bằng cách sử dụng producer (sản phẩm) và tiêu thụ bằng consumer (người tiêu thụ). Kafka giúp các ứng dụng big data có thể gửi và nhận dữ liệu theo thời gian thực, đồng bộ hóa các sự kiện, và phân tán dữ liệu trên nhiều máy chủ.

Trong ngữ cảnh big data: Kafka là một phần quan trọng trong kiến trúc dữ liệu thời gian thực trong big data, cho phép ứng dụng big data thu thập dữ liệu từ nhiều nguồn khác nhau, như logs, cơ sở dữ liệu, cảm biến, và nhiều nguồn dữ liệu khác. Nó giúp giải quyết vấn đề lưu trữ và truyền dữ liệu trực tuyến một cách hiệu quả, đảm bảo tính toàn vẹn và khả năng mở rộng của dữ liệu trong big data pipeline.

Ứng dụng của Kafka trong big data: Kafka thường được sử dụng trong kiến trúc big data để kết nối các phần của hệ thống, chẳng hạn như Apache Spark, Apache Hadoop, cơ sở dữ liệu NoSQL, và các ứng dụng xử lý luồng dữ liệu thời gian thực khác. Nó giúp vận chuyển dữ liệu từ nguồn gốc tới các ứng dụng xử lý và lưu trữ, giúp cải thiện khả năng phân tích, báo cáo, và phản hồi trong thời gian thực.

Kafka là một công cụ quan trọng để xây dựng kiến trúc big data dựa trên luồng dữ liệu thời gian thực, giúp ứng dụng big data hoạt động hiệu quả và đảm bảo tính nhất quán và toàn vẹn của dữ liệu.

3.3 Ứng dụng của Big Data [5]

3.3.1 Ngân hàng



Dữ liệu lớn đã chuyển đổi cách ngân hàng quản lý tài chính và thu tiền mặt một cách toàn diện, mang lại sự hiệu quả và minh bạch hơn trong mọi khía cạnh của ngành này. Sự ứng dụng của công nghệ đã đánh bại những thách thức gặp phải trước đây, giúp các ngân hàng tối ưu hóa hoạt động của họ và tạo ra nhiều nguồn doanh thu hơn từ các

khía cạnh khác nhau.

Một trong những lợi ích quan trọng của việc sử dụng dữ liệu lớn là khả năng phát hiện và ngăn chặn gian lận. Bằng cách phân tích mẫu dữ liệu, các ngân hàng có thể xác định những hoạt động không bình thường và cảnh báo về các giao dịch đáng ngờ. Điều này giúp bảo vệ tài sản của người dùng và tạo ra môi trường an toàn hơn.

Ngoài ra, dữ liệu lớn cũng cải thiện việc thực hiện giao dịch. Các ngân hàng có thể sử dụng dữ liệu để tối ưu hóa quá trình xử lý giao dịch, làm cho chúng trở nên nhanh chóng và hiệu quả hơn. Điều này không chỉ giúp tiết kiệm thời gian cho người dùng mà còn giúp giảm chi phí hoạt động của ngân hàng.

Một ví dụ tiêu biểu về việc sử dụng dữ liệu lớn trong ngành ngân hàng là Western Union. Họ đã áp dụng cách tiếp cận đa kênh để cá nhân hóa trải nghiệm của người dùng, xử lý hàng chục giao dịch mỗi giây và tổng hợp dữ liệu trên một nền tảng

chung. Điều này giúp họ xây dựng các mô hình thống kê và dự đoán, từ đó cung cấp dự đoán chính xác hơn về các giao dịch tài chính và xu hướng thị trường.

Tổng cộng, dữ liệu lớn đã mở ra một loạt ứng dụng mới cho ngân hàng, nâng cao hiểu biết của người dùng, cải thiện quá trình thực hiện giao dịch và tạo ra trải nghiệm người dùng đáng nhớ hơn bao giờ hết.

3.3.2 Giáo dục



Khi nói về ngành Giáo dục, dữ liệu thu thập được từ các khóa học, sinh viên, giảng viên và kết quả là rất lớn, việc giải thích dữ liệu này có thể mang lại những hiểu biết sâu sắc hữu ích để cải thiện hoạt động và hoạt động của các cơ sở giáo dục. Từ việc thúc đẩy học tập hiệu quả, cải thiện tuyển dụng quốc tế cho các trường đại học, hỗ trợ sinh

viên xác lập mục tiêu nghề nghiệp, giảm tỷ lệ bỏ học đại học, thúc đẩy đánh giá sinh viên rõ ràng, tăng cường quá trình ra quyết định và cải thiện kết quả của sinh viên, Dữ liệu lớn có vai trò không thể thiếu trong lĩnh vực này.

Một ví dụ nổi bật là trường Đại học Florida. Trường đại học sử dụng IBM InfoSphere để thu thập, tải và vận chuyển dữ liệu qua nhiều tài nguyên khác nhau. Trường đại học sử dụng IBM SPSS Modeler trong trường hợp phân tích dự đoán và lập mô hình dữ liệu, cũng như IBM Cognos Analytics để kiểm tra và dự báo kết quả học tập của sinh viên.

Một số biến số khác nhau từ điểm số, nhân khẩu học và nền tảng kinh tế của học sinh giúp đo lường khả năng đánh giá khả năng bỏ học của học

sinh. Điều này giúp nhà trường xây dựng chính sách và thúc đẩy sự can thiệp thường xuyên đối với sinh viên có nguy cơ bỏ học.

3.3.3 Phương tiện truyền thông

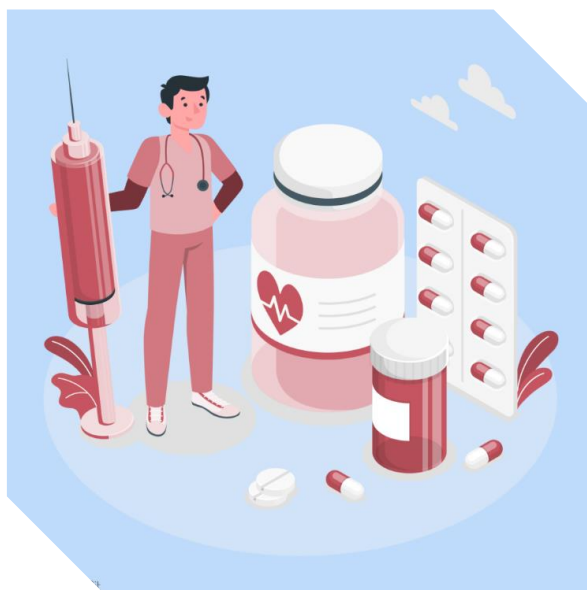


Tin đồn về các phương pháp sử dụng phương tiện thông thường đang dần biến mất vì các chiến lược tiêu thụ nội dung trực tuyến với sự trợ giúp của các thiết bị tiện ích hiện nay đã trở thành xu hướng mới nhất. Kể từ khi một lượng lớn dữ liệu được tạo ra, dữ liệu lớn đã xâm nhập thành công vào ngành này. Từ hỗ trợ đến dự đoán nhu cầu của khán giả, về thể loại, âm

nhạc và nội dung theo nhóm tuổi của họ, cho đến đề xuất cho họ những hiểu biết sâu sắc về tỷ lệ rời bỏ khách hàng, Dữ liệu lớn đã giúp cuộc sống của các công ty truyền thông trở nên dễ dàng hơn nhiều.

Một ví dụ thích hợp khác về cách dữ liệu lớn đóng vai trò then chốt trong việc chuyển đổi nền tảng truyền thông là Netflix. Công nghệ này không chỉ tác động đến loạt phim được Netflix đầu tư mà còn ảnh hưởng đến cách giới thiệu loạt phim này tới người dùng của họ. Lịch sử tìm kiếm và lịch sử xem của người dùng, bao gồm cả những vị trí mà người dùng đã tạm dừng video, tác động đến mọi thứ từ hình thu nhỏ được cá nhân hóa cho đến các chương trình chúng tôi xem trên phần “Phổ biến trên Netflix”.

3.3.4 Chăm sóc sức khỏe



Dữ liệu lớn đã có tầm ảnh hưởng to lớn trong việc cải thiện ngành chăm sóc sức khỏe hiện đại. Nó đã định hình lại hoàn toàn cách chúng ta tiếp cận và cung cấp dịch vụ y tế, từ việc giảm chi phí điều trị đến dự đoán và ngăn chặn dịch bệnh bùng phát, cũng như tối ưu hóa phòng ngừa bệnh tật.

Một trong những khía cạnh quan trọng nhất của việc sử dụng dữ liệu lớn trong lĩnh vực chăm sóc sức khỏe là cung cấp dự đoán chính xác về sự phát triển của tình trạng sức khỏe của mỗi bệnh nhân. Dữ liệu thu thập hàng ngày có thể giúp xác định các biểu hiện sớm của bệnh, từ đó cho phép can thiệp kịp thời và cung cấp liệu pháp phù hợp.

Hồ sơ sức khỏe điện tử (EHRs) đã thay đổi cách chúng ta quản lý thông tin sức khỏe. Chúng giúp tạo ra một kho dữ liệu tổng hợp về bệnh nhân, cho phép các nhà y tế truy cập thông tin quan trọng một cách nhanh chóng và hiệu quả. Điều này giúp cải thiện quá trình chăm sóc bệnh nhân, đồng thời giảm thiểu sai sót và sơ sót trong việc ghi chép thông tin.

Mayo Clinic là một minh chứng rõ ràng về sự thành công của việc áp dụng dữ liệu lớn trong chăm sóc sức khỏe. Nền tảng này đã sử dụng phân tích dữ liệu lớn để nâng cao chất lượng cuộc sống của bệnh nhân, đặc biệt là trong việc theo dõi tình trạng sức khỏe và cung cấp can thiệp y tế theo thời gian thực. Điều này không chỉ giúp giảm bớt gánh nặng cho người bệnh mà còn tạo điều kiện cho một chăm sóc toàn diện và hiệu quả hơn.

Tóm lại, dữ liệu lớn đã thay đổi ngành chăm sóc sức khỏe, giúp dự đoán, phòng ngừa và điều trị bệnh tốt hơn, cải thiện chất lượng cuộc sống và giúp tối ưu hóa việc quản lý thông tin sức khỏe cá nhân. Mayo Clinic và nhiều tổ chức y tế khác là ví dụ điển hình cho việc tận dụng dữ liệu lớn để mang lại lợi ích thực sự trong lĩnh vực này.

3.3.5 Nông nghiệp



Phân tích dữ liệu lớn đã trở thành một trụ cột quan trọng trong ngành nông nghiệp, thúc đẩy sự phát triển của nông nghiệp thông minh và chính xác. Các ứng dụng của dữ liệu lớn trong lĩnh vực này đã mang lại nhiều lợi ích quan trọng, bắt đầu từ việc đáp ứng nhu cầu lương thực của thế giới đang ngày càng gia tăng.

Một trong những lĩnh vực chính mà dữ liệu lớn đã tạo ra sự thay đổi đáng kể là trong việc cung cấp thông tin liên quan đến thời tiết và khí hậu. Nông dân ngày nay có thể dựa vào dữ liệu lớn để theo dõi và dự đoán các yếu tố như lượng mưa, nhiệt độ, và điều kiện khí hậu. Điều này giúp họ tối ưu hóa việc quản lý cây trồng và vụ mùa, đảm bảo rằng họ có thể ứng phó với biến đổi khí hậu và tối ưu hóa năng suất của mình.

Sử dụng thuốc trừ sâu thông minh và chính xác cũng là một khía cạnh quan trọng trong nông nghiệp hiện đại. Dữ liệu lớn cho phép theo dõi mức độ nhiễm sâu, loại cỏ dại và các vấn đề khác một cách chi tiết. Điều này giúp nông dân chỉ sử dụng thuốc trừ sâu khi cần thiết, giảm lãng phí và tối ưu hóa hiệu suất nông nghiệp.

Ngoài ra, quản lý thiết bị thông qua Internet of Things (IoT) là một ứng dụng quan trọng khác của dữ liệu lớn trong nông nghiệp. Việc theo dõi và điều khiển các thiết bị nông nghiệp có thể giúp tối ưu hóa hoạt động nông nghiệp và giảm chi phí hoạt động.

Một ví dụ tiêu biểu về cách dữ liệu lớn đang được tận dụng trong ngành nông nghiệp là trường hợp của Bayer Digital Farming. Họ đã phát triển ứng dụng sử dụng máy học và trí tuệ nhân tạo để xác định loại cỏ dại. Nông dân có thể chia sẻ hình ảnh cỏ dại qua ứng dụng, và dữ liệu được đối chiếu với cơ sở dữ liệu lớn của Bayer để xác định loại cỏ dại đó. Ứng dụng này cho phép can thiệp kịp thời, bảo vệ cây trồng khỏi cỏ dại và cải thiện năng suất.

Tổng cộng, dữ liệu lớn đang biến đổi cách nông nghiệp hoạt động, tạo ra mô hình nông nghiệp thông minh, tiết kiệm chi phí và tạo ra các cơ hội kinh doanh mới trong ngành này.

3.3.6 Du lịch



Dữ liệu lớn đã có tác động cực kỳ quan trọng đối với ngành giao thông vận tải, góp phần cải thiện mô hình hoạt động của nó, từ việc quản lý doanh thu và duy trì danh tiếng cho đến việc thực hiện chiến lược tiếp thị. Việc tận dụng dữ liệu lớn trong ngành này đã tạo ra những thay đổi đáng kể và nâng cao hiệu suất tổng thể của hệ thống

giao thông.

Dữ liệu lớn đã cho phép xác định một loạt thông tin quan trọng liên quan đến giao thông, từ dự báo nhu cầu cho các dịch vụ vận tải đến quản lý thông tin

về thời gian chờ đợi và xác định các điểm nguy hiểm tiềm năng để nâng cao an toàn giao thông. Cụ thể, nó đã giúp tạo ra các ứng dụng và dịch vụ thông minh như hệ thống điều hành taxi thông qua ứng dụng di động.

Một ví dụ tiêu biểu về cách dữ liệu lớn đã thay đổi ngành giao thông vận tải là Uber. Nền tảng này đã tạo ra và sử dụng một lượng lớn dữ liệu về phương tiện, taxi, địa điểm và thông tin về mỗi chuyến đi. Dữ liệu này không chỉ đơn thuần được sử dụng để theo dõi chuyến đi và tính phí mà còn để dự đoán nhu cầu của khách hàng, tối ưu hóa nguồn cung cấp và vị trí của taxi, cũng như điều chỉnh giá vé chuyến đi dựa trên nhiều yếu tố.

Uber không chỉ giúp kết nối người dùng với dịch vụ vận tải một cách thuận tiện, mà còn cung cấp thông tin liên quan đến lưu lượng giao thông và xu hướng di chuyển, hỗ trợ quản lý thông tin lưu lượng và cải thiện việc điều tiết giao thông.

Tổng cộng, dữ liệu lớn đã chuyển đổi ngành giao thông vận tải, làm cho nó trở nên hiệu quả hơn, an toàn hơn và cung cấp cho người dùng những trải nghiệm thuận tiện và đáng nhớ hơn bao giờ hết. Uber là một ví dụ xuất sắc về cách sử dụng dữ liệu lớn để tạo ra một mô hình giao thông thông minh và hiệu quả.

3.3.7 Sản xuất



Dữ liệu lớn đã chuyển đổi cách sản xuất được thực hiện, biến nó từ một quy trình thủ công và gian khổ thành một quy trình hiện đại và tối ưu hóa. Sự kết hợp giữa công nghệ và dữ liệu đã mang lại sự cách mạng toàn diện trong lĩnh vực sản xuất. Dữ liệu lớn đóng vai trò quan trọng trong việc tối ưu hóa quy trình sản xuất, cá nhân hóa thiết kế sản phẩm, đảm bảo chất

lượng, quản lý chuỗi cung ứng và đánh giá các rủi ro tiềm ẩn.

Một trong những ví dụ thú vị về tầm quan trọng của dữ liệu lớn trong lĩnh vực sản xuất là Rolls Royce. Công ty này đã sử dụng phân tích dữ liệu lớn để cải thiện quy trình thiết kế, giảm thời gian phát triển sản phẩm và tăng hiệu suất cũng như chất lượng của sản phẩm. Rolls Royce cũng đã giảm thiểu chi phí sản xuất bằng cách sử dụng dữ liệu lớn để phát hiện và khắc phục các lỗi trong quá trình thiết kế.

Dữ liệu lớn cho phép các công ty sản xuất theo dõi hiệu suất của các thiết bị và máy móc, dự đoán khi cần bảo dưỡng hoặc thay thế, giúp tối ưu hóa sử dụng tài nguyên và giảm thời gian ngừng sản xuất. Nó cũng hỗ trợ trong việc đảm bảo chất lượng sản phẩm thông qua việc theo dõi từng quy trình sản xuất và phát hiện nguy cơ sự cố sớm để khắc phục chúng.

Tổng cộng, dữ liệu lớn đã biến đổi cách sản xuất được thực hiện, tạo ra những ưu điểm rõ rệt trong việc tối ưu hóa quy trình, nâng cao chất lượng và giảm chi phí sản xuất. Rolls Royce là một ví dụ xuất sắc về cách dữ liệu lớn đã đóng vai trò quan trọng trong việc định hình tương lai của ngành sản xuất.

3.3.8 Chính phủ

Dữ liệu lớn đã trở thành một tài nguyên quý báu cho các chính phủ trên khắp thế giới, đặc biệt là trong việc quản lý thông tin đa dạng và phức tạp về công dân, tài nguyên, và các khía cạnh khác của xã hội. Các chính phủ nhận được một lượng dữ liệu khổng lồ hàng ngày từ nhiều nguồn khác nhau, và việc sử dụng hiệu quả dữ liệu này đã trở thành một yếu tố quyết định trong quá trình ra quyết định và kế hoạch phát triển.

Trong lĩnh vực phát triển, chính phủ sử dụng dữ liệu lớn để hiểu hơn về tốc độ tăng trưởng dân số, khảo sát địa lý, và tài nguyên năng lượng. Điều này giúp họ xây dựng các kế hoạch phát triển chi tiết, đảm bảo sử dụng tài nguyên hiệu quả và đáp ứng nhu cầu của cộng đồng.

Cũng không thể bỏ qua vai trò của dữ liệu lớn trong bảo đảm an ninh quốc gia. Bộ An ninh Nội địa (DHS) là một ví dụ đáng chú ý. DHS sử dụng hệ thống nhận dạng xâm nhập để theo dõi lưu lượng truy cập internet, cả trong và ngoài hệ thống Liên bang, để phát hiện các hoạt động đe dọa an ninh mạng. Họ tập trung vào việc phát hiện các nỗ lực của phần mềm độc hại và truy cập không được kiểm soát, đặc biệt là các cuộc tấn công và thâm nhập từ các thực thể thù địch.

Nhờ vào việc sử dụng dữ liệu lớn và công nghệ tiên tiến, chính phủ có khả năng đánh giá và phản ứng nhanh chóng trước các mối đe dọa, bảo vệ quốc gia và công dân khỏi các mối đe dọa an ninh mạng ngày càng tinh vi. Điều này đồng nghĩa với việc duy trì an ninh và bình yên cho quốc gia và thúc đẩy phát triển bền vững trong tương lai.

Ngoài ra còn có thêm **Thống kê sự quan tâm của người dùng Facebook đối với một số sản phẩm**. Đây chính chủ đề báo cáo của nhóm. Với thời đại công nghệ hiện nay mạng xã hội đang phát triển nhanh chóng cùng với đó là sự mở rộng các mô hình kinh doanh thông qua mạng qua xã hội như livestream, đăng post quảng cáo, giới thiệu các sản phẩm thông qua video hay các bài đăng trên

mạng xã hội, cụ thể là Facebook nơi đang có lượng người dùng đông đảo, phù hợp là nơi để các doanh nghiệp, người buôn bán, giới thiệu và buôn bán hàng hóa của bản thân với mọi người dùng.

Ở đây nhóm chúng sử dụng công cụ Spark để thực hiện việc thông kê bởi những tác dụng hữu ích như :

Phân tích Sự Quan Tâm: Bằng cách thu thập và phân tích dữ liệu về sự quan tâm của người dùng đối với các sản phẩm cụ thể trên Facebook, bạn có thể hiểu rõ hơn về đối tượng mục tiêu và sở thích của họ. Điều này có thể giúp bạn tối ưu hóa chiến dịch tiếp thị, cung cấp nội dung tùy chỉnh và cải thiện trải nghiệm người dùng.

Đề Xuất Sản Phẩm: Dựa trên dữ liệu thu thập được, bạn có thể sử dụng các thuật toán đề xuất để gợi ý sản phẩm tương tự hoặc liên quan mà người dùng có thể quan tâm. Điều này có thể tăng cơ hội bán hàng và tăng doanh số bán hàng.

Tùy chỉnh Nội Dung: Bằng việc hiểu sâu hơn về sở thích của người dùng, bạn có thể tạo nội dung tùy chỉnh và quảng cáo mà phù hợp với họ, từ đó tạo ra tương tác tốt hơn và cải thiện tỷ lệ chuyển đổi.

Xây dựng Mối Quan Hệ: Sử dụng dữ liệu về sự quan tâm của người dùng, bạn có thể xây dựng mối quan hệ sâu hơn với họ thông qua việc cung cấp giá trị thực sự dựa trên nhu cầu và mong muốn của họ.

2. Bài toán

Trong thời đại số hóa hiện nay, việc thu thập và phân tích dữ liệu về sự quan tâm của người dùng Facebook đối với các sản phẩm trở thành một phần không thể thiếu đối với mọi doanh nghiệp và người buôn bán. Dữ liệu này, nếu không được phân loại và xử lý một cách hiệu quả, có thể gây ra nhiều khó khăn và thách thức đối với cả người tìm kiếm thông tin và người cung cấp sản phẩm hoặc dịch vụ.

Một điểm quan trọng là dữ liệu về sự quan tâm của người dùng trên Facebook có thể rất phong phú và đa dạng. Nếu không được phân loại hoặc không sử dụng công cụ và kỹ thuật phù hợp để xử lý, dữ liệu này có thể trở nên mơ hồ và khó hiểu. Điều này làm cho việc tìm kiếm thông tin trở nên khó khăn, đặc biệt là khi người dùng Facebook có sự quan tâm đa dạng đối với nhiều sản phẩm hoặc lĩnh vực khác nhau.

Từ góc độ của những người đang cố gắng xây dựng chiến lược quảng cáo và kinh doanh, việc không thể nắm bắt được sự thịnh hành của các sản phẩm có thể dẫn đến các hậu quả đáng lo ngại. Nếu họ không thể hiểu rõ được xu hướng và sở thích của đối tượng mục tiêu, họ có thể tiêu tốn nhiều nguồn lực vào quảng cáo và tiếp thị không hiệu quả. Điều này đồng nghĩa với việc lãng phí ngân sách quảng cáo, thất bại trong việc thu hút khách hàng, và mất cơ hội cạnh tranh.

Tuy nhiên, việc sử dụng công cụ phân tích dữ liệu hiện đại, như Apache Spark, có thể giúp giải quyết các thách thức này. Bằng cách thu thập và xử lý dữ liệu một cách hiệu quả, chúng ta có thể tạo ra các hệ thống thông tin cụ thể, dễ đọc và dễ hiểu về sự quan tâm của người dùng đối với các sản phẩm. Nó giúp tạo ra cái nhìn sâu sắc về thị trường, xu hướng, và nhu cầu của khách hàng.

Điều này giúp người kinh doanh và người quảng cáo tạo ra các chiến lược tiếp thị tùy chỉnh, dựa trên sở thích và nhu cầu cụ thể của đối tượng mục tiêu. Việc sử dụng dữ liệu phân tích giúp họ hiểu rõ hơn và tạo ra những chiến dịch quảng cáo chính xác, hướng đến những người có khả năng mua sắm và quan tâm thực sự. Điều này giúp tối ưu hóa ngân sách quảng cáo, tăng khả năng thành công của chiến dịch, và cải thiện tỷ lệ chuyển đổi.

Trong bối cảnh thị trường cạnh tranh và phát triển công nghệ ngày càng nhanh, việc sử dụng dữ liệu lớn để phân tích sự quan tâm của người dùng trên Facebook trở nên quan trọng hơn bao giờ hết. Sự hiểu biết sâu rộng về đối tượng mục tiêu và khả năng tùy chỉnh chiến lược là chìa khóa để thành công trong kinh doanh trực tuyến. Do đó, việc đảm bảo rằng dữ liệu này được thu thập, xử lý và phân tích một cách chính xác và hiệu quả là vô cùng quan trọng để đáp ứng nhu cầu của thị trường và khách hàng một cách tối ưu.

3. Giải pháp

Apache Spark là công nghệ mạnh mẽ và đa nhiệm, chúng có thể giúp bạn xây dựng hệ thống phân tích dữ liệu sâu và mô hình dữ liệu đồ thị để nắm bắt sự quan tâm của người dùng Facebook đối với sản phẩm một cách hiệu quả. Dưới đây là cách bạn có thể mở rộng ứng dụng của bạn:

Thu thập và xử lý dữ liệu bằng Apache Spark: Apache Spark là một framework xử lý dữ liệu lớn, cho phép bạn thu thập và xử lý dữ liệu từ Facebook một cách hiệu quả. Sử dụng Spark để kết nối với API của Facebook hoặc các nguồn dữ liệu khác để thu thập thông tin về sự quan tâm của người dùng đối với sản phẩm cụ thể, bao gồm lượt xem, lượt thích, bình luận, và nhiều dạng tương tác khác. Spark có khả năng xử lý dữ liệu theo thời gian thực, cho phép bạn theo dõi sự quan tâm của người dùng ngay lập tức và tổ chức dữ liệu một cách hiệu quả.

Tùy chỉnh nội dung và xây dựng mối quan hệ: Sử dụng dữ liệu về sự quan tâm của người dùng và sản phẩm, bạn có thể tạo nội dung tùy chỉnh và quảng cáo phù hợp với từng người dùng. Xây dựng mối quan hệ sâu hơn với người dùng bằng cách cung cấp giá trị thực sự dựa trên nhu cầu và mong muốn của họ, từ đó tạo ra tương tác tốt hơn và cải thiện tỷ lệ chuyển đổi.

Tóm lại, kết hợp Apache Spark cho phép bạn thu thập, lưu trữ, xử lý dữ liệu và xây dựng mối quan hệ giữa người dùng và sản phẩm để hiểu rõ hơn về sự quan tâm của họ và tạo ra các chiến dịch tiếp thị, đề xuất sản phẩm và nội dung tùy chỉnh hiệu quả hơn trên mạng xã hội như Facebook.

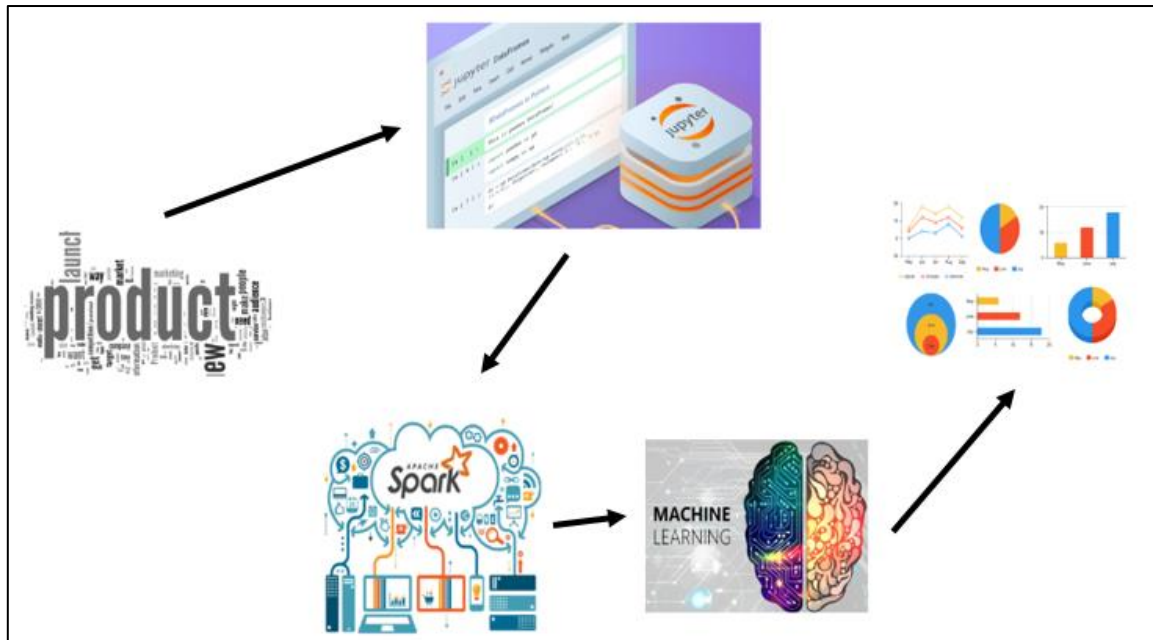
*a. Dùng phương pháp **Lmlib Regression (hồi quy tuyến tính)** vào bài toán*

- **Phương thức làm việc:** sử dụng hàm tuyến tính để xác định mối quan hệ giữa biến đầu vào và đầu ra. Hồi quy tuyến tính cố gắng tìm ra các

trọng số sao cho tổng sai số bình phương (bình phương tổng của sai số dự đoán và giá trị thực tế) là nhỏ nhất.

- **Loại mô hình:** đây là mô hình hồi quy sử dụng một hàm tuyến tính để dự đoán giá trị đầu ra dựa trên các biến đầu vào. Mô hình này giả định mối quan hệ tuyến tính giữa biến đầu vào và đầu ra.
- **Tính khả thi với dữ liệu phi tuyến tính:** hiệu quả khi mối quan hệ giữa biến đầu vào và đầu ra là tuyến tính. Khó khăn khi xử lý dữ liệu phi tuyến tính.
- **Xử lý đặc trưng và nhiễu:** đòi hỏi sự lựa chọn đặc trưng và xử lý nhiễu để đảm bảo tính ổn định của mô hình.
- **Phân loại và hồi quy:** thích hợp cho nhiệm vụ hồi quy (dự đoán giá trị số liên tục).

3.1 Mô hình giải pháp



Cách hoạt động của mô hình theo chủ đề của nhóm:

- Bước 1, đưa bộ dữ liệu vào Apache Spark

- Bước 2, làm sạch dữ liệu bằng Spark Dataframe, để làm sạch dữ liệu (Dùng Spark DataFrame)
- Bước 3, sau khi làm sạch dữ liệu, ta sử dụng Mllib để training (Dự đoán số liệu)
- Cuối cùng, Report Visualize dưới dạng đồ thị, thể hiện các mối liên quan của các chủ thể.

3.2 Cài đặt thử nghiệm

3.2.1 Datasets

Quá trình thực nghiệm học máy bao gồm các bước sau đây:

- Thu thập dữ liệu: thu thập thông tin dữ liệu của một số sản phẩm được người dùng Facebook quan tâm.
- Định dạng trường dữ liệu:

Tên trường	Mô tả
Date	Ngày, tháng, năm
Time	Giờ
Page Name	Tên của trang Facebook chứa sản phẩm mà người dùng quan tâm
Product Name	Tên của sản phẩm
Category	Loại sản phẩm
Origin	Nguồn gốc
No of Reaction	Người đã mua hoặc đã dùng
No of Comments	Người dùng facebook bình luận vào sản phẩm mà họ quan tâm
No of Share	Người dùng facebook chia sẻ rộng rãi sản phẩm mà họ quan tâm
Age Value	Tuổi của người dùng facebook

Age	Giá trị độ tuổi (1-15 tuổi, 16-30 tuổi)
Male_Value	Giới tính của người quan tâm đến sản phẩm là nam
Female_value	Giới tính của người quan tâm đến sản phẩm là nữ

- Tiền xử lý dữ liệu: để training học máy, điều quan trọng nhất là dữ liệu phải sạch, ở đây nhóm dùng PySpark để tiền xử lý dữ liệu.

3.2.2 Mô tả Datasets.

Thông tin của các trường có trong tệp ‘product.csv’:

- Date: tất cả dữ liệu đều được quét ở năm 2022.
- Page name: tên các page bán các sản phẩm được người dùng facebook quan tâm.
- Product name: tên cụ thể của các sản phẩm.
- Category: phân loại các sản phẩm, các các loại như sau:
 - Baby Essentials
 - Bags
 - Beauty & Skin Care
 - Body Paint
 - Clothing
 - Drinks & Food
 - Fashion Wear
 - Footwear
 - Grocery
 - Hair Care
 - Home Decoration
 - Jewellery
 - Mobile Phone
 - Modest Wear
 - Saree
- Reaction: số lượng người đánh giá, lên video hoặc bài viết bình phẩm về sản phẩm.
- Comments: số lượng bình luận của người dùng vào bài viết hay video hay link của sản phẩm được đăng lên facebook.
- Shared: số lượng người dùng share bài đăng sản phẩm mà họ quan tâm.

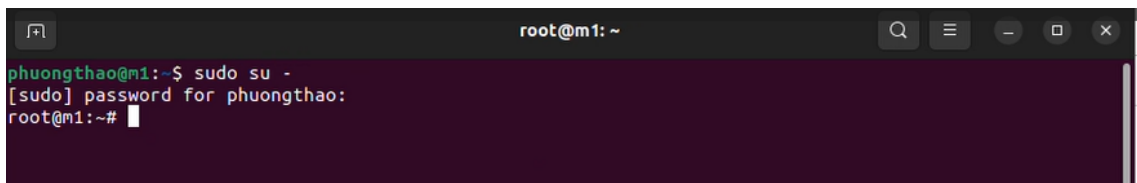
- Age_value: độ tuổi cụ thể từ 1 đến hơn 100 tuổi
- Age: giá trị độ tuổi
 - Từ 1 đến 10 tuổi.
 - Từ 11 đến 30 tuổi.
 - Từ 31 đến 40 tuổi.
 - Từ 41 đến 60 tuổi.
 - Và trên 60 tuổi.
- Male_value: số lượng người mang giới tính nam quan tâm đến một số sản phẩm trên facebook.
- Female_value: số lượng người mang giới tính nữ quan tâm đến một số sản phẩm trên facebook.

3.2.3 Cài đặt thử nghiệm

3.2.3.1 Cài đặt Apache Spark

Cài đặt Spark Lệnh vào chế độ user root. User chạy trên terminal là phuongthao@m1, sau khi vào chế độ user root sẽ là → root@m1

Sudo su –



```
root@m1: ~  
phuongthao@m1:~$ sudo su -  
[sudo] password for phuongthao:  
root@m1:~#
```

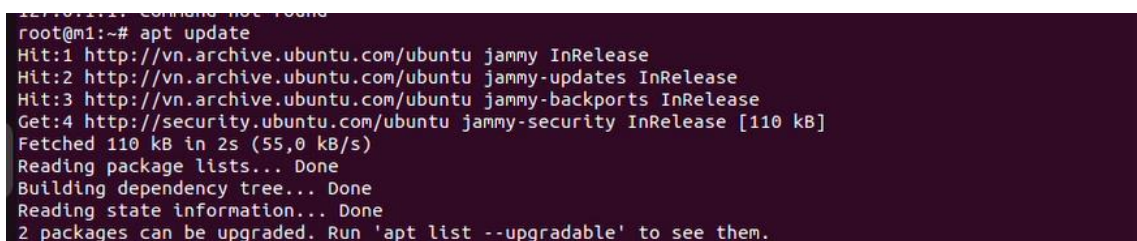
Lệnh kiểm tra lại sự trùng khớp của HostName và File Host, để khởi động được Spark thì bắt buộc các thông số phải giống nhau, nếu như các thông số khác nhau thì chắc chắn khi khởi động Apache Spark sẽ bị lỗi.

cat /etc/hosts



```
root@m1:~# wget https://dlcdn.apache.org/spark/spark-3.5.0/spark-3.5.0-bin-hadoop3.tgz  
--2023-09-17 19:44:13-- https://dlcdn.apache.org/spark/spark-3.5.0/spark-3.5.0-bin-hadoop3.tgz  
Resolving dlcdn.apache.org (dlcdn.apache.org)... 151.101.2.132, 2a04:4e42::644  
Connecting to dlcdn.apache.org (dlcdn.apache.org)|151.101.2.132|:443... connected.  
HTTP request sent, awaiting response... 200 OK  
Length: 400395283 (382M) [application/x-gzip]  
Saving to: 'spark-3.5.0-bin-hadoop3.tgz.1'  
  
spark-3.5.0-bin-hadoop3.t 100%[=====] 381,85M 8,29MB/s in 51s
```

Lệnh update các gói (package) có trong Ubuntu, để Ubuntu chạy ở phiên bản mới nhất: **apt update, apt -y upgrade**



```
root@m1:~# apt update  
Hit:1 http://vn.archive.ubuntu.com/ubuntu jammy InRelease  
Hit:2 http://vn.archive.ubuntu.com/ubuntu jammy-updates InRelease  
Hit:3 http://vn.archive.ubuntu.com/ubuntu jammy-backports InRelease  
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]  
Fetched 110 kB in 2s (55,0 kB/s)  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
2 packages can be upgraded. Run 'apt list --upgradable' to see them.
```



```

root@m1:~# apt -y upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
Get more security updates through Ubuntu Pro with 'esm-apps' enabled:
python3-jupyter-core libjs-jquery-ui python3-scipy jupyter-core
Learn more about Ubuntu Pro at https://ubuntu.com/pro
The following packages have been kept back:
gjs libgjs0g
0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.
root@m1:~# java -version
openjdk version "1.8.0_382"
OpenJDK Runtime Environment (build 1.8.0_382-8u382-ga-1~22.04.1-b05)
OpenJDK 64-Bit Server VM (build 25.382-b05, mixed mode)

```

Lệnh cài đặt Spark trong terminal: ta thêm cụm **wget** và dán đường link của Spark vào:

Wget <https://www.apache.org/dyn/closer.lua/spark/spark-3.5.0/spark-3.5.0-bin-hadoop3.tgz>

```

root@m1:~# wget https://d1cdn.apache.org/spark/spark-3.5.0/spark-3.5.0-bin-hadoop3.tgz
--2023-09-17 19:44:13-- https://d1cdn.apache.org/spark/spark-3.5.0/spark-3.5.0-bin-hadoop3.tgz
Resolving d1cdn.apache.org (d1cdn.apache.org)... 151.101.2.132, 2a04:4e42::644
Connecting to d1cdn.apache.org (d1cdn.apache.org)|151.101.2.132|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 400395283 (382M) [application/x-gzip]
Saving to: 'spark-3.5.0-bin-hadoop3.tgz.1'

spark-3.5.0-bin-hadoop3.t 100%[=====] 381,85M 8,29MB/s in 51s
2023-09-17 19:45:05 (7,47 MB/s) - 'spark-3.5.0-bin-hadoop3.tgz.1' saved [400395283/400395283]

```

Sau khi tải xong Spark về máy, ta tiếp tục giải nén file bằng lệnh sau đây (Lệnh phía dưới sẽ có sự thay đổi về version nên khi sử dụng cần chú ý đến version mà chúng ta đã tải trên máy. *Phiên bản máy đang được cài là Spark-3.5.0-bin-hadoop3*)

tar xvf spark-3.5.0-bin-hadoop3.tgz


```

root@m1:~# apt install python3-pip
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
python3-pip is already the newest version (22.0.2+dfsg-1ubuntu0.3).
0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.
root@m1:~# pip install pyspark
Requirement already satisfied: pyspark in /usr/local/lib/python3.10/dist-packages (3.4.1)
Requirement already satisfied: py4j==0.10.9.7 in /usr/local/lib/python3.10/dist-packages (from pyspark) (0.10.9.7)
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv

```

```

root@m1:~# pyspark
Python 3.10.12 (main, Jun 11 2023, 05:26:28) [GCC 11.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
23/09/17 19:47:49 WARN Utils: Your hostname, m1 resolves to a loopback address: 127.0.1.1; using 192.168.79.132 instead (on interface ens32)
23/09/17 19:47:49 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
23/09/17 19:47:51 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Welcome to

  ____      _
 / ___|    / \
| |  | |  / _ \
| |  | | / ___ \
| |  | |/_/   \_\
| |  | |
|_|  |_|

 version 3.5.0

Using Python version 3.10.12 (main, Jun 11 2023 05:26:28)
Spark context Web UI available at http://192.168.79.132:4040
Spark context available as 'sc' (master = local[*], app id = local-1694954872188).
SparkSession available as 'spark'.
>>>

```


3.2.3.2 Cài đặt Jupyter

```
phuongthao@m1:~/bigdata$ pip install jupyter
Defaulting to user installation because normal site-packages is not writeable
Collecting jupyter
  Downloading jupyter-1.0.0-py2.py3-none-any.whl (2.7 kB)
Collecting qtconsole
  Downloading qtconsole-5.4.4-py3-none-any.whl (121 kB)
    121.9/121.9 KB 1.7 MB/s eta 0:00:00
Collecting nbconvert
  Downloading nbconvert-7.8.0-py3-none-any.whl (254 kB)
    254.9/254.9 KB 2.0 MB/s eta 0:00:00
Collecting ipykernel
  Downloading ipykernel-6.25.2-py3-none-any.whl (154 kB)
    154.2/154.2 KB 17.5 MB/s eta 0:00:00
Collecting jupyter-console
  Downloading jupyter_console-6.6.3-py3-none-any.whl (24 kB)
Collecting notebook
  Downloading notebook-7.0.3-py3-none-any.whl (4.0 MB)
    4.0/4.0 MB 9.2 MB/s eta 0:00:00
Collecting ipywidgets
  Downloading ipywidgets-8.1.1-py3-none-any.whl (139 kB)
    139.4/139.4 KB 30.1 MB/s eta 0:00:00
Collecting traitlets>=5.4.0
  Downloading traitlets-5.9.0-py3-none-any.whl (117 kB)
    117.4/117.4 KB 22.4 MB/s eta 0:00:00
Collecting debugpy>=1.6.5
  Downloading debugpy-1.8.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (3.3 MB)
    3.3/3.3 MB 7.7 MB/s eta 0:00:00
Collecting jupyter-client>=6.1.12
  Downloading jupyter_client-8.3.1-py3-none-any.whl (104 kB)
    104.1/104.1 KB 13.9 MB/s eta 0:00:00
Collecting ipython>=7.23.1
  Downloading ipython-8.15.0-py3-none-any.whl (806 kB)
    806.6/806.6 KB 8.5 MB/s eta 0:00:00
Collecting tornado>=6.1
  Downloading tornado-6.3.3-cp38-abi3-manylinux_2_5_x86_64.manylinux1_x86_64.manylinux2014_x86_64.whl (427 kB)
    427.7/427.7 KB 4.8 MB/s eta 0:00:00
Collecting comm>=0.1.1
  Downloading comm-0.1.4-py3-none-any.whl (6.6 kB)
Collecting nest-asyncio
  Downloading nest_asyncio-1.5.7-py3-none-any.whl (5.3 kB)
Collecting packaging
```

Khởi động Jupyter

```
phuongthao@m1:~$ PYPARK_DRIVER_PYTHON="jupyter" PYPARK_DRIVER_PYTHON_OPTS="notebook" /home/phuongthao/bigdata/spark-3.5.0-bin-hadoop3/bin/pyspark --driver-memory 4g --driver-class-path /home/phuongthao/bigdata/elasticsearch-hadoop-5.3.0/dist/elasticsearch-spark-20_2.11-5.3.0.jar
[I 2023-09-13 22:31:28.184 ServerApp] Package notebook took 0.0000s to import
[I 2023-09-13 22:31:28.193 ServerApp] Package jupyter_lsp took 0.0078s to import
[W 2023-09-13 22:31:28.193 ServerApp] A '_jupyter_server_extension_points' function was not found in jupyter_lsp. Instead, a '_jupyter_server_extension_paths' function was found and will be used for now. This function name will be deprecated in future releases of Jupyter Server.
[I 2023-09-13 22:31:28.200 ServerApp] Package jupyter_server_terminals took 0.0068s to import
[I 2023-09-13 22:31:28.200 ServerApp] Package jupyterlab took 0.0000s to import
[I 2023-09-13 22:31:28.249 ServerApp] Package notebook_shim took 0.0000s to import
[W 2023-09-13 22:31:28.249 ServerApp] A '_jupyter_server_extension_points' function was not found in notebook_shim. Instead, a '_jupyter_server_extension_paths' function was found and will be used for now. This function name will be deprecated in future releases of Jupyter Server.
[I 2023-09-13 22:31:28.250 ServerApp] jupyter_lsp | extension was successfully linked.
[I 2023-09-13 22:31:28.258 ServerApp] jupyter_server_terminals | extension was successfully linked.
[I 2023-09-13 22:31:28.265 ServerApp] jupyterlab | extension was successfully linked.
[I 2023-09-13 22:31:28.273 ServerApp] notebook | extension was successfully linked.
[I 2023-09-13 22:31:28.521 ServerApp] notebook_shim | extension was successfully linked.
[I 2023-09-13 22:31:28.546 ServerApp] notebook_shim | extension was successfully loaded.
[I 2023-09-13 22:31:28.550 ServerApp] jupyter_lsp | extension was successfully loaded.
[I 2023-09-13 22:31:28.552 ServerApp] jupyter_server_terminals | extension was successfully loaded.
[I 2023-09-13 22:31:28.553 LabApp] JupyterLab extension loaded from /home/phuongthao/.local/lib/python3.10/site-packages/jupyterlab
[I 2023-09-13 22:31:28.554 LabApp] JupyterLab application directory is /home/phuongthao/.local/share/jupyter/lab
[I 2023-09-13 22:31:28.554 LabApp] Extension Manager is 'pypi'.
[I 2023-09-13 22:31:28.558 ServerApp] jupyterlab | extension was successfully loaded.
[I 2023-09-13 22:31:28.560 ServerApp] notebook | extension was successfully loaded.
[I 2023-09-13 22:31:28.561 ServerApp] Serving notebooks from local directory: /home/phuongthao
```

Cài đặt Numpy

```
phuongthao@n1:~/bigdata$ pip install numpy
Defaulting to user installation because normal site-packages is not writeable
Collecting numpy
  Downloading numpy-1.25.2-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (18.2 MB)
    18.2/18.2 MB 8.9 MB/s eta 0:00:00
Installing collected packages: numpy
  WARNING: The scripts f2py, f2py3 and f2py3.10 are installed in '/home/phuongthao/.local/bin' which
  is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-sc
  ript-location.
Successfully installed numpy-1.25.2
phuongthao@n1:~/bigdata$
```

3.2.4 Thử nghiệm

Nhóm có 2 file dữ liệu: product.csv và user.csv

3.4 Code python đưa các số liệu cơ bản từ file product.csv

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

```
# lấy dữ liệu từ đường dẫn
path = '/home/phuongthao/bigdata/product.csv'
dataframe = pd.read_csv(path)
dataframe.head(10)
```

```
#Kiểm tra kiểu dữ liệu của cột Date
print(dataframe['Date'].dtype)
```

```
#Tạo thêm cột Month
dataframe['Month'] = ''
```

```
#tách dữ liệu tháng từ cột Date, add vào cột Month vừa tạo ra.
dataframe['Month'] = dataframe['Date'].str.slice(0,2,1)
```

```
dataframe.head(10)
```

```
#kiểm tra lại cột tháng có dữ liệu nào sai form không, bằng cách  
in ra các tháng trong cột Month
```

```
print(set(dataframe['Month']))
```

```
{'02', '11', '07', '03', '10', '09', '05', '04', '06', '08', '01', '12'}
```

#Câu hỏi số 1: Loại sản phẩm nào được khách hàng quan tâm Reaction nhiều nhất trong năm?

#TÍNH TỔNG LƯỢNG Reaction CỦA CÁC Category Product Facebook

```
dataframe.groupby('Category').sum()['Reaction']
```

```
...
Category
Baby Essentials      688774
Bags                 82400
Beauty & Skin Care  1004402
Body Paint          129800
Clothing            5012765
Drinks & Food       128686
Fashion Wear        928958
Footwear           101804
Grocery              5137
Hair Care           52754
Home Decoration     137887
Jewellery           596167
Mobile Phone        7600
Modest Wear         3651031
Saree               64000
Name: Reaction, dtype: int64
```

#lượng reaction cao nhất của một Category Product Facebook

```
reaction_max = dataframe.groupby('Category').sum()['Reaction']
```

```
reaction_max.max()
```

... 5012765

#lấy tất cả Category ProductFB gán vào 1 biến

```
nameCa = dataframe['Category'].unique()
```

```
plt.bar(x=nameCa, height=reaction_max)
```

```
plt.xticks(nameCa, rotation = 90) #Hiển thị tên Category ở cột giá trị,  
chỉnh chữ nghiêng
```

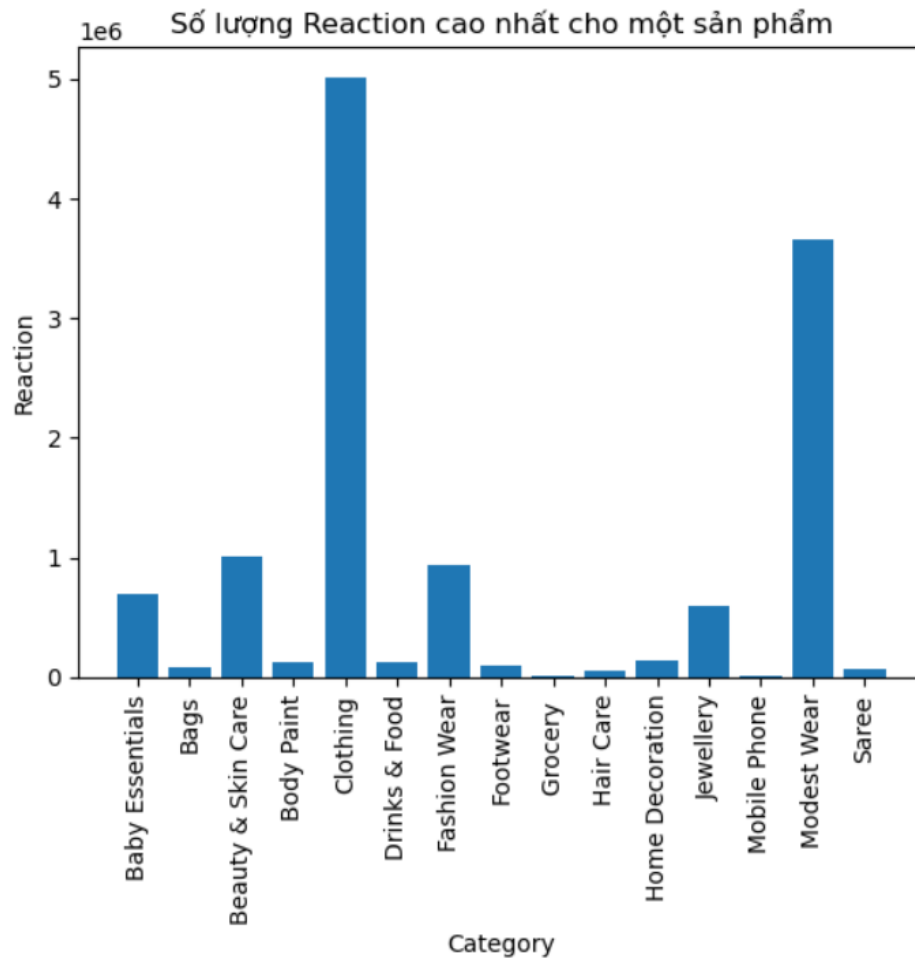
```
plt.xlabel('Category') #Tên hàng ngang
```

```
plt.ylabel('Reaction') #ten chiều dọc
```

```
plt.title('Số lượng Reaction cao nhất cho một sản phẩm')
```

```
plt.show()
```

...



#Câu hỏi số 2: Facebook Product Category nào được người dùng Share nhiều nhất?

#Groupby các Category cùng loại, hiện ra tổng lượt Comments của Category Product Facebook trong 1 năm

dataframe.groupby('Category').sum()['Comments']


```
...
Category
Baby Essentials      120412
Bags                 61155
Beauty & Skin Care   441028
Body Paint           14619
Clothing             3785841
Drinks & Food        36995
Fashion Wear         496829
Footwear             122712
Grocery              1276
Hair Care            3379
Home Decoration      98877
Jewellery            389276
Mobile Phone         4700
Modest Wear          1344818
Saree                77000
Name: Comments, dtype: int64
```

#Hiện ra lượng Comments cao nhất trong năm

```
cmt_max = dataframe.groupby('Category').sum()['Comments']
cmt_max.max()
```

#Gán tất cả các Category vào 1 biến

```
nameCmt = dataframe['Category'].unique()
```

```
plt.bar(x = nameCmt, height = cmt_max)
```

```
plt.xticks(nameCmt, rotation = 90)
```

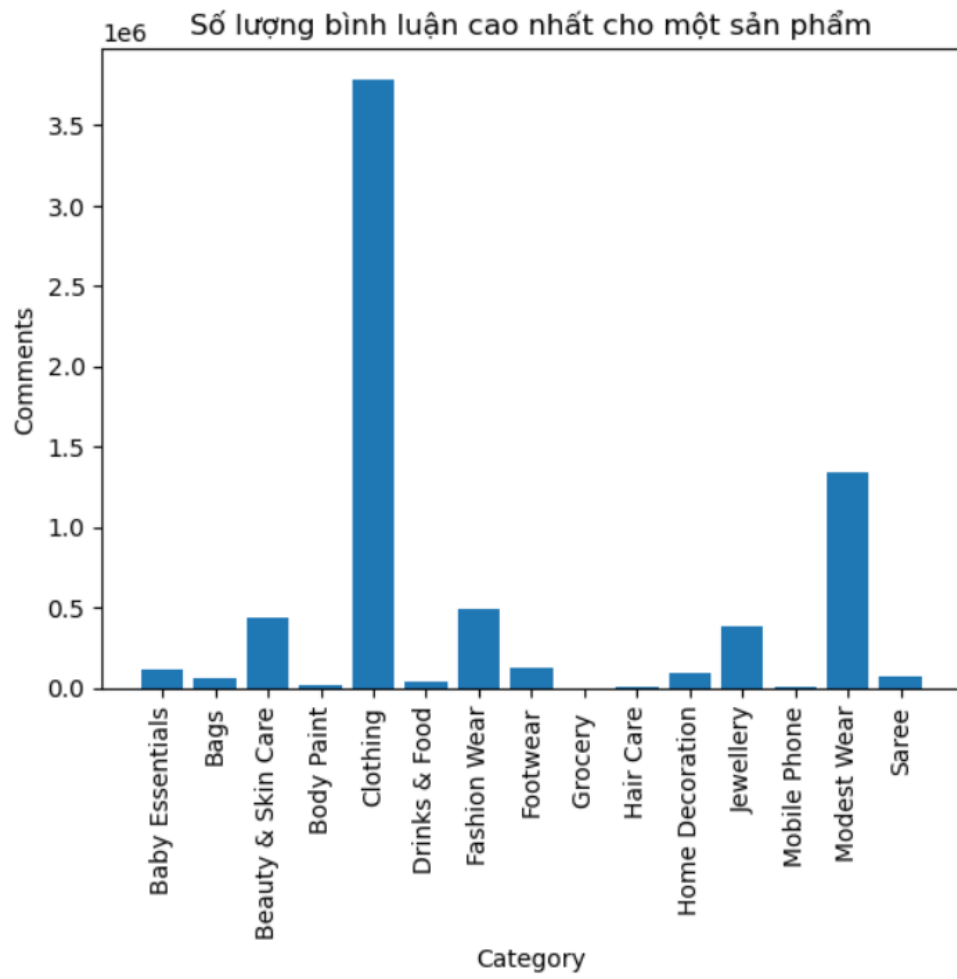
```
plt.xlabel('Category')
```

```
plt.ylabel('Comments')
```

```
plt.title('Số lượng bình luận cao nhất cho một sản phẩm')
```

```
plt.show()
```

...



#Câu hỏi số 3: có bao nhiêu sản phẩm trong 1 Facebook Product Category?

#đếm lượng sản phẩm của các Category Product Facebook

```
dataframe.groupby('Category').count()['Product Name']
```

```
... Category
Baby Essentials      109
Bags                 53
Beauty & Skin Care   579
Body Paint           33
Clothing             3164
Drinks & Food         86
Fashion Wear         453
Footwear             62
Grocery              11
Hair Care            21
Home Decoration      51
Jewellery            377
Mobile Phone         1
Modest Wear          498
Saree                20
Name: Product Name, dtype: int64
```

#Sau khi đếm số lượng sản phẩm của 1 Category, ta gán vào 1 biến

```
countSP = dataframe.groupby('Category').count()['Product Name']
```

#gán tất cả Category vào 1 biến

```
nameProName = dataframe['Category'].unique()
```

#dùng plt

```
plt.bar(x = nameProName, height = countSP)
```

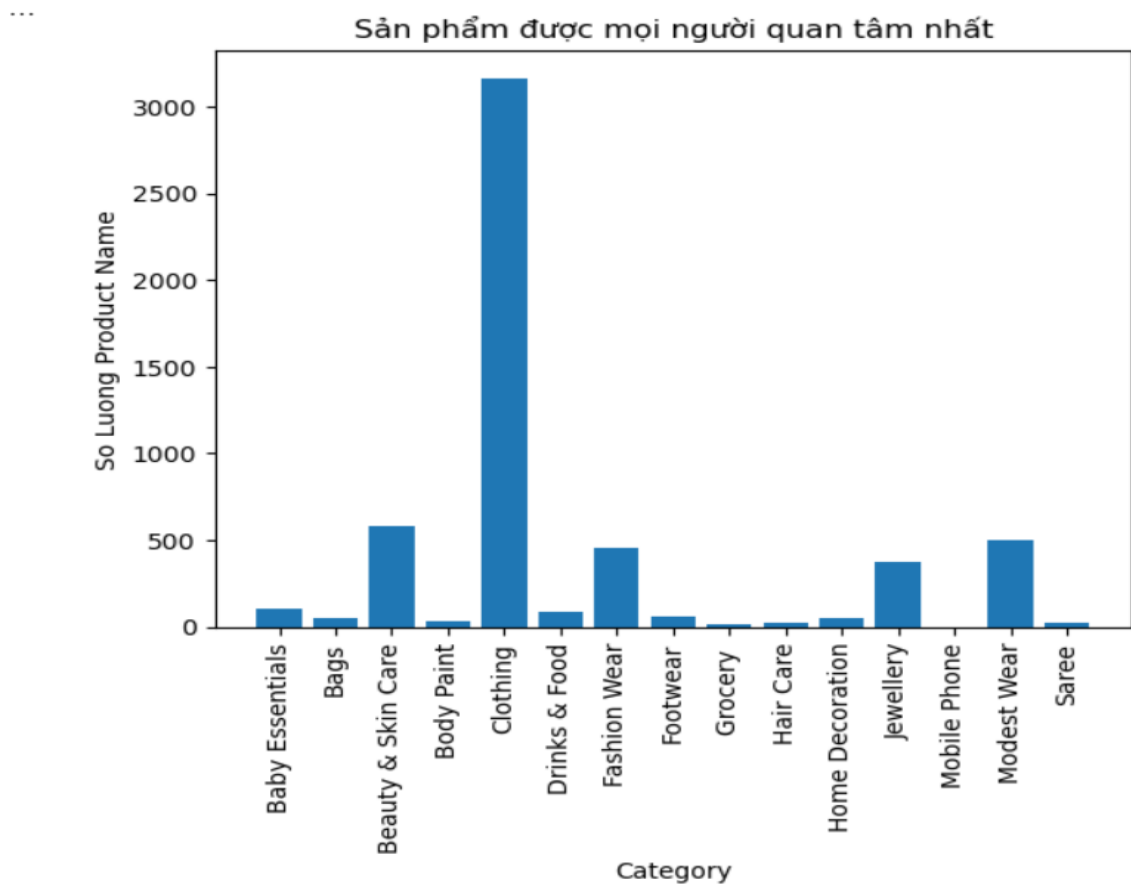
```
plt.xticks(nameProName, rotation = 90)
```

```
plt.xlabel('Category')
```

```
plt.ylabel('So Luong Product Name')
```

```
plt.title('Sản phẩm được mọi người quan tâm nhất')
```

```
plt.show()
```



#Câu hỏi số 4: Sản phẩm Clothing được độ tuổi nào quan tâm nhiều nhất?

```
Clothing_pro = dataframe[dataframe['Category'] == 'Jewellery']
```

```
AgeCount = Clothing_pro['Age'].value_counts()
```

```
AgeCount = Clothing_pro['Age'].value_counts().sort_index()
```

Vẽ biểu đồ đường

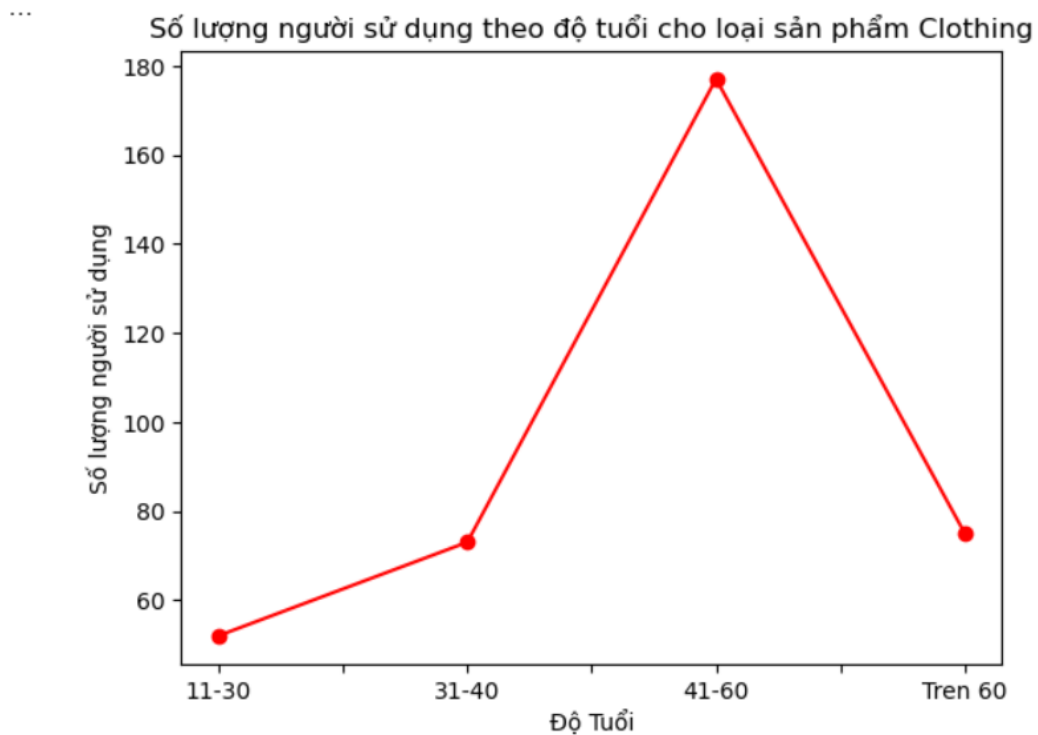
```
AgeCount.plot.line(marker='o', linestyle='-', color='red')
```

```
plt.title('Số lượng người sử dụng theo độ tuổi cho loại sản phẩm  
Clothing')
```

```
plt.xlabel('Độ Tuổi')
```

```
plt.ylabel('Số lượng người sử dụng')
```

```
plt.show()
```



#Câu hỏi số 6: Tính tổng tỷ lệ của từng độ tuổi quan tâm đến Facebook Product Category

```
print(set(dataframe['Age']))
```

```
| ... {'11-30', '41-60', '31-40', '1-10', 'Tren 60'}
```

```
Age_sort = dataframe.sort_values(by='Age')
```

```
Age_ratio = Age_sort['Age'].value_counts(normalize=True)
```

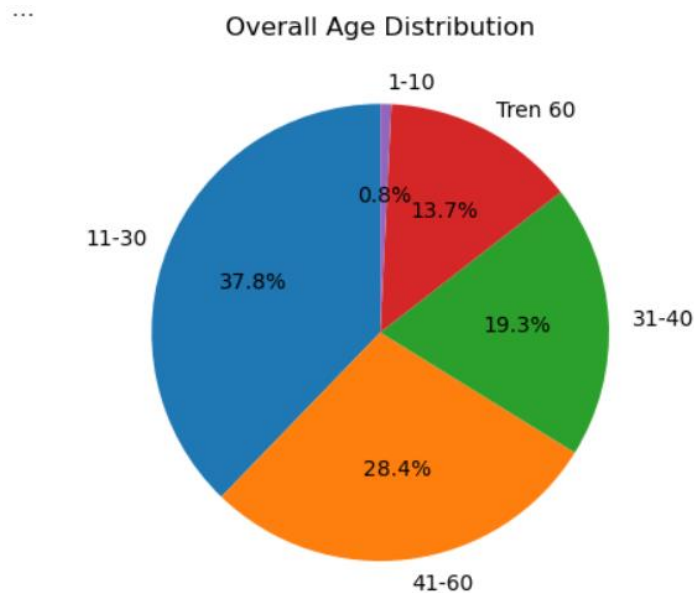
Tạo biểu đồ pie chart tổng hợp

```
Age_ratio.plot.pie(autopct='%1.1f%%', startangle=90)
```

```
plt.title('Overall Age Distribution')
```

```
plt.ylabel('')
```

`plt.show()`



3.5 Kết hợp với Machine Learning xử lý file *product.csv* và *user.csv*

Dùng thuật toán *Regression* hồi quy tuyến tính để dự đoán số liệu cho file *product.csv*

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns; sns.set()
from sklearn import datasets
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score
```

```

from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.neural_network import MLPRegressor
from sklearn.ensemble import RandomForestRegressor

#lấy dữ liệu
path = '/home/phuongthao/bigdata/product.csv'
df = pd.read_csv(path)
df.head(10)
# Chuyển đổi cột dấu thời gian sang định dạng thời gian
df.index = pd.to_datetime(df['Date']).dt.floor('T')
df = df.iloc[:, 1:]
#Tạo một tập hợp tên cột phụ trợ trong các giá trị đo được
product_measurement_data = ['Category', 'Reaction', 'Comments',
'Shared']
# Loại bỏ hàng "count" khỏi bảng tóm tắt thống kê
summary = df[product_measurement_data].describe().drop("count")
summary.head()

# Hiển thị tóm tắt thống kê cho các cột đã chọn
df[product_measurement_data].describe()

# Màu
color_1 = '#BFAF9D'
color_2 = '#C1C1D5'
color_3 = '#FF0000'
#in ra màn hình biểu đồ dự đoán theo tháng
plt.figure(figsize=(20, 30))

```

```
plt.subplot(611)
```

```
sns.lineplot(data=df, x='Month', y='Reaction', color=color_3)
```

```
plt.subplot(612)
```

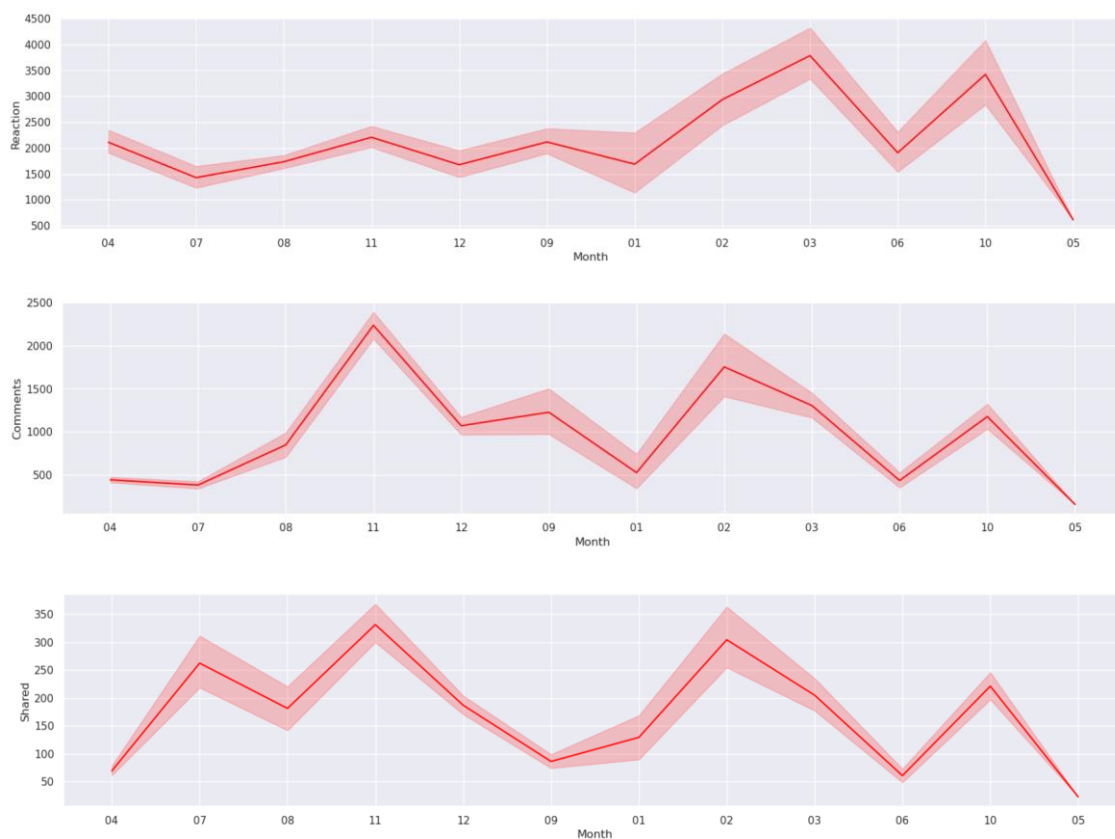
```
sns.lineplot(data=df, x='Month', y='Comments', color=color_3)
```

```
plt.subplot(613)
```

```
sns.lineplot(data=df, x='Month', y='Shared', color=color_3)
```

```
plt.subplots_adjust(hspace=0.3)
```

```
plt.show()
```



Hiển thị tóm tắt thống kê cho các cột đã chọn

```
df[product_measurement_data].describe()
```

Màu

```
color_1 = '#BFAF9D'
```

```
color_2 = '#C1C1D5'
```

```
color_3 = '#FF0000'
```

#in ra màn hình biểu đồ dự đoán theo năm

```
plt.figure(figsize=(20, 30))
```

```
plt.subplot(611)
```

```
sns.lineplot(data=df, x=df.index, y='Reaction', color=color_3)
```

```
plt.subplot(612)
```

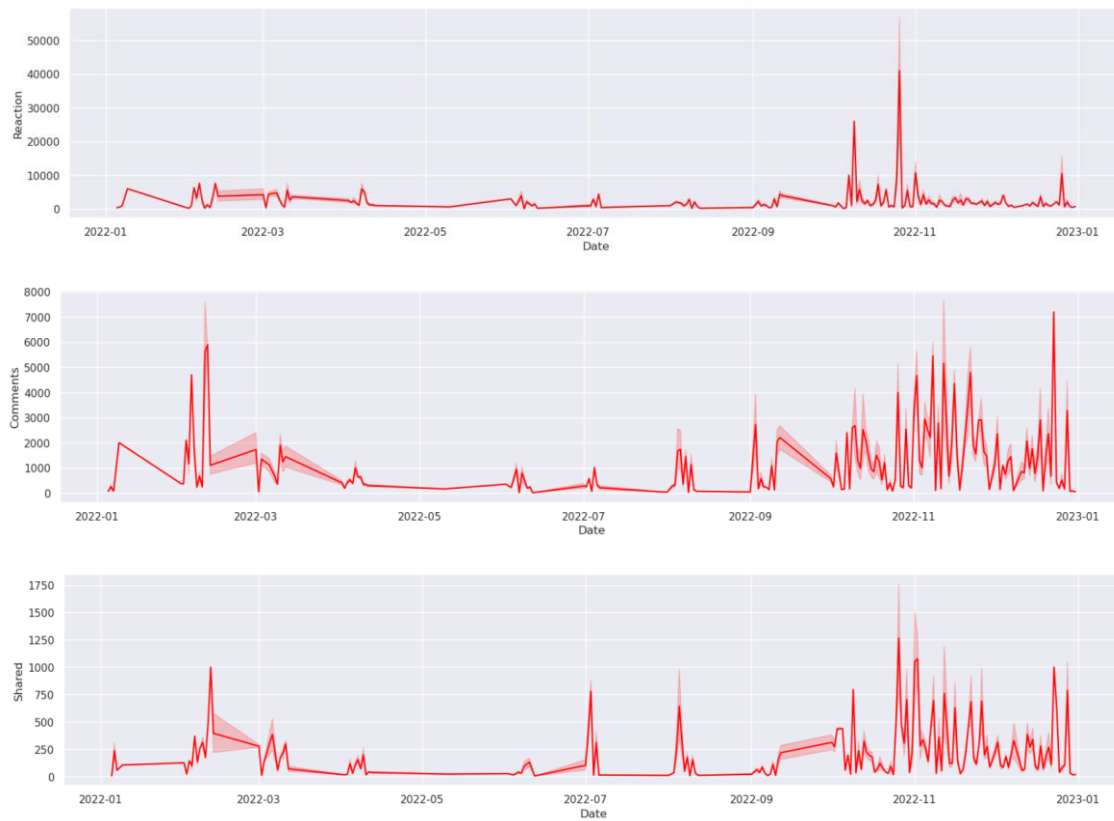
```
sns.lineplot(data=df, x=df.index, y='Comments', color=color_3)
```

```
plt.subplot(613)
```

```
sns.lineplot(data=df, x=df.index, y='Shared', color=color_3)
```

```
plt.subplots_adjust(hspace=0.3)
```

```
plt.show()
```



#train máy dự đoán cột Reaction

```
df['Date (month)'] = pd.to_datetime(df.index).month
```

```
df['Date (day)'] = pd.to_datetime(df.index).day
```

```
param_ml_input = ['Date (month)', 'Date (day)', 'Reaction', 'Comments',  
'Shared']
```

```
param_ml_output = 'Reaction'
```

```
X_data = df[param_ml_input]
```

```
y_data = df[param_ml_output]
```

```
X_train, X_test, y_train, y_test = train_test_split(X_data, y_data,  
test_size = 1/3)
```

```
scaler = StandardScaler()
```

```
scaler.fit(X_train)
```

```
X_train_scaled = scaler.transform(X_train)
```

```
X_test_scaled = scaler.transform(X_test)
```

```
#Regression
```

```
mlp = MLPRegressor(max_iter = 5000).fit(X_train_scaled, y_train)
```

```
y_pred = mlp.predict(X_test_scaled)
```

```
result = pd.DataFrame({'Giá trị thực tế (y_test)': y_test,  
                       'Giá trị được mô hình dự đoán (y_pred)': y_pred,  
                       'Sự khác biệt': abs(y_pred - y_test)})
```

```
result.head(10)
```

```
plt.figure(figsize=(10, 6))
```

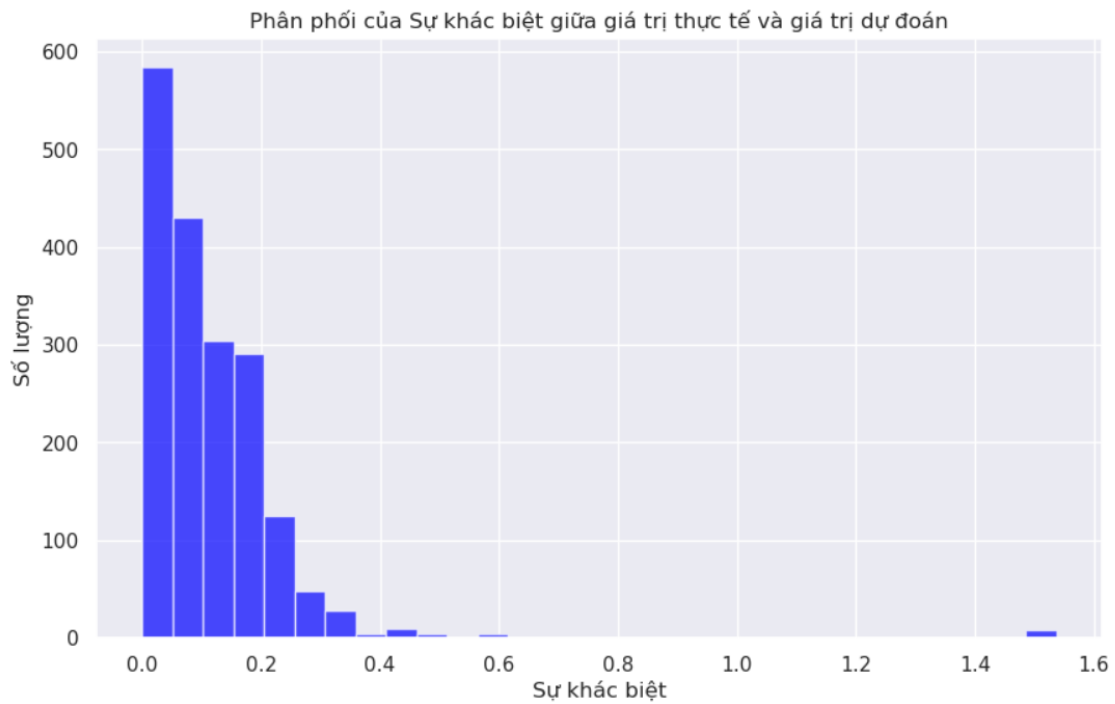
```
plt.hist(result['Sự khác biệt'], bins=30, color='blue', alpha=0.7)
```

```
plt.xlabel('Sự khác biệt')
```

```
plt.ylabel('Số lượng')
```

```
plt.title('Phân phối của Sự khác biệt giữa giá trị thực tế và giá trị dự  
đoán')
```

```
plt.show()
```

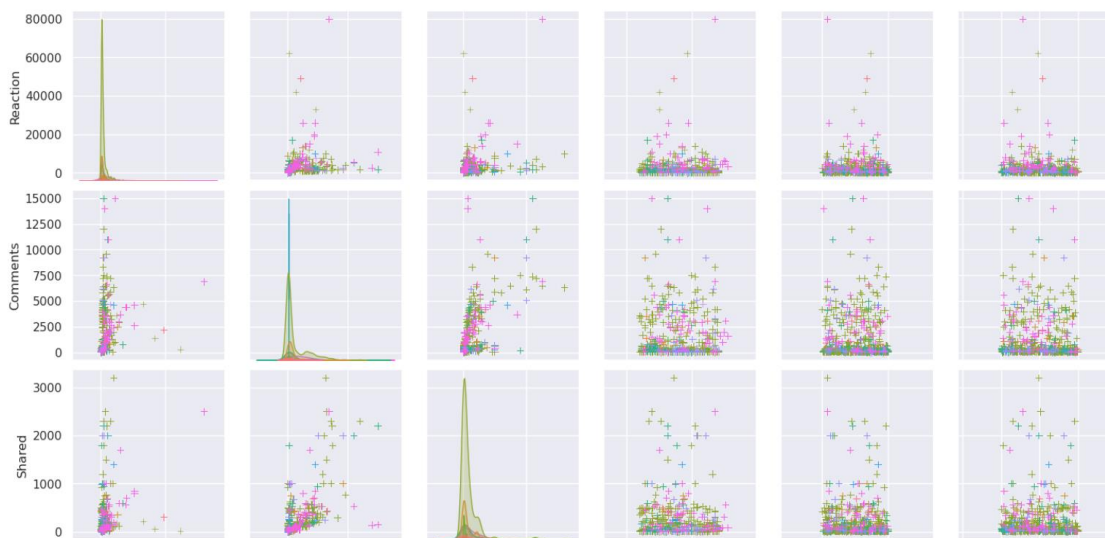


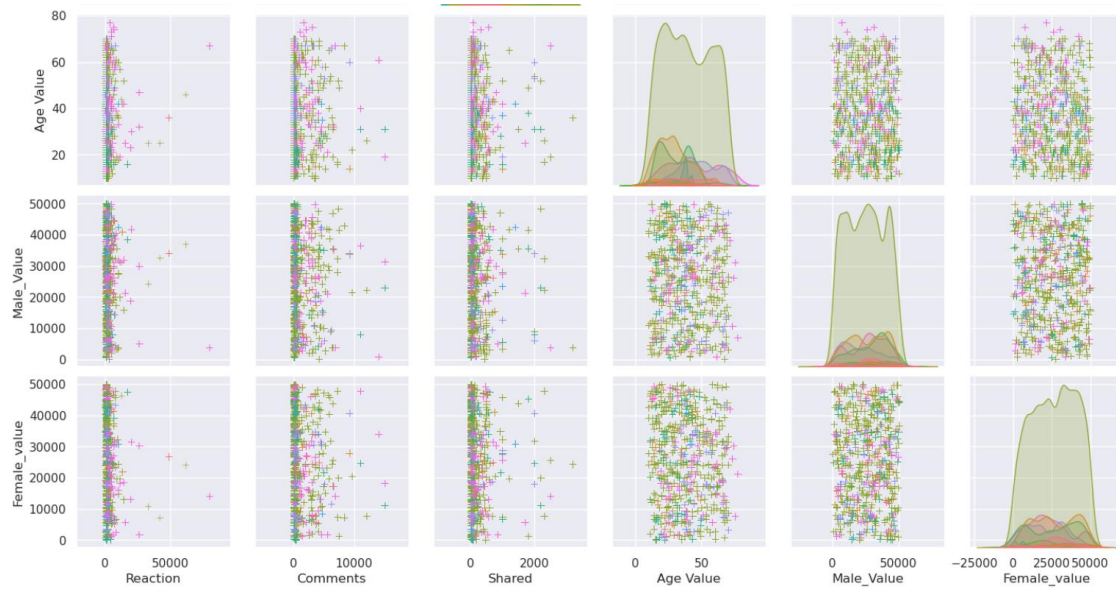
Thêm các thư viện seaborn và matplotlib

```
product_measurement_data = ['Category', 'Reaction', 'Comments',  
'Shared']
```

```
g = sns.pairplot(df, hue='Category', markers='+')
```

```
plt.show()
```





- Category**
- Baby Essentials
 - Bags
 - Beauty & Skin Care
 - Body Paint
 - Clothing
 - Drinks & Food
 - Fashion Wear
 - Footwear
 - Grocery
 - Hair Care
 - Home Decoration
 - Jewellery
 - Mobile Phone
 - Modest Wear
 - Saree

Dùng thuật toán Regression hồi quy tuyến tính để dự đoán số liệu cho file user.csv

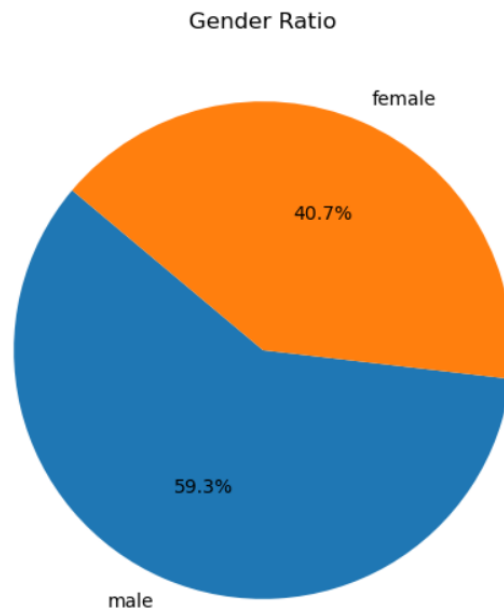
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

from sklearn.neighbors import KernelDensity
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error

#lay du lieu
path = '/home/phuongthao/bigdata/user.csv'
df = pd.read_csv(path)
df.head(10)

# Thống kê tỷ lệ nam và nữ
gender_counts = df["gender"].value_counts()

# Vẽ biểu đồ tròn
plt.figure(figsize=(6, 6))
plt.pie(gender_counts, labels=gender_counts.index, autopct='%1.1f%%',
startangle=140)
plt.title("Gender Ratio")
plt.show()
```



df.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 990030 entries, 0 to 990029
Data columns (total 7 columns):
#   Column          Non-Null Count  Dtype
---  -
0   userid          990030 non-null  int64
1   age             990030 non-null  int64
2   dob_day         990030 non-null  int64
3   dob_year        990030 non-null  int64
4   dob_month       990030 non-null  int64
5   gender          988280 non-null  object
6   status_type     990030 non-null  object
dtypes: int64(5), object(2)
memory usage: 52.9+ MB
```

Tính tỷ lệ của cột 'status_type'

status_type_counts = df['status_type'].value_counts()

total_count = len(df)

status_type_percentage = status_type_counts / total_count

Vẽ biểu đồ dạng line với đường độ lệch chuẩn

```
%matplotlib inline
```

```
plt.figure(figsize=(8, 6))
```

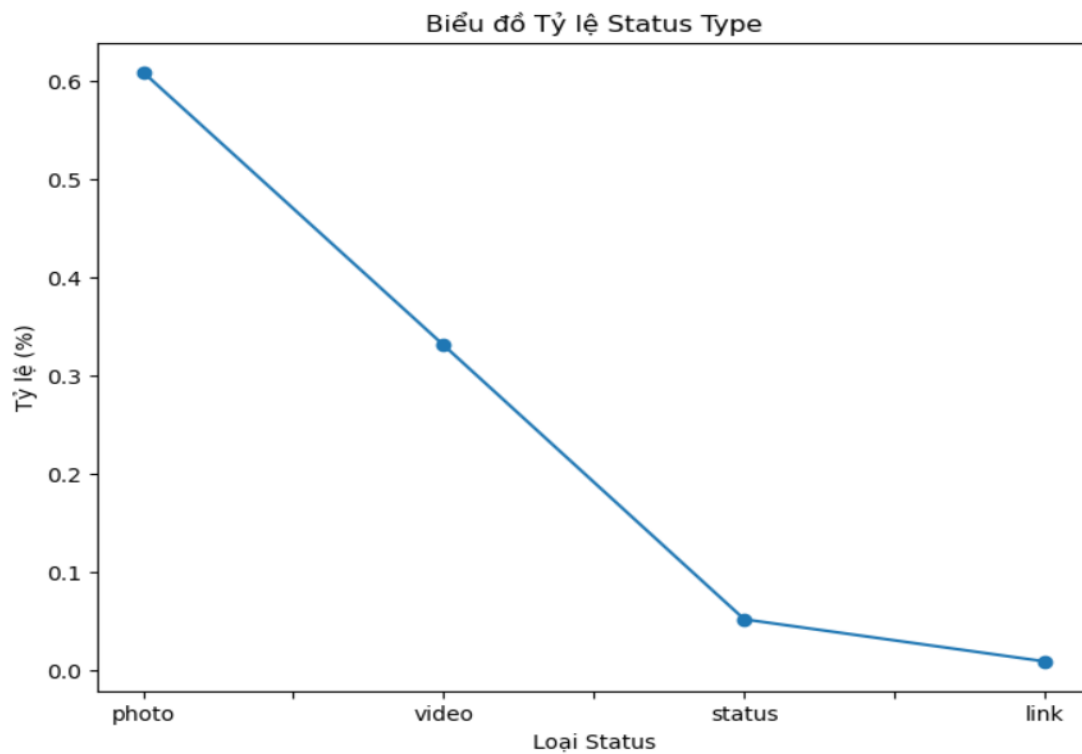
```
status_type_percentage.plot(kind='line', marker='o')
```

```
plt.title('Biểu đồ Tỷ lệ Status Type')
```

```
plt.xlabel('Loại Status')
```

```
plt.ylabel('Tỷ lệ (%)')
```

```
plt.show()
```



Tạo tập hợp tên cột phụ trợ trong các giá trị đo được

```
post_measurement_data = ['age', 'dob_year', 'status_type']
```

Mã hóa one-hot cho cột 'status_type' và 'gender'

```
df_encoded = pd.get_dummies(df, columns=['status_type', 'gender'],
```

```
prefix=['status', 'gender'])
```



```

# Danh sách các loại status_type
status_types = df['status_type'].unique()

# Tạo một đối tượng DataFrame để lưu trữ kết quả dự đoán cho từng
status_type
predicted_ages = pd.DataFrame()

# Xây dựng mô hình hồi quy tuyến tính cho từng status_type
for status_type in status_types:
    # Lọc dữ liệu cho status_type cụ thể
    df_filtered = df_encoded[df_encoded[f'status_{status_type}'] == 1]

    # Chia dữ liệu thành dữ liệu huấn luyện và dữ liệu kiểm tra
    X = df_filtered.drop(['age', f'status_{status_type}', 'gender_female',
'gender_male'], axis=1)
    y = df_filtered['age']
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=0)

    # Xây dựng mô hình hồi quy tuyến tính
    model = LinearRegression()
    model.fit(X_train, y_train)

    # Dự đoán độ tuổi cho dữ liệu kiểm tra
    y_pred = model.predict(X_test)

    # Tạo một DataFrame tạm thời để lưu trữ kết quả dự đoán

```

```
temp_result = pd.DataFrame({'status_type': [status_type] * len(y_test),
                             'age_true': y_test, 'age_predicted': y_pred})
```

Kết hợp kết quả dự đoán vào DataFrame tổng hợp

```
predicted_ages = pd.concat([predicted_ages, temp_result])
```

Hiển thị kết quả dự đoán

```
print(predicted_ages)
```

Đánh giá mô hình (đoạn này có thể thực hiện sau khi dự đoán cho tất cả các status_type)

```
mse = mean_squared_error(predicted_ages['age_true'],
```

```
predicted_ages['age_predicted'])
```

```
print(f'Mean Squared Error: {mse}')
```

	status_type	age_true	age_predicted
209634	video	26	26.0
64664	video	16	16.0
554130	video	21	21.0
540066	video	51	51.0
469463	video	22	22.0
...
584786	status	103	103.0
755654	status	18	18.0
332654	status	62	62.0
60269	status	14	14.0
424031	status	22	22.0

```
[198007 rows x 3 columns]
```

```
Mean Squared Error: 4.601691904249199e-19
```

```
import matplotlib.pyplot as plt
```

Tạo biểu đồ điểm cho dự đoán độ tuổi theo từng status_type

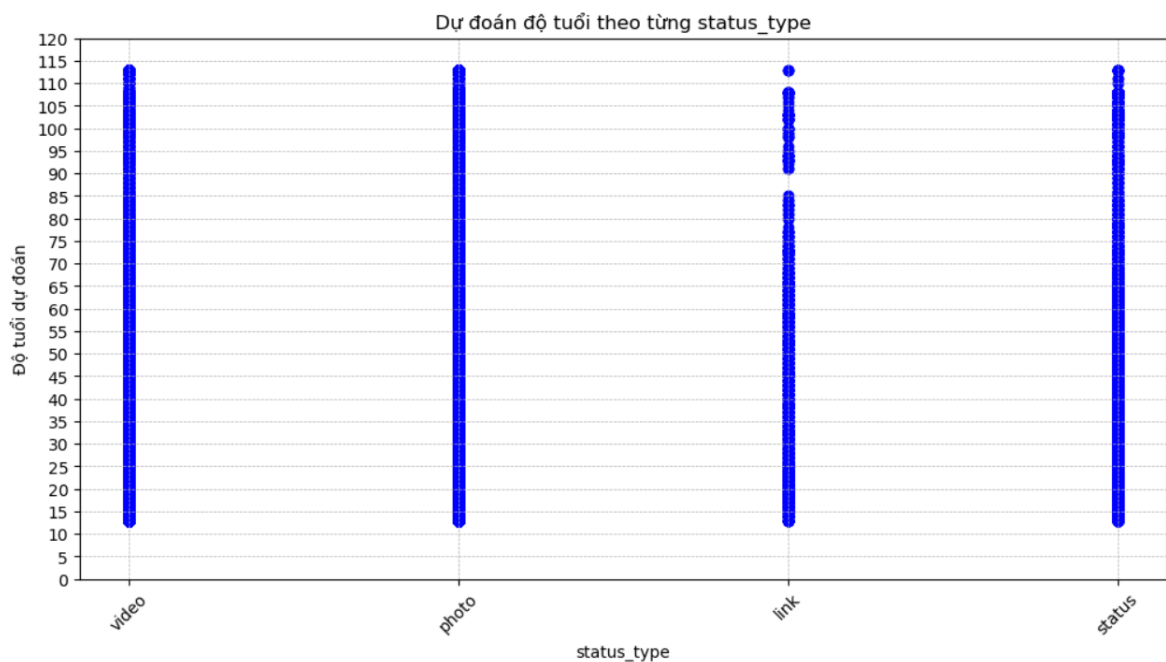
```

plt.figure(figsize=(12, 6))
plt.scatter(predicted_ages['status_type'], predicted_ages['age_predicted'],
            color='blue')
plt.xlabel('status_type')
plt.ylabel('Độ tuổi dự đoán')
plt.title('Dự đoán độ tuổi theo từng status_type')
plt.xticks(rotation=45)

# Đặt khoảng cách của trục y cách nhau 5 đơn vị và bắt đầu từ 0
plt.yticks(range(0, int(max(predicted_ages['age_predicted']) + 10), 5))

plt.grid(True, which='both', linestyle='--', linewidth=0.5)
plt.show()

```



```

# Tạo biểu đồ điểm cho độ tuổi theo từng status_type
plt.figure(figsize=(10, 6))
for status_type, group in df.groupby('status_type'):

```

```
plt.scatter(group['status_type'], group['age'], label=status_type,
alpha=0.7)
```

```
plt.title('Độ Tuổi Theo Từng status_type')
```

```
plt.xlabel('status_type')
```

```
plt.ylabel('Độ Tuổi')
```

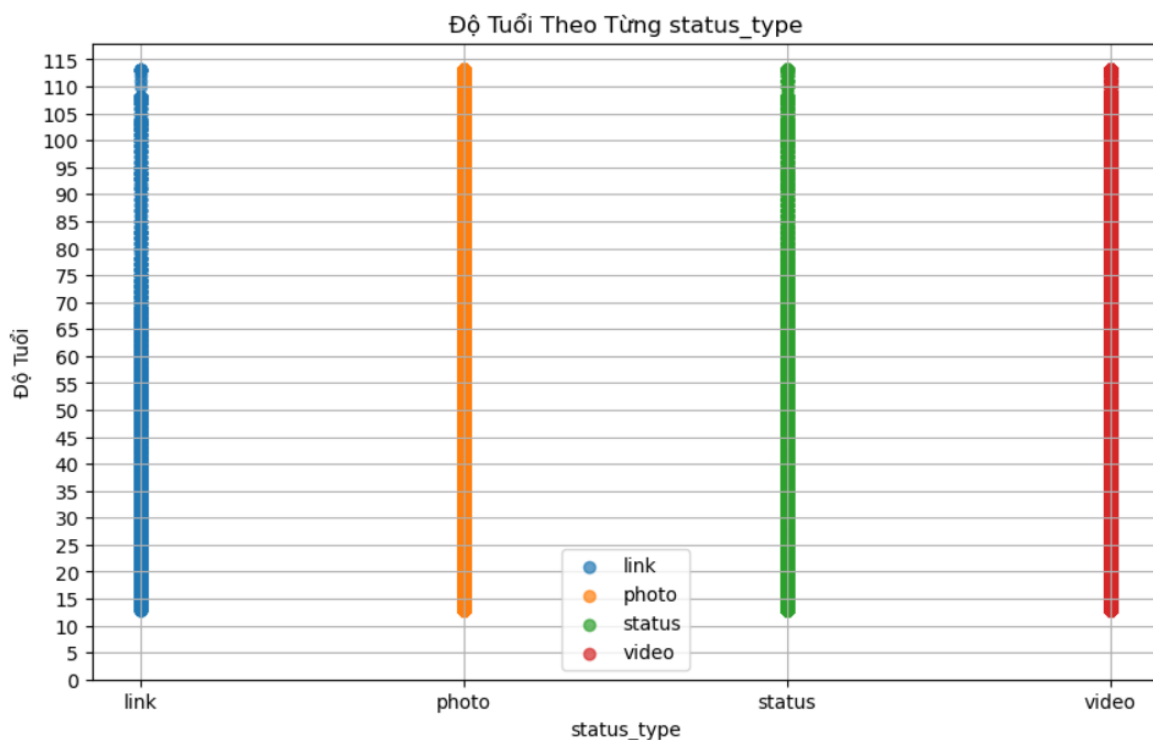
```
plt.legend()
```

```
plt.grid(True)
```

Đặt khoảng cách giữa các giá trị trên trục y là 5 đơn vị

```
plt.yticks(range(0, max(df['age'])+5, 5))
```

```
plt.show()
```



Tạo biểu đồ cho dữ liệu thực và dự đoán

```
fig, axs = plt.subplots(1, 2, figsize=(16, 6))
```

Biểu đồ cho dữ liệu thực

```
for status_type, group in df.groupby('status_type'):
    axs[0].scatter(group['status_type'], group['age'], label=status_type,
alpha=0.7, marker='o')
```

```
axs[0].set_title('Độ Tuổi Thực Theo Từng status_type')
```

```
axs[0].set_xlabel('status_type')
```

```
axs[0].set_ylabel('Độ Tuổi')
```

```
axs[0].legend()
```

```
axs[0].grid(True)
```

Biểu đồ cho dữ liệu dự đoán

```
axs[1].scatter(predicted_ages['status_type'],
predicted_ages['age_predicted'], color='blue', label='Dự đoán',
alpha=0.7, marker='x')
```

```
axs[1].set_title('Dự Đoán Độ Tuổi Theo Từng status_type')
```

```
axs[1].set_xlabel('status_type')
```

```
axs[1].set_ylabel('Độ Tuổi Dự Đoán')
```

```
axs[1].xaxis.set_tick_params(rotation=45)
```

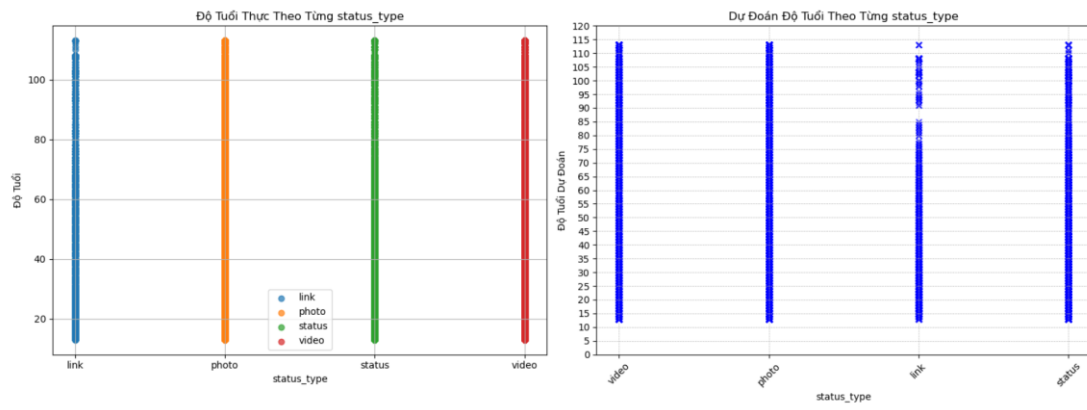
```
axs[1].set_yticks(range(0, int(max(predicted_ages['age_predicted']) + 10),
5))
```

```
axs[1].grid(True, which='both', linestyle='--', linewidth=0.5)
```

```
plt.tight_layout()
```

```
plt.show()
```

...



III. TỔNG KẾT

1. Đề xuất

Apache Spark xử lý các dữ liệu đưa ra kết quả về sự quan tâm của người dùng và sản phẩm, bạn có thể tạo nội dung tùy chỉnh và quảng cáo phù hợp với từng người dùng. Xây dựng mối quan hệ sâu hơn với người dùng bằng cách cung cấp giá trị thực sự dựa trên nhu cầu và mong muốn của họ, từ đó tạo ra tương tác tốt hơn và cải thiện tỷ lệ chuyển đổi. Apache Spark quản trị bao gồm Machine Learning có thể đưa ra các dự đoán các số liệu liên quan đến nhu cầu của người dùng, có thể áp dụng trong việc quảng cáo trên Facebook.

2. Đánh giá

2.1 Ưu điểm

- *Hiệu quả cao:* Apache Spark là một framework xử lý dữ liệu lớn, cho phép thu thập và xử lý dữ liệu từ Facebook một cách hiệu quả.
- *Tùy chỉnh nội dung:* Dựa trên dữ liệu về sự quan tâm của người dùng, bạn có thể tạo nội dung tùy chỉnh và quảng cáo phù hợp với từng người dùng.
- *Chi phí không cao:* Là hình thức Marketing hoàn toàn miễn phí, hiệu quả cao, chi phí thấp.

2.2 Nhược điểm

- *Độ chính xác:* Có thể có những sai sót trong quá trình thu thập và xử lý dữ liệu.
- *Phụ thuộc vào API của Facebook:* Việc thu thập dữ liệu phụ thuộc vào API của Facebook, nếu có bất kỳ thay đổi nào từ phía Facebook có thể ảnh hưởng đến việc thu thập dữ liệu.

3. Kết luận

Big Data, kết hợp với Deep Learning, Machine Learning và AI, đã mở ra một thế giới mới với nhiều lợi ích to lớn. Thế giới hiện đang chứng kiến một cuộc cách mạng công nghiệp mới được thúc đẩy bởi các công nghệ Big Data, Internet

kết nối vạn vật và tự động hóa. Sự giao thoa giữa các xu hướng công nghệ và các vấn đề trong phát triển kinh tế - xã hội đã tạo ra những khối lượng dữ liệu khổng lồ, gọi chung là Big Data. Đây chính là nguồn lực để thúc đẩy hình thành các ngành công nghiệp, các quy trình sản xuất kinh doanh và tạo ra sản phẩm mới.

Tại Việt Nam, Big Data không chỉ là một cơ hội để phát triển bức phá trong các lĩnh vực, mà còn là một bài toán và thách thức rất lớn. Thách thức này không chỉ đến từ việc chọn lựa công nghệ phù hợp để bảo vệ thông tin, mà còn từ việc xây dựng chính sách quản lý hiệu quả. Để có được chính sách thông minh, chúng ta cần những nhà quản trị xã hội thông minh và có kinh nghiệm trong những chuyên ngành này. Bên cạnh việc quản lý, việc phát triển các công ty công nghệ trong nước cũng rất quan trọng. Chính lực lượng này mới là nòng cốt của công nghệ quốc gia.

Big Data đang biến đổi thế giới theo cách không ai có thể tưởng tượng. Với sự phát triển của các công ty công nghệ trong nước và sự điều chỉnh của chính sách, Việt Nam có thể tận dụng tối đa lợi ích từ Big Data. Trong lĩnh vực Big Data, việc khai thác và quản lý dữ liệu đang có ảnh hưởng sâu sắc đến nhiều khía cạnh của cuộc sống. Một trong những lĩnh vực tiềm năng khác mà Big Data có thể ứng dụng mạnh mẽ là lĩnh vực tài chính. Dữ liệu tài chính rất lớn và phức tạp, từ giao dịch hàng ngày trên thị trường chứng khoán đến thông tin về ngân hàng và tín dụng của cá nhân. Big Data và các công nghệ liên quan có thể giúp phân tích dữ liệu này để dự đoán biến động thị trường, quản lý rủi ro tài chính, và tối ưu hóa các quyết định đầu tư. Ngoài ra, lĩnh vực môi trường cũng có tiềm năng lớn trong việc sử dụng Big Data. Dữ liệu về môi trường, khí hậu, và tài nguyên tự nhiên có thể được thu thập và phân tích để dự đoán biến đổi khí hậu, tối ưu hóa sử dụng tài nguyên, và quản lý các vấn đề liên quan đến môi trường. Việc sử dụng Big Data trong lĩnh vực này có thể giúp bảo vệ môi trường và tạo ra các giải pháp bền vững cho tương lai.

Chính xác, Big Data đang mở ra những cơ hội mới mẻ không chỉ cho Việt Nam mà còn cho toàn thế giới. Với sự phát triển của AI và Machine Learning, chúng ta có thể khai thác và phân tích dữ liệu lớn để tạo ra những giải pháp thông minh, tối ưu và cá nhân hóa.

Một trong những lĩnh vực tiềm năng nhất của Big Data là y tế. Bằng cách sử dụng dữ liệu từ các bệnh nhân và nghiên cứu y tế, các nhà khoa học có thể phát triển các phương pháp điều trị mới, dự đoán các bệnh tật và cải thiện chất lượng chăm sóc sức khỏe. Ngoài những lĩnh vực tiềm năng này, Big Data còn có thể ứng dụng trong nhiều ngành khác nhau như giáo dục, du lịch, và nhiều lĩnh vực khác. Tuy nhiên, để thực hiện tối ưu tiềm năng của Big Data, cần có sự đầu tư vào cơ sở hạ tầng công nghệ, nâng cao kiến thức và kỹ năng của người lao động, và đảm bảo an ninh thông tin và quyền riêng tư. Việc phát triển Big Data không chỉ mang lại lợi ích kinh tế mà còn góp phần vào sự phát triển và tiến bộ của xã hội.

Ngoài ra, Big Data cũng đóng vai trò quan trọng trong việc cải thiện hiệu quả của các doanh nghiệp. Các công ty có thể sử dụng dữ liệu để hiểu rõ hơn về khách hàng của mình, tối ưu hóa quy trình làm việc và đưa ra quyết định kinh doanh thông minh hơn. Tuy nhiên, việc sử dụng Big Data cũng đặt ra những thách thức về quyền riêng tư và an ninh thông tin. Chính vì vậy, việc xây dựng một chính sách quản lý dữ liệu hiệu quả và tuân thủ luật pháp là rất quan trọng.

Cuối cùng, để tận dụng tối đa lợi ích từ Big Data, Việt Nam cần tiếp tục đầu tư vào giáo dục và đào tạo, nhằm chuẩn bị cho lực lượng lao động có kỹ năng cần thiết để làm việc trong thời đại số hóa này. Đồng thời, việc hỗ trợ và khuyến khích sự phát triển của các công ty công nghệ trong nước cũng rất quan trọng để xây dựng một nền tảng công nghệ mạnh mẽ cho tương lai.

TÀI LIỆU THAM KHẢO

- [1] T. Nguyễn Lê Phương Hoài, “BIG DATA VÀ XU HƯỚNG ỨNG DỤNG TRONG HOẠT ĐỘNG THÔNG TIN - THƯ VIỆN”, tháng 2 2020. [Online]. Available at: http://tailieudientu.lrc.tnu.edu.vn/Upload/Collection/brief/218727_1012202010311948077-Article%20Text-151840-1-10-20200520.pdf
- [2] “BIG DATA LÀ GÌ? ĐẶC ĐIỂM VÀ ỨNG DỤNG BIG DATA VÀO CÁC NGÀNH”. [Online]. Available at: <https://www.pace.edu.vn/tin-kho-tri-thuc/big-data-la-gi#dac-trung-cua-big-data>
- [3] “Apache Spark™ history”. [Online]. Available at: <https://spark.apache.org/history.html>
- [4] Phuc Ngoc Nghia, “Tìm hiểu về Apache Spark”. [Online]. Available at: <https://viblo.asia/p/tim-hieu-ve-apache-spark-ByEZkQQW5Q0>
- [5] “Applications of Big Data”. [Online]. Available at: <https://www.interviewbit.com/blog/applications-of-big-data/>