TRƯỜNG ĐẠI HỌC TÀI NGUYÊN VÀ MÔI TRƯỜNG

TP. HỒ CHÍ MINH

**KHOA HỆ THỐNG THÔNG TIN VÀ VIỄN THÁM**



**Môn: BIGDATA**

Đề tài:  **PHÂN TÍCH BÀI ĐĂNG CỦA NGƯỜI NỔI TIẾNG TRÊN MẠNG XÃ HỘI TWITTER**

Nhóm: 02

Lớp: 10\_ĐH\_CNTT2

***Thành phố Hồ Chí Minh, tháng 11 năm 2023***

**MỤC LỤC**

[THÔNG TIN NHÓM 3](#_Toc168)

[BÁO CÁO TÓM TẮT 5](#_Toc10207)

[BÁO CÁO CHI TIẾT 6](#_Toc20471)

[1. Giới thiệu về bigdata. 6](#_Toc4202)

[1.1 định nghĩa. 6](#_Toc12042)

[1.2 Cách hoạt động. 7](#_Toc21677)

[1.3 Vai Trò 8](#_Toc15084)

[2. Bài toán 16](#_Toc12202)

[3. Giải pháp 17](#_Toc5164)

[4. Đánh giá 29](#_Toc13030)

[5. Kết luận 30](#_Toc29862)

**THÔNG TIN NHÓM**

|  |  |  |
| --- | --- | --- |
| Tên đề tài | Phân tích bài đăng của người nổi tiếng trên mạng xã hội twitter  - Link Youtube: <https://www.youtube.com/channel/UCd2ceSgOM7zNMf51xRIDqmw> | Đóng góp của thành viên |
| Họ và tên  thành  viên 1 | Phan Tân Phước – 1050080070 | Đóng góp: Coder chính, cài đặt thực nghiệm  Điểm tự đánh giá quá trình: 7  Điểm tự đánh giá cuối kì: 7 |
| Họ và tên  thành  viên 2 | Nguyễn Lê Minh Khoa - 1050080055 | Đóng góp: Tìm kiếm các người dùng nổi tiếng trên twitter và đề xuất cấu trúc dữ liệu  Điểm tự đánh giá quá trình: 6  Điểm tự đánh giá cuối kì: 6 |
| Họ và Tên  thành  viên 3 | Lê Quốc Việt – 1050080083 | Đóng góp: tìm kiếm, cách hoạt động mô hình cho dựa án  Điểm tự đánh giá quá trình: 6  Điểm tự đánh giá cuối kì: 6 |
| Họ và Tên  thành  viên 4 | Cao Quốc Khánh - 1050080054 | Đóng góp: Tìm dữ liệu về các bài viết ,cài đặt code đào dữ liệu  Điểm tự đánh giá quá trình: 6  Điểm tự đánh giá cuối kì: 6 |

**BÁO CÁO TÓM TẮT**

• Lí do chọn đề tài

Một mô hình ứng dụng trong các bài toán sử dụng Big data hiện nay là công cụ phân tích tweets của người dung trên twitter (hiện tại là X). Đã từng có rất nhiều video hướng dẫn trích xuất dữ liệu nhưng bởi vì các chính sách mới của twitter làm nhiều nhà phân tích không còn lấy được data hoặc lượng dữ liệu thu thập được rất thấp.

Nên hiện tại nhóm chọn đề tài phân tích bài viết của người nổi tiếng trên twitter để tìm hiểu về cách truy cập dữ liệu mới để có thể dử dụng hiệu quả dữ liệu hơn.

• Tầm quan trọng

Đề tài phù hợp với nhu cầu xã hội, ứng dụng thực tế, hiện nay đang sử dụng

rộng rãi trên các ứng dụng tìm kiếm

Các thành viên trong nhóm có cơ hội đi sâu tìm hiểu về một ứng dụng thực tế

của lĩnh vực big data: cách thức setup, cài đặt một hệ thống cluster, các kỹ

thuật, công nghệ được sử dụng trong bài toán đề xuất, gợi ý.

• Các nội dung và kết quả

• Mô tả bài toán

• Giải pháp, mô hình

• Các công nghệ sử dụng

• Cài đặt Spark và Pycharm

• Demo

• Đánh giá

• Kết luận

**BÁO CÁO CHI TIẾT**

1. **Giới thiệu về bigdata.**
   1. định nghĩa.

Big data là sự kết hợp của các loại dữ dự liệu có cấu trúc, bán cấu trúc và phi cấu trúc được thu thập bỏi các tổ chức có thể khai thác thông tin và sử dụng chúng trong các dự án machine learning, predictive modeling (mô hình dự đoán) và các ứng dụng phân tích khác.



Ngày nay các hệ thống phân tích, xử lí và lưu trữ big data đã trở thành một phần phổ biến trong kiến trúc quản lý của các công ty, tổ chức lớn, kết hợp với các công cụ hỗ trợ sử dụng phân tích Big data. Big data thường được đặc trưng bởi 3 chữ V:

- Lượng dữ liệu lớn trong nhiều môi trường: Mặc dù Big data không tương đương bất kỳ dữ liệu cụ thể nào nhưng việc triển khai dữ liệu lớn thường liên quan đến hàng terabyte, petabyte hay thậm chí lên đến exabyte dữ liệu được tạo và thu thập theo thời gian.

- Sự đa dạng về sự thể hiện dữ liệu trong Big data với 3 kiểu dữ liệu chính:

o Dữ liệu có cấu trúc: Dữ liệu có cấu trúc được xem là dữ liệu đơn giản nhất để quản lý và tìm kiếm. Nó là những dữ liệu có thể truy cập, lưu trữ và xử lý ở định dạng cố định. Các thành phần của dữ liệu có cấu trúc được phân loại dễ dàng, cho phép các nhà thiết kế và quản trị viên cơ sở dữ liệu xác định các thuật toán đơn giản để tìm kiếm và phân tích.

o Dữ liệu phi cấu trúc: Dữ liệu phi cấu trúc là bất kỳ tập hợp dữ liệu nào không được tổ chức hoặc xác định rõ ràng. Loại dữ liệu này hỗn loạn, khó xử lý, khó hiểu và đánh giá. Nó không có cấu trúc cố định và có thể thay đổi vào những thời điểm khác nhau. Dữ liệu phi cấu trúc bao gồm các nhận xét, tweet, lượt chia sẻ, bài đăng trên mạng xã hội, video trên YouTube mà người dùng xem, ...

o Dữ liệu bán cấu trúc: Dữ liệu bán cấu trúc là sự kết hợp giữa dữ liệu có cấu trúc và dữ liệu phi cấu trúc. Email là một ví dụ điển hình vì chúng bao gồm dữ liệu phi cấu trúc trong nội dung thư, cũng như nhiều thuộc tính tổ chức khác như người gửi, người nhận, chủ đề và ngày tháng. Các thiết bị sử dụng gắn thẻ địa lý, thời gian cũng có thể cung cấp dữ liệu có cấu trúc bên cạnh nội dung phi cấu trúc.

- Tốc độ dữ liệu được tạo ra và 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.

1.2 Cách hoạt động.

- Tích Hợp:

Big data có thể tích hợp dữ liệu từ các nguồn, ứng dụng, công cụ khác nhau. Với cơ chế tích hợp dữ liệu truyền thống như trích xuất, biến đổi và tải (ETL) không còn phù hợp với nhiệm vụ mới. Do đó, cần có công nghệ và chiến lược mới để phân tích các tập cơ sở dữ liệu Big data ở quy mô khổng lồ, như terabyte, thậm chí là petabyte.

Trong quá trình tích hợp, cần đảm bảo dữ liệu được định dạng và luôn có sẵn để các nhà phân tích kinh doanh xử lý và đưa ra kết luận chính xác dựa trên nguồn dữ liệu đó.

- Quản lý:

Khối lượng dữ liệu lớn cũng đồng nghĩa cần một kho lưu trữ lớn, đó có thể là Cloud (Đám mây điện tử), lưu trữ tại chỗ, cũng có thể là cả hai. Cho dù quản lý ở hình thức nào, chỉ cần đưa ra yêu cầu xử lý thì các công cụ đó sẽ hỗ trợ một cách linh hoạt.

- Phân Tích:

Để có được kết quả phù hợp từ việc phân tích Big data, các nhà khoa học dữ liệu và phân tích dữ liệu phải hiểu chi tiết về các dữ liệu có sẵn cùng những điều mà họ đang tìm kiếm ở đó. Việc chuẩn bị dữ liệu như lập hồ sơ, xác nhận, chuyển đổi các tập dữ liệu là những bước đầu tiên trong quá trình phân tích.

1.3 Vai Trò

Với khả năng thu thập, lưu trữ và xử lý các dữ liệu lớn, phức tạp, Big data đóng vai trò quan trọng và thiết yếu trong các doanh nghiệp hiện nay, đặc biệt là trong chuyển đổi số.

Cải thiện trải nghiệm khách hàng

Big data giúp doanh nghiệp phân tích, thấu hiểu và nhắm đúng khách hàng mục tiêu. Hiểu được hành vi, xu hướng mua hàng của khách hàng, đồng thời dự đoán tình hình thị trường tương lai để đưa ra các quyết định kinh doanh chính xác hơn.

Big data cho phép các doanh nghiệp điều chỉnh sản phẩm phù hợp với thị trường mục tiêu của họ mà không phải chi nhiều tiền cho các chiến dịch quảng cáo không hiệu quả. Bằng cách theo dõi các giao dịch tại điểm bán hàng (POS) và mua hàng trực tuyến, các doanh nghiệp có thể sử dụng Big data để nghiên cứu khách hàng tiềm năng.

Sử dụng những nghiên cứu sâu sắc này, doanh nghiệp có thể đưa ra các chiến lược tiếp thị và mục tiêu đúng hướng, đáp ứng mong đợi của người tiêu dùng, thúc đẩy lòng trung thành với thương hiệu.



Nâng cao năng suất

Các công cụ Big data hiện đại giúp các nhà khoa học và nhà phân tích dữ liệu có thể kiểm tra hiệu quả lượng dữ liệu khổng lồ, giúp họ có cái nhìn tổng quan nhanh về dữ liệu.

Đồng thời, chúng cho phép các nhà khoa học và phân tích dữ liệu tìm hiểu thêm về hiệu quả của các đường dẫn dữ liệu, cho phép họ chọn cách hoàn thành nhiệm vụ một cách hiệu quả hơn, từ đó giúp tăng năng suất.

Tối ưu hoá hiệu suất cá nhân

Với Big data, giờ đây chúng ta có thể ứng dụng những dữ liệu được tạo từ các thiết bị tiện lợi như đồng hồ thông minh hoặc vòng đeo tay thông minh. Những dữ liệu từ người dùng đem lại cho doanh nghiệp cái nhìn rõ ràng hơn về xu hướng, kỳ vọng mua hàng, nhằm xây dựng một lộ trình và chiến lược cụ thể, đúng đắn.

Giảm thiểu rủi ro nhờ phát hiện gian lận

Big data được áp dụng trong việc cải thiện bảo mật và góp phần vào việc thực thi các quy định của pháp luật. Nhiều doanh nghiệp sử dụng các phân tích dữ liệu lớn để phát hiện và kịp thời ngăn chặn các giao dịch gian lận.

Trong môi trường kinh doanh cạnh tranh khốc liệt như hiện nay, doanh nghiệp cần áp dụng Big data để bảo mật quyền lợi thương hiệu, đồng thời ngăn chặn rủi ro từ tác động của môi trường bên ngoài.

Tối ưu hoá giá cả

Big data cũng tham gia vào quá trình định giá sản phẩm/ dịch vụ cho doanh nghiệp. Các doanh nghiệp có thể dựa trên các phân tích dữ liệu về giá cả cũng như giá của đối thủ cạnh tranh cùng phân khúc và xu hướng mua hàng của khách hàng.

Nhờ đó có thể giảm thiểu thời gian đi thị trường phân tích mà vẫn có được dữ liệu chất lượng. Big data được xem là một lợi ích giúp doanh nghiệp tối ưu hóa giá cả, gia tăng lợi nhuận cho doanh nghiệp về lâu dài.

Nắm bắt được các giao dịch tài chính

Vai trò tiếp theo của Big data là nắm bắt được các giao dịch tài chính. Giao dịch ở tần số cao (HFT) là nơi mà dữ liệu lớn được tìm thấy rất nhiều, tại đây, các thuật toán Big data được dùng để đưa ra quyết định giao dịch.

Rất nhiều các thanh toán và giao dịch điện tử diễn ra phổ biến trên thế giới, Việt Nam cũng không ngoại lệ, nhiều thương hiệu tập trung vào các giao dịch để phân tích dữ liệu của người dùng. Đặc biệt là các doanh nghiệp hoạt động trong lĩnh vực Thương mại điện tử, Big data sẽ có nhiều lợi ích to lớn cho các thương hiệu này.

Hỗ trợ đổi mới

Đổi mới kinh doanh dựa trên những hiểu biết mà các nhà phân tích khám phá thông qua Big data. Việc phát triển sản phẩm có thể được thực hiện bằng cách biết được người tiêu dùng nghĩ gì về sản phẩm/ dịch vụ của doanh nghiệp.

Các nhà quản lý phải đưa ra quy trình theo dõi, phản hồi, và các đối thủ trong thị trường hiện tại. Phân tích dữ liệu đồng thời cũng giúp giám sát thị trường theo thời gian thực, hỗ trợ đổi mới kịp thời để bắt kịp xu hướng.

Tăng cường sự thích nghi, sáng tạo

Việc phân tích Big data có thể hỗ trợ các doanh nghiệp trở nên sáng tạo và dễ thích nghi hơn trên thị trường. Nhờ phân tích, doanh nghiệp có thể hiểu rõ hơn về đối thủ cạnh tranh và xử lý nỗi đau của khách hàng một cách hiệu quả, linh hoạt hơn.

Ngoài ra, có sẵn nhiều dữ liệu cũng cho phép doanh nghiệp đánh giá rủi ro, nâng cao sản phẩm/ dịch vụ. Điều này cung cấp cho những người ra quyết định thông tin cần thiết để giúp doanh nghiệp đưa ra những quyết định chính xác và kịp thời.



Lĩnh vực ứng dụng tốt nhất Big data

1. Ngân hàng

Khi thu thập được một khối lượng lớn dữ liệu từ các nguồn, ngân hàng phải tìm ra những phương pháp mới để quản lý Big data. Bên cạnh việc hiểu và làm hài lòng khách hàng, các ngân hàng cũng phải giảm thiểu rủi ro và chống gian lận mà vẫn tuân thủ quy định của pháp luật.

Ứng dụng Big data vào hoạt động giúp các ngân hàng đưa ra quyết định quan trọng, chẳng hạn:

• Hệ thống phân tích có thể xác định các địa điểm xây dựng chi nhánh mới - nơi tập trung các khách hàng tiềm năng.

• Dự đoán số lượng tiền mặt cần thiết để cung ứng tại một địa điểm giao dịch cụ thể.

• Đặc biệt là các ngân hàng số, dữ liệu chính là xương sống của họ.

• Học máy, Ai được sử dụng để phát hiện các hành vi gian lận và kịp thời báo cáo cho các chuyên viên làm nhiệm vụ.

2. Giáo dục

Trong ngành giáo dục, việc xử lý các dữ liệu bảo mật, thông tin về học sinh, sinh viên, giảng viên, tài liệu,... đều phải được xử lý nhanh gọn. Do đó, Big data ứng dụng vào giáo dục mang lại nhiều chức năng hữu ích, bao gồm:

• Phân tích, lưu trữ, quản lý các bộ dữ liệu lớn bao gồm hồ sơ của sinh viên, học sinh.

• Sử dụng hệ thống quản lý Big data có khả năng trích xuất phân cấp để duy trì tính bảo mật.

• Cung cấp các dữ liệu cần thiết về các hoạt động trong lớp và giúp đưa ra quyết định cho giáo viên, người tham gia giảng dạy.

• Giúp các bộ đề kiểm tra không thể bị lộ.

Ví dụ: Trường đại học Kinh tế Quốc dân có gần 30000 sinh viên đa dạng các ngành đã triển khai Hệ thống Quản lý và Học tập, hệ thống này giúp theo dõi thời gian sử dụng các trang trong hệ thống, tổng kết quá trình và kết quả học tập của sinh viên.



3. Ngành bán lẻ

Trong ngành bán lẻ, một khối lượng dữ liệu lớn luôn cần được xử lý, những dữ liệu này được các doanh nghiệp bán lẻ thu thập để đưa ra chiến lược phát triển sản phẩm, các chiến dịch Marketing,...

Ứng dụng Big data trong ngành bán lẻ bao gồm những chức năng như sau:

• Giúp nhà quản lý xây dựng mô hình chi tiêu của mỗi khách hàng

• Với các phân tích dự đoán, có thể nắm bắt và so sánh được tỷ lệ cung - cầu, đồng thời tránh các sản phẩm không phù hợp với nhu cầu thị trường.

• Xác định được vị trí bày trí các sản phẩm lên kệ dựa vào thói quen và nhu cầu mua sắm của khách hàng

• Kết hợp phân tích các dữ liệu về thời điểm, truyền thông xã hội, giao dịch,... để xác định các sản phẩm phù hợp nhằm cung ứng cho khách hàng.

4. Y tế

Một hệ thống kém hiệu quả có thể kìm hãm những lợi ích chăm sóc sức khỏe tốt hơn. Điều này xảy ra khi dữ liệu điện tử không đủ hoặc không có sẵn. Big data sẽ đóng góp nhiều chức năng hữu ích trong ngành Y tế. Tại Việt Nam, ngành Y tế đang sử dụng các dữ liệu thu thập được từ ứng dụng điện thoại về những bệnh nhân bị Covid để quản lý và đưa ra những thông tin kịp thời.

Hoặc tại một số bệnh viện hiện nay sử dụng Big data để thu thập dữ liệu thông tin người khám bệnh, đặt lịch hẹn với bác sĩ thông qua ứng dụng điện thoại. Một số chức năng của Big data trong ngành Y tế bao gồm:

• Cho phép người quản lý ca biết được bác sĩ cụ thể vào những thời điểm khác nhau.

• Dựa vào hồ sơ sức khỏe điện tử để theo dõi tình trạng của bệnh nhân

• Đánh giá tình trạng bệnh thông qua các triệu chứng và xác định một số bệnh ở giai đoạn đầu.

• Sử dụng các thiết bị kỹ thuật số như vòng đeo tay thông minh, hệ thống Big data có thể dựa vào đó để theo dõi tình trạng bệnh và gửi báo cáo cho bác sĩ.

• Lưu trữ những hồ sơ nhạy cảm, có tính bảo mật cao một cách hiệu quả.

• Ứng dụng Big data cũng có thể báo các khu vực có nguy cơ bùng phát dịch như sốt rét, sốt xuất huyết, Covid-19,...



5. Truyền thông và giải trí

Các công ty truyền thông và giải trí cần đẩy mạnh chuyển đổi số để phân phối sản phẩm và nội dung nhanh nhất đến thị trường. Big data có nhiều ứng dụng hữu ích như:

• Xác định thiết bị và thời gian có hiệu quả nhất thông qua việc phân tích các dữ liệu

• Các công ty truyền thông, nghệ sĩ hoặc người phụ trách truyền thông có thể chọn địa điểm tần suất phân phối.

• Xem xét mức độ phổ biến, nghệ sĩ có thể chọn thiết bị, hệ điều hành để phân phối các sản phẩm, nội dung của mình.

6. Digital Marketing

Giờ đây, dường như bất kỳ doanh nghiệp nào cũng cần đến Digital Marketing để đưa sản phẩm/ dịch vụ của mình đến gần và rộng hơn với người tiêu dùng. Big data đã góp một phần lớn cho Digital Marketing phát triển mạnh mẽ như hiện tại, ví dụ:

• Phân tích thị trường và các đối thủ cạnh tranh. Đồng thời đánh giá mục tiêu kinh doanh của doanh nghiệp, xác định đâu là cơ hội, đâu là thách thức để tiếp tục tiến hành các kế hoạch kinh doanh khác.

• Xác định người dùng trên các phương tiện truyền thông xã hội, dựa vào nhân khẩu học như giới tính, tuổi tác, sở thích, thu nhập,... để xác định khách hàng mục tiêu.

• Xác định các chủ đề, nội dung được người dùng tìm kiếm nhiều để xây dựng chiến lược nội dung SEO trên công cụ tìm kiếm.

• Tạo các đối tượng tương tự thông qua việc sử dụng các cơ sở dữ liệu đối tượng hiện có để nhắm mục tiêu đến các khách hàng tương tự, từ đó kiếm được lợi nhuận.

7. Giao thông vận tải

Ứng dụng Big data vào ngành giao thông vận tải cũng mang lại nhiều chức năng như:

• Kiểm soát các tuyến đường xe buýt để cung cấp thông tin cho người dùng, bắt tuyến xe đúng điểm dừng và đúng thời gian.

• Các công ty vận tải tư nhân có thể ứng dụng Big data vào quản lý tài sản, kiểm tra và cải tiến công cụ, tối ưu hóa quy trình vận hành,...

• Ứng dụng Big data vào lập kế hoạch lộ trình, sắp xếp các phương tiện đi lại, di chuyển tới các điểm điểm khác nhau khi đi du lịch,...

8. Dịch vụ khách hàng

Ngoài chất lượng sản phẩm, dịch vụ tốt là điều mà khách hàng kỳ vọng nhiều nhất. Trong giai đoạn mới tham gia vào thị trường, Big data sẽ giúp chủ doanh nghiệp tìm ra những giải pháp và những đề xuất tối ưu để hiểu được khách hàng và tìm ra lợi thế cạnh tranh.

Cách thức ứng dụng Big Data vào dịch vụ khách hàng:

• Xác định yêu cầu khách hàng, tập trung thực hiện các nhu cầu và kỳ vọng để làm hài lòng họ.

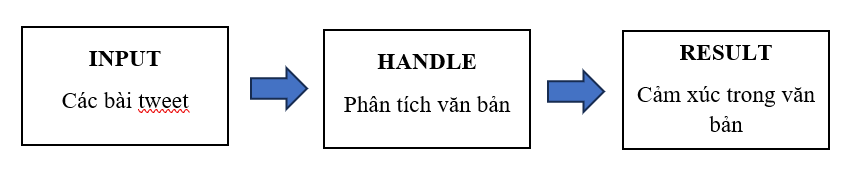
• Phân tích hành vi, sự quan tâm của khách hàng để tạo ra các sản phẩm/ dịch vụ khách hàng phù hợp, đồng thời thiết kế mô hình tiếp thị tối ưu.

• Nắm được sự tương đồng giữa khách hàng và nhu cầu, kỳ vọng của họ, từ đó có thể xây dựng các chiến dịch quảng cáo chính xác và mang lại hiệu quả cao.



1. **Bài toán**

Trên các bài tweet trên twitter của những người nổi tiếng trên mạng xã hội twitter và chúng ta quan tâm người đó và muốn phản hồi bằng cách bình luận cho bài tweet đó mà không biết cảm xúc trong bài tweet nó có cảm xúc tích cực hay tiêu cựa để dựa theo cảm xúc đó và phản hồi bình luận. Nên đây là giải pháp.



1. **Giải pháp**
   1. Apache Spark, Machine Learning LogisticRegression
2. Apache Spark.

- Khái niệm: Apache Spark là một hệ thống mã nguồn mở dùng để xử lý dữ liệu lớn và phân tích dữ liệu phân tán. Nó cung cấp một nền tảng mạnh mẽ cho việc tính toán song song trên các cluster máy tính và hỗ trợ xử lý và phân tích dữ liệu lớn với hiệu suất cao. Spark được phát triển bởi Apache Software Foundation và hỗ trợ nhiều ngôn ngữ lập trình như Scala, Java, Python và R

- Đặc điểm:

• Xử lý dữ liệu lớn: Apache Spark được thiết kế đặc biệt để xử lý dữ liệu lớn, với khả năng phân tích và tính toán trên nhiều node máy tính trong một cluster.

• Hỗ trợ nhiều ngôn ngữ: Spark hỗ trợ nhiều ngôn ngữ lập trình như Scala, Java, Python và R, giúp người phát triển lựa chọn ngôn ngữ ưa thích để phát triển ứng dụng.

• Xử lý dữ liệu trực tiếp trong bộ nhớ: Spark sử dụng cơ chế xử lý dữ liệu trực tiếp trong bộ nhớ (in-memory processing) để tăng hiệu suất, giúp giảm thời gian xử lý so với Hadoop MapReduce.

• Hỗ trợ nhiều công cụ và thư viện: Spark cung cấp một loạt các thư viện và công cụ cho xử lý dữ liệu, machine learning, đồ thị, xử lý dòng dữ liệu, và nhiều tác vụ khác.

• Tích hợp với hệ thống lưu trữ khác nhau: Spark có khả năng tích hợp với nhiều hệ thống lưu trữ dữ liệu khác nhau như HDFS, Apache HBase, Apache Cassandra, và nhiều loại cơ sở dữ liệu khác.

- Ưu và nhược điểm:

• Ưu điểm:

+ Hiệu suất cao: Apache Spark sử dụng xử lý dữ liệu trực tiếp trong bộ nhớ (in-memory processing), giúp tăng hiệu suất và giảm thời gian xử lý dữ liệu so với Hadoop MapReduce.

+ Xử lý dữ liệu lớn: Spark có khả năng xử lý dữ liệu lớn và phân tích dữ liệu phân tán trên các cluster máy tính.

+ Hỗ trợ nhiều ngôn ngữ: Spark hỗ trợ nhiều ngôn ngữ lập trình, cho phép các nhà phát triển sử dụng ngôn ngữ mà họ thoải mái.

+ Các thư viện và công cụ phong phú: Spark cung cấp một loạt các thư viện và công cụ cho xử lý dữ liệu, machine learning, đồ thị, xử lý dòng dữ liệu, và nhiều tác vụ khác.

+ Tích hợp với nhiều hệ thống lưu trữ: Spark có khả năng tích hợp với nhiều hệ thống lưu trữ khác nhau, giúp quản lý và truy cập dữ liệu dễ dàng.

• Nhược điểm:

+ Yêu cầu tài nguyên lớn: Spark yêu cầu cơ sở hạ tầng phần cứng mạnh mẽ để hoạt động hiệu quả, điều này có thể tạo ra một khả năng đầu tư ban đầu cao.

+ Khó khăn trong việc triển khai: Triển khai và cấu hình một hệ thống Spark có thể phức tạp và đòi hỏi kiến thức kỹ thuật sâu rộng.

+ Khó khăn trong việc gỡ lỗi: Gỡ lỗi trong môi trường Spark có thể khó khăn hơn so với môi trường xử lý dữ liệu truyền thống.

+ Học phức tạp: Học cách sử dụng Spark và tirne thực hiện các tác vụ phân tán có thể đòi hỏi thời gian và nỗ lực đáng kể.

* 1. Machine Learning LogisticRegression.

- Khái niệm: Mô hình Logistic Regression (Hồi quy Logistic) là một phương pháp thuộc lĩnh vực Machine Learning, thường được sử dụng cho các bài toán phân loại. Điều đặc biệt của Logistic Regression là nó được sử dụng để dự đoán xác suất xảy ra của một sự kiện dựa trên các biến đầu vào. Logistic Regression không giống như Linear Regression - mô hình dùng cho việc dự đoán giá trị liên tục. Thay vào đó, nó được áp dụng cho các tác vụ phân loại, nơi mục tiêu là dự đoán xác suất xảy ra của một sự kiện hoặc phân loại dữ liệu thành các nhóm khác nhau.

- Đặc điểm:

* Phù hợp cho dữ liệu có tính chất phân loại.
* Dễ hiểu và dễ triển khai.
* Có thể xử lý dữ liệu không đồng nhất và dữ liệu nhiễu.

Ưu điểm của logistic regression:

* Dễ hiểu và triển khai
* Tính toán nhanh chóng
* Có thể xử lý cả dữ liệu có tính tuyến tính và phi tuyến tính

Nhược điểm của logistic regression:

* Không thể xử lý các mô hình phức tạp
* Dễ bị ảnh hưởng bởi nhiễu và outlier trong dữ liệu
* Không hiệu quả khi có quá nhiều biến đầu vào
  1. Builtfullsoup python.

- Khái niệm BS4:Beautiful Soup (bs4) là một thư viện Python được sử dụng để phân tích và trích xuất thông tin từ các tài liệu HTML và XML. Thư viện này cung cấp một cách tiện lợi để trích xuất dữ liệu từ trang web và tài liệu có cấu trúc. Beautiful Soup giúp bạn tìm kiếm, lọc, và trích xuất thông tin từ mã nguồn HTML hoặc XML của một trang web dễ dàng hơn.



- Đặt Điểm:

• Phân tích HTML và XML: Beautiful Soup giúp bạn phân tích dữ liệu HTML hoặc XML và biến chúng thành cây phân cấp để bạn có thể truy cập các phần tử và thuộc tính dễ dàng.

• Tìm kiếm và lọc dữ liệu: Bạn có thể tìm kiếm các phần tử dựa trên tên, thuộc tính, hoặc nội dung. Nó cung cấp nhiều phương thức để tìm và lọc dữ liệu một cách linh hoạt.

• Trích xuất thông tin: Beautiful Soup cho phép bạn trích xuất văn bản, liên kết, hình ảnh, và các thành phần khác từ trang web dựa trên các yêu cầu tùy chỉnh.

• Xử lý dữ liệu không hoàn chỉnh: Thậm chí khi tài liệu HTML không hoàn chỉnh hoặc chứa lỗi, Beautiful Soup cũng có khả năng phân tích và trích xuất thông tin từ nó một cách linh hoạt.

• Hỗ trợ các trình phân tích cú pháp khác nhau: Beautiful Soup có khả năng làm việc với các trình phân tích cú pháp khác nhau như lxml, html5lib, và built-in parser.

- Ưu điểm

• Nhanh chóng và hiệu quả: Thư viện bs4 sử dụng thư viện lxml để phân tích cú pháp HTML và XML. Lxml là một thư viện nhanh và hiệu quả, giúp thư viện bs4 cũng nhanh và hiệu quả.

• Dễ sử dụng: Thư viện bs4 cung cấp các phương thức và hàm đơn giản và dễ sử dụng. Điều này giúp cho việc sử dụng thư viện bs4 trở nên dễ dàng hơn đối với các lập trình viên.

• Mạnh mẽ: Thư viện bs4 cung cấp một bộ tính năng rộng lớn để truy cập và thao tác với dữ liệu HTML và XML. Điều này giúp cho việc phân tích và trích xuất dữ liệu từ các trang web trở nên dễ dàng hơn.

- Nhược điểm

• Yêu cầu kiến thức về HTML và XML: Để sử dụng thư viện bs4 hiệu quả, lập trình viên cần có kiến thức cơ bản về HTML và XML.

• Không hỗ trợ tất cả các tính năng HTML: Thư viện bs4 không hỗ trợ tất cả các tính năng HTML. Điều này có thể gây ra một số vấn đề khi phân tích các trang web sử dụng các tính năng HTML không được hỗ trợ.

- Ứng dụng thư viện bs4 được sử dụng trong nhiều ứng dụng khác nhau, bao gồm:

• Crawler web: Thư viện bs4 được sử dụng để tạo các crawler web để thu thập dữ liệu từ các trang web.

• Xử lý ngôn ngữ tự nhiên: Thư viện bs4 được sử dụng để xử lý ngôn ngữ tự nhiên từ các trang web.

Tự động hóa: Thư viện bs4 được sử dụng để tự động hóa các tác vụ trên các trang web.

* 1. Selenium.

- Khái niệm :Selenium là một bộ công cụ tự động hóa được sử dụng để kiểm soát và tương tác với các trình duyệt web như Google Chrome, Mozilla Firefox, Microsoft Edge và nhiều trình duyệt khác. Nó cung cấp một API (Application Programming Interface) cho Python (cũng như nhiều ngôn ngữ lập trình khác) để tự động hóa các hoạt động trình duyệt web, chẳng hạn như điều hướng, điền biểu mẫu, bấm nút, và thu thập dữ liệu từ trang web.



- Đặc điểm:

• Tự động hóa trình duyệt web: Selenium cho phép bạn mở, điều hướng và tương tác với trình duyệt web như Google Chrome, Mozilla Firefox, Microsoft Edge và nhiều trình duyệt khác. Bạn có thể thực hiện các hoạt động như mở trình duyệt, mở URL, chạy mã JavaScript, và thao tác với các yếu tố trên trang web.

• Tự động kiểm thử phần mềm (Selenium WebDriver): Selenium WebDriver là một phần của Selenium Suite, được sử dụng rộng rãi trong kiểm thử phần mềm. Bạn có thể viết các kịch bản kiểm thử tự động để đảm bảo rằng ứng dụng web của bạn hoạt động đúng cách. Selenium WebDriver giúp kiểm tra tích hợp, xác minh giao diện và thực hiện kiểm thử chức năng tự động.

• Web scraping: Selenium cũng có thể được sử dụng cho việc trích xuất dữ liệu từ trang web, một quá trình thường được gọi là "web scraping". Bạn có thể tự động thăm các trang web, thu thập dữ liệu và lưu trữ chúng cho mục đích phân tích hoặc tự động hóa công việc. Điều này đặc biệt hữu ích khi bạn cần thu thập thông tin từ nhiều trang web hoặc khi dữ liệu không có sẵn qua API hoặc các phương tiện khác.

• Hỗ trợ nhiều trình duyệt: Selenium hỗ trợ nhiều trình duyệt phổ biến như Google Chrome, Mozilla Firefox, Microsoft Edge, Safari và nhiều trình duyệt khác. Điều này cho phép bạn tự động hóa các hoạt động trên nhiều trình duyệt và đảm bảo tính đa nền tảng của ứng dụng web của bạn.

• Hỗ trợ nhiều ngôn ngữ lập trình: Selenium hỗ trợ nhiều ngôn ngữ lập trình, bao gồm Python, Java, C#, Ruby, JavaScript, và nhiều ngôn ngữ khác. Điều này giúp cho các lập trình viên có thể lựa chọn ngôn ngữ mà họ ưa thích để sử dụng Selenium.

- Ưu điểm

• Mở nguồn: Selenium là mã nguồn mở, vì vậy nó miễn phí để sử dụng và phát triển.

• Cộng đồng lớn: Selenium có một cộng đồng lớn, với nhiều tài nguyên và hỗ trợ có sẵn.

• Tương thích với nhiều trình duyệt: Selenium có thể được sử dụng với nhiều trình duyệt web phổ biến, bao gồm Chrome, Firefox, Edge, v.v.

• Có thể mở rộng: Selenium có thể được mở rộng bằng cách sử dụng các plugin và tiện ích bổ sung.

- Nhược điểm

• Có thể phức tạp: Selenium có thể phức tạp để học và sử dụng.

• Yêu cầu cài đặt trình duyệt: Selenium yêu cầu cài đặt trình duyệt web trên máy tính của bạn.

• Có thể chậm: Selenium có thể chậm hơn các công cụ tự động hóa kiểm tra phần mềm khác.

- Ứng dụng

• Kiểm tra tự động: Selenium có thể được sử dụng để viết các kịch bản kiểm tra tự động cho các ứng dụng web.

• Tự động hóa: Selenium có thể được sử dụng để tự động hóa các tác vụ trên web, chẳng hạn như đăng nhập vào trang web, đặt hàng trực tuyến, v.v.

• Crawling: Selenium có thể được sử dụng để crawl dữ liệu từ các trang web.

* 1. Hadoop hdfs

- Khái niệm: HDFS (tên viết tắt của từ Hadoop Distributed File System” là một hệ thống lưu dữ dữ dữ liệu được sử dụng bởi Hadoop. Chức năng của hệ thống này là cung cấp khả năng truy cập với hiệu suất cao đến với các dữ liệu nằm trên các cụm của Hadoop.



- Ưu điểm:

• HDFS cho phép dữ liệu có thể phân tán: Điều này có thể hiểu như sau: Nếu như có một cụm Hadoop mà trong đó bao gồm 20 máy tính thì bạn chỉ cần đưa một file dữ liệu vào HDFS. Khi đó, thì file sẽ tự động được chia nhỏ thành nhiều phần rồi được lưu trữ ở 20 máy tính đó.

• HDFS cho phép tính toán và phân tán song song: Thay vì chỉ sử dụng một máy để xử lý công việc, thì với HDFS thì bạn có thể để các máy hoạt động song song để xử lý chung một công việc để tiết kiệm thời gian.

• HDFS cho phép nhân bản các file: Đặc điểm này sẽ giúp bạn đề phòng được các trường hợp một máy tính trong cụm Hadoop phát sinh sự cố thì dữ liệu sẽ được backup lại mà không bị mất.

• HDFS có thể mở rộng theo chiều dọc: Lúc này, bạn sẽ có nhiệm vụ nâng cấp cho các hệ thống bằng cách tăng cấu hình cho máy tính lên. Tính năng này còn được gọi là Scale Up hay Vertical scaling.

- Kiến trúc:



• NameNode có tác vụ chính là đóng, mở và đổi tên cho các tập tin, thư mục. Ngoài ra, nó còn điều chỉnh cho các truy cập đến hệ thống tập tin. Trong khi đó, DataNode có tác vụ là ghi, đọc vào hệ thống tập tin; nó còn có nhiệm vụ tạo, xóa, nhân rộng các dữ liệu dựa trên chỉ dẫn của NameNode. Lưu ý: NameNode và DataNode đều là phần mềm được thiết kế nhằm mục đích chạy trên máy chủ và chúng được viết bằng Java. Quá trình hoạt động của NameNode và DataNode như sau:

• NameNode: Có trách nhiệm điều phối cho các thao tác truy cập của client với hệ thống HDFS. Bởi vì các DataNode là nơi lưu trữ thật sự các block của các file trên HDFS nên chúng là nơi đáp ứng các truy cập này. NameNode sẽ thực hiện nhiệm vụ của nó thông qua daemon tên namemode chạy trên port 8021.

DataNode: DataNode server sẽ chạy một daemon datanode trên port 8022, theo định kỳ thì mỗi DataNode sẽ có nhiệm vụ báo cáo cho Namenode biết được danh sách tất cả các block mà nó đang lưu trữ. Để NameNode có thể dựa vào nó để cập nhật lại các metadata trong nó.

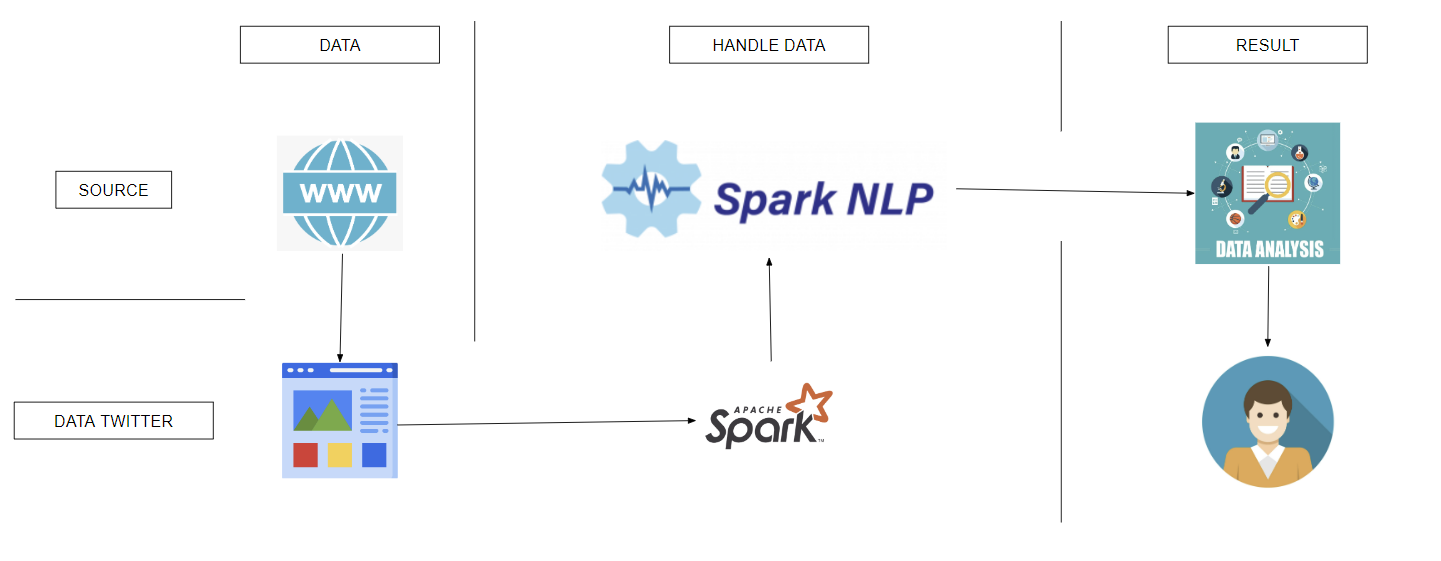
Sau mỗi lần cập nhật thì metadata trên NameNode đều sẽ đạt được các tình trạng thống nhất dữ liệu trên các DataNode. Toàn bộ trạng thái của metadata trên NameNode sẽ đạt được sự thống nhất với các dữ liệu ở trên DataNode. Tất cả các trạng thái của metadata ngay khi đang ở tình trạng hệ thống này sẽ được gọi là checkpoint.

Mỗi một metadata ở trạng thái checkpoint đều sẽ được sử dụng cho mục đích nhân bản metadata với mục đích phục hồi lại NameNode nếu như NameNode xuất hiện lỗi. Đọc file trên HDFS: Khi các máy con client gửi yêu cầu đọc đến NameNode, khi đó NameNode nhận được phải sẽ thực hiện các tác vụ để kiểm tra xem file có tồn tại không, file có bị lỗi hoặc bị nhiễm virus không?

Nếu như file không có vấn đề thì NameNode sẽ gửi các danh sách của các Block của file cùng với địa chỉ của các DataNode. Sau đó, hệ thống sẽ mở kết nối với DataNode rồi thực hiện chức năng RPC để nhận được các dữ liệu cần đọc, rồi đóng kết nối với DataNode còn lại.

Khi đó, các client đọc các block của file liên tục và lặp lại cho đến block cuối của file. Tiếp theo, lập trình viên sẽ sử dụng một tập tin API của Hadoop để có thể tương tác trực tiếp được với HDFS. Những tập API sẽ có chức năng giấu đi các NameNode để giúp kết nối với các DataNode để nhận được dữ liệu.

* 1. Mô hình giải pháp.

- Lấy dữ liệu theo thời gian thực trên trang web nitter

- Lấy thành công lưu dữ liệu dưới dạng file csv

- Đưa file csv vào spark để sử lý dử liệu

- Dùng mô hình Machine Learning LogisticRegression để phân tích dữ liệu

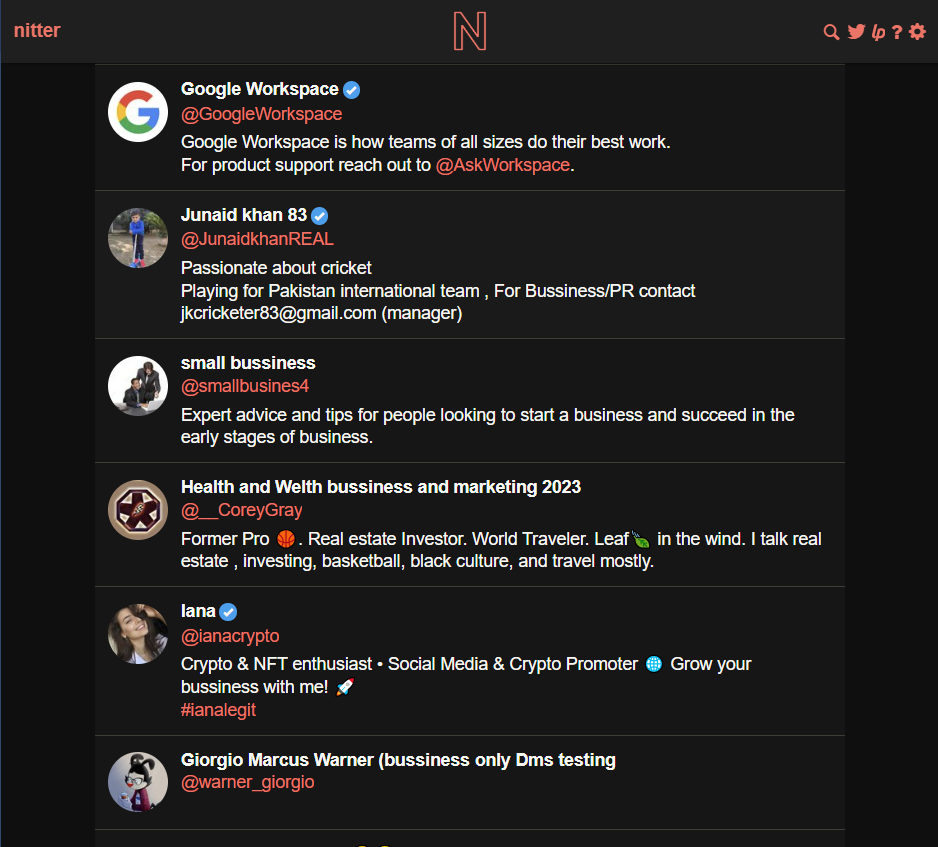
- Đưa dữ liệu đã phân tích cho người dùng

3.3 Cài đặt thử nghiệm:

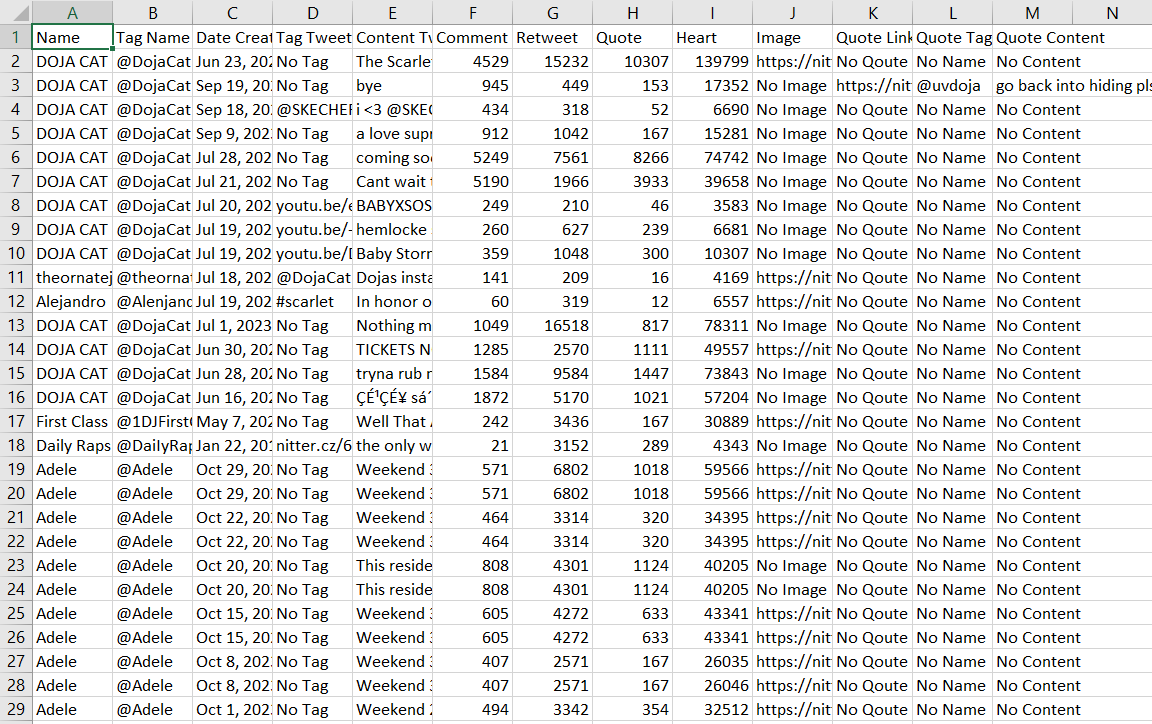
DATASET:

• Thu nhập dữ liệu:

Tất cả dữ liệu được lấy từ trang web nitter.net và netter.cz. hệ thống crawler được thực hiện qua mỗi lần chạy ứng dụng, dữ liệu được lấy trực tiếp từ web ( sử dụng ngôn ngữ Python và thư viện bs4 và selenium để tương tác với web để lấy dữ liệu) và lưu trử trên máy cục bộ.



Nitter là một trang web chia sẻ tweet mã nguồn mở, được tạo ra như một giải pháp thay thế cho Twitter. Trang web này cung cấp một giao diện người dùng đơn giản và dễ sử dụng, cho phép người dùng xem tweet, theo dõi người dùng và tìm kiếm tweet theo thẻ.



Bảng trên đây là dữ liệu thu nhập được:

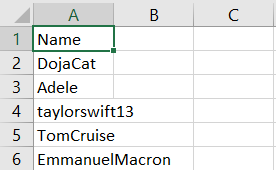
MÔ TẢ FILE

Thông tin các tweet được lưu trong tệp data\_tweet.csv và có định dạng sau:

* Name: nơi lưu trử thông tin tên người dùng đăng tweet.
* Tag Name: lưu trử các thẻ người dùng
* Date Create Tweet: ngày lưu tweet được đăng tải
* Tag Tweet: nơi bài tweet có gắn thẻ người dùng khác
* Content Tweet: nơi lưu trữ nội dung tweet
* Comment, Retweet, Quote, Heart: nơi lưu trữ các Stats của bài tweet
* Image: nơi lưu trữ ảnh của bài tweet
* Quote Link, Quote Tag, Quote Content: vì đa số các bài tweet đều có trích dẩn nên có thêm các đường liên kết, thẻ, nội dung của các trích dẩn này.

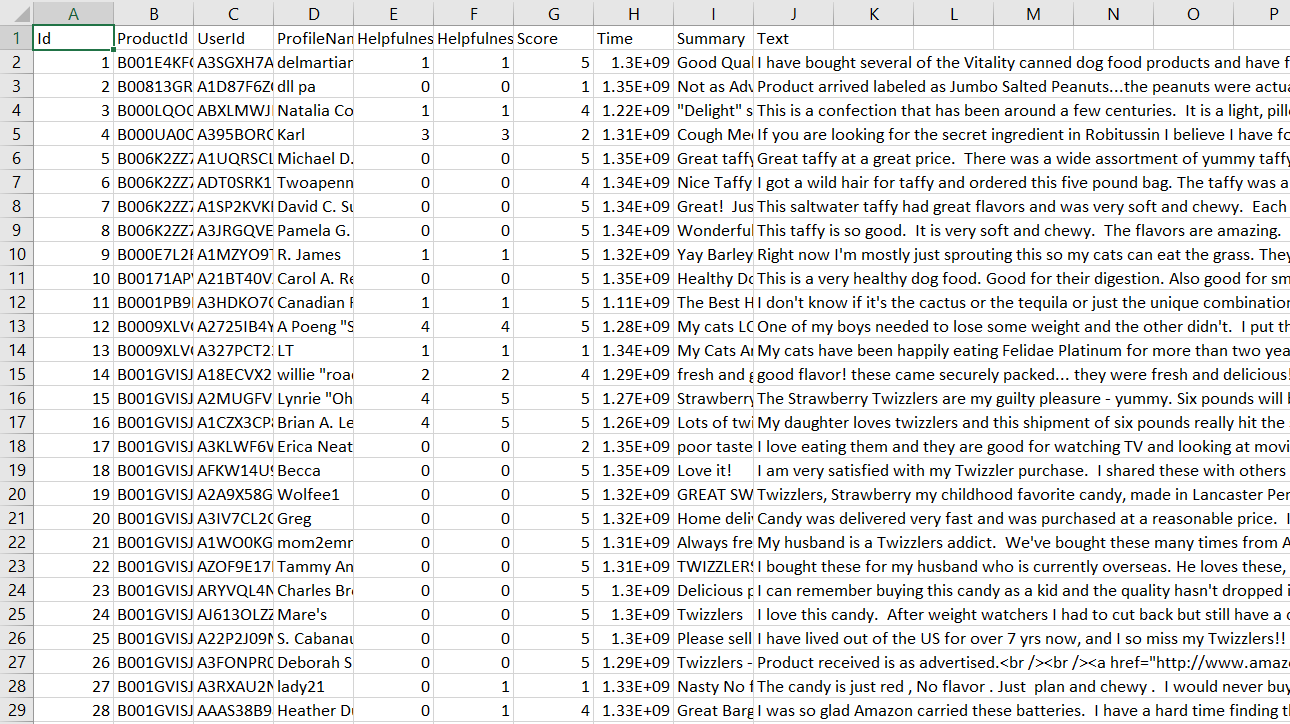
MÔ TẢ FILE

Thông tin tên các người nổi tiếng được lưu trong file User.csv có định dạng sau:



- Name: chứa các thẻ tên người dùng

MÔ TẢ FILE

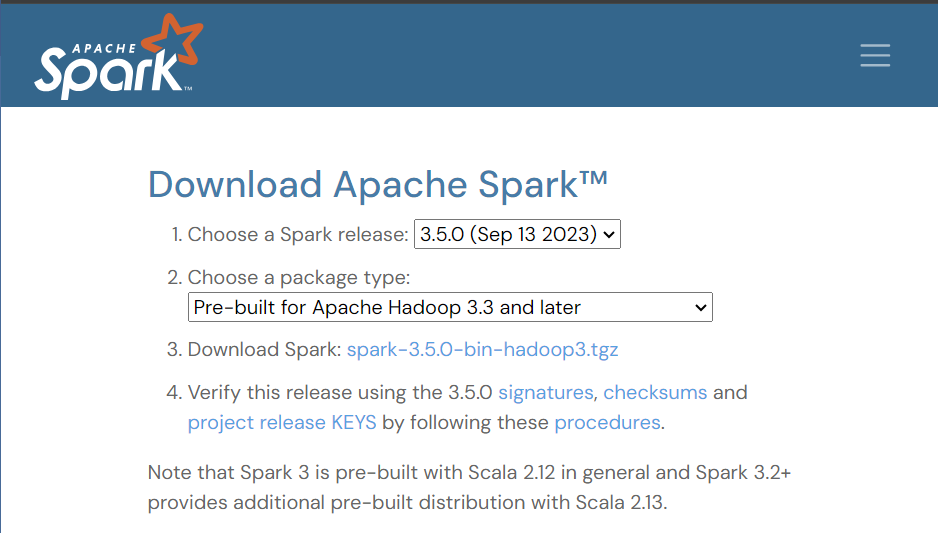


- là file lưu trữ các đánh giá sản phẩm trên twitter

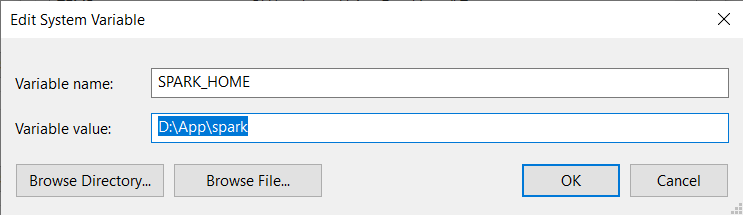
THỬ NGHIỆM

* Cài đặt Spark (Trên Windows)

Bước 1: tải gói Spark tại trang web <https://spark.apache.org/downloads.html>



Bước 2: cài Path cho Spark:



Bước 3: Khởi động Spark

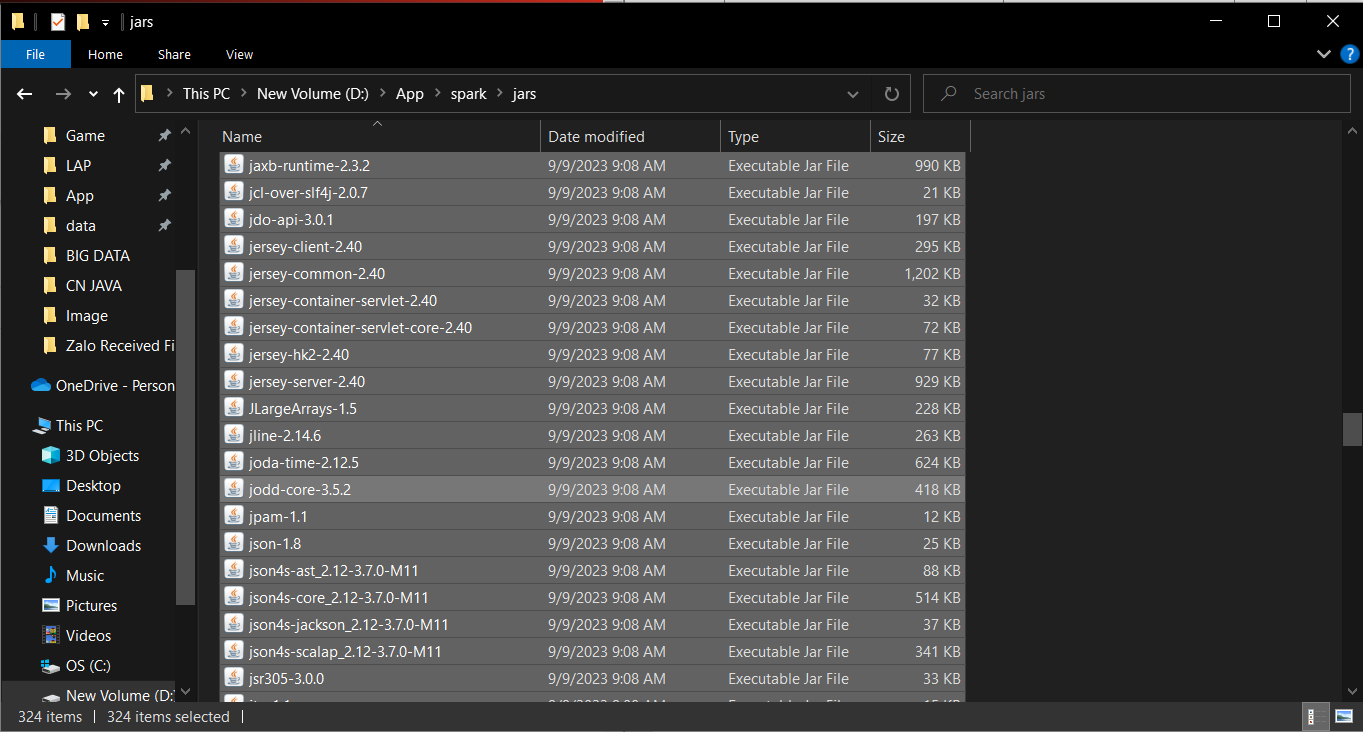


* Cài đặt mô hình Machine Learning LogisticRegression:

Bước 1: tải các file jar giành cho spark nlp với đường link sau <https://jar-download.com/?search_box=spark%20nlp>



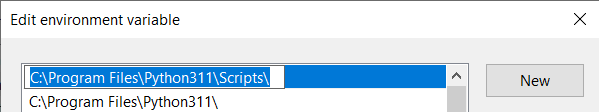
Bước 2: sao chép các tệp jar này vào tệp jars của spark

* Sử dụng IDE Pycharm để thực hiện ứng dụng:

Bước 1: cài đặt Python với đường link sau <https://www.python.org/downloads/>



Bước 2: set Path cho Python



Bước 3: cài đặt Pycharm với đường link sau <https://www.jetbrains.com/pycharm/download/?section=windows>

Bước 4: tạo dự án và cài đặt 1 số thư viện trong Terminal Pycharn Project với từng câu lệnh sau:

- Crawl-data.py

class post\_tweet:

def \_\_init\_\_(self, Name, Tag\_Name, Date, Tag\_Tweet, Content\_tweet, Comment, Retweet, Quote, Heart, Picture,

Quote\_Link, Quote\_Name, Quote\_Content):

self.Name = Name

self.Tag\_Name = Tag\_Name

self.Date = Date

self.Tag\_Tweet = Tag\_Tweet

self.Content\_Tweet = Content\_tweet

self.Comment = Comment

self.Retweet = Retweet

self.Quote = Quote

self.Heart = Heart

self.Picture = Picture

self.Quote\_Link = Quote\_Link

self.Quote\_Name = Quote\_Name

self.Quote\_Content = Quote\_Content

def remove\_emoji(string):

emoji\_pattern = re.compile("["

u"\U0001F600-\U0001F64F" # emoticons

u"\U0001F300-\U0001F5FF" # symbols & pictographs

u"\U0001F680-\U0001F6FF" # transport & map symbols

u"\U0001F1E0-\U0001F1FF" # flags (iOS)

u"\U00002500-\U00002BEF" # chinese char

u"\U00002702-\U000027B0"

u"\U00002702-\U000027B0"

u"\U000024C2-\U0001F251"

u"\U0001f926-\U0001f937"

u"\U00010000-\U0010ffff"

u"\u2640-\u2642"

u"\u2600-\u2B55"

u"\u200d"

u"\u23cf"

u"\u23e9"

u"\u231a"

u"\ufe0f" # dingbats

u"\u3030"

"]+", flags=re.UNICODE)

return emoji\_pattern.sub(r'', string)

def clean\_data(Name, Tag\_Name, Date, Tag\_Tweet, Content\_tweet,

Comment, Retweet, Quote, Heart, Picture,

Quote\_links, Quote\_name, Quote\_Content):

Name = remove\_emoji(Name)

Content\_tweet = re.sub(r"[!?@$%^&~`\*.,’'()\_+=:;-]", "", Content\_tweet.replace("\n", ""))

Quote\_Content = re.sub(r"[!?@$%^&~`\*.,’()\_+=:;-]", "", Quote\_Content.replace("\n", ""))

Comment = Comment.replace(" ", "").replace(",", "")

Retweet = Retweet.replace(" ", "").replace(",", "")

Quote = Quote.replace(" ", "").replace(",", "")

Heart = Heart.replace(" ", "").replace(",", "")

data = post\_tweet(Name, Tag\_Name, Date, Tag\_Tweet, Content\_tweet,

Comment, Retweet, Quote, Heart, Picture,

Quote\_links, Quote\_name, Quote\_Content)

print(data.Name + " " + data.Tag\_Name + " " + data.Date + " " + data.Tag\_Tweet + " " + remove\_emoji(

Content\_tweet) + " " + data.Comment + " " + data.Retweet + " " + data.Quote + " " + data.Heart + " " + data.Picture + " " + data.Quote\_Link + " " + data.Quote\_Name + " " + remove\_emoji(

Quote\_Content))

print("Success Save Data!")

save\_data(data)

def save\_list\_tag\_tweet(list\_tweet):

tag\_tweet = ""

for list in list\_tweet:

tag\_tweet = tag\_tweet + " " + list.text

return tag\_tweet

def crawler\_data(posts, User\_Name):

for post in posts:

dr.get(url + post.get("href"))

page\_post = dr.find\_element(By.TAG\_NAME, "html")

soup\_link = BeautifulSoup(page\_post.get\_attribute("innerHTML"), "html.parser")

if soup\_link.find("a", {"class", "fullname"}) is not None:

Name = soup\_link.find("a", {"class", "fullname"}).text

Tag\_Name = soup\_link.find("a", {"class", "username"}).text

Date = soup\_link.find("span", {"class", "tweet-date"}).find("a").get("title")

if soup\_link.find("div", {"class", "tweet-content media-body"}).find("a") is None:

Tag\_Tweet = "No Tag"

else:

Tag\_Tweet = soup\_link.find("div", {"class", "tweet-content media-body"}).find\_all("a")

Tag\_Tweet = save\_list\_tag\_tweet(Tag\_Tweet)

Content\_tweet = soup\_link.find("div", {"class", "tweet-content media-body"}).text

Tweet\_State = soup\_link.find("div", {"class", "tweet-stats"}).find\_all("div", {"class", "icon-container"})

Image = soup\_link.find("div", {"class", "tweet-body"}).find("div", {"class", "attachments"})

if Image is None:

Picture = "No Image"

elif Image.find("a", {"class", "still-image"}) is None:

Picture = "No Image"

else:

Image = Image.find("a", {"class", "still-image"}).get("href")

Picture = url + str(Image)

Qoute = soup\_link.find("div", {"class", "tweet-body"}).find("div", {"class", "quote quote-big"})

if Qoute is None:

Quote\_links = "No Qoute"

Quote\_name = "No Name"

Quote\_Content = "No Content"

else:

Qoute\_link = Qoute.find("a").get("href")

Quote\_links = url + Qoute\_link

Quote\_name = Qoute.find("a", {"class", "username"}).text

if Qoute.find("div", {"class", "quote-text"}).text == "":

Quote\_Content = "No Content"

else:

Quote\_Content = Qoute.find("div", {"class", "quote-text"}).text

dr.get(url + "/" + "".join(User\_Name))

clean\_data(Name, Tag\_Name, str(Date), Tag\_Tweet, Content\_tweet,

Tweet\_State[0].text, Tweet\_State[1].text, Tweet\_State[2].text, Tweet\_State[3].text, Picture,

Quote\_links, Quote\_name, Quote\_Content)

else: dr.get(url + "/" + "".join(User\_Name))

def save\_data(Data):

table = mydb["Posts"]

data\_form = {

"Name": Data.Name,

"Tag Name": Data.Tag\_Name,

"Date Create Tweet": Data.Date,

"Tag Tweet": Data.Tag\_Tweet,

"Content Tweet": Data.Content\_Tweet,

"Comment": Data.Comment,

"Retweet": Data.Retweet,

"Quote": Data.Quote,

"Heart": Data.Heart,

"Image": Data.Picture,

"Quote Link": Data.Quote\_Link,

"Quote Tag Name": Data.Quote\_Name,

"Quote Content": Data.Quote\_Content

}

table.insert\_one(data\_form)

with open("D:/data/data\_tweet.csv", "a", encoding="utf-8",

newline="") as files:

writer\_file = csv.writer(files)

writer\_file.writerow(

[Data.Name, Data.Tag\_Name, Data.Date, Data.Tag\_Tweet, Data.Content\_Tweet, Data.Comment, Data.Retweet,

Data.Quote, Data.Heart, Data.Picture, Data.Quote\_Link, Data.Quote\_Name, Data.Quote\_Content])

def select\_user():

with open("D:/data/data\_tweet.csv", "w", encoding="utf-8",

newline="") as file:

writer = csv.writer(file)

writer.writerow(

["Name", "Tag Name", "Date Create Tweet", "Tag Tweet", "Content Tweet", "Comment", "Retweet",

"Quote", "Heart", "Image", "Quote Link", "Quote Tag Name", "Quote Content"])

with open("D:/data/User.csv", "r") as file:

read = pandas.read\_csv(file, header=0)

for i in read.iterrows():

url\_user = url + "/" + "".join(i[1].values)

print("Select User: " + url\_user)

dr.get(url\_user)

get\_page = dr.find\_element(By.TAG\_NAME, "html")

soup = BeautifulSoup(get\_page.get\_attribute("innerHTML"), "html.parser")

posts = soup.find\_all("a", {"class", "tweet-link"})

for m in range(5):

try:

find = dr.find\_element(By.LINK\_TEXT, "Load more").text

except:

find = ""

if find != "":

dr.find\_element(By.LINK\_TEXT, "Load more").click()

get\_page = dr.find\_element(By.TAG\_NAME, "html")

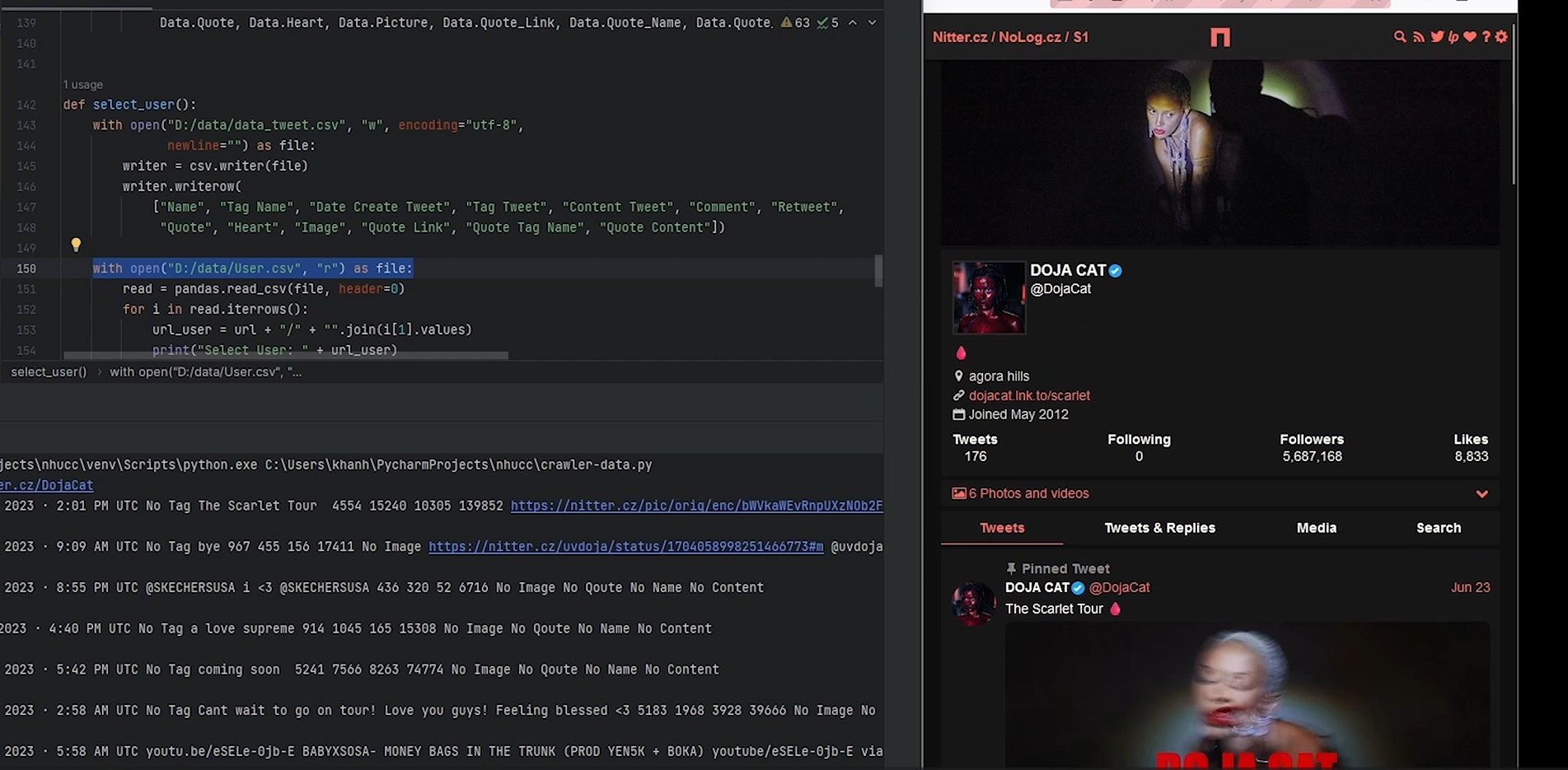
soup = BeautifulSoup(get\_page.get\_attribute("innerHTML"), "html.parser")

posts\_more = soup.find\_all("a", {"class", "tweet-link"})

posts.extend(posts\_more)

crawler\_data(posts, i[1].values)

print("Complete!")



url = "https://nitter.cz"

headers = {

'authority': 'www.google.com',

'accept': 'text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,\*/\*;q=0.8,'

'application/signed-exchange;v=b3;q=0.7',

'accept-language': 'en-US,en;q=0.9',

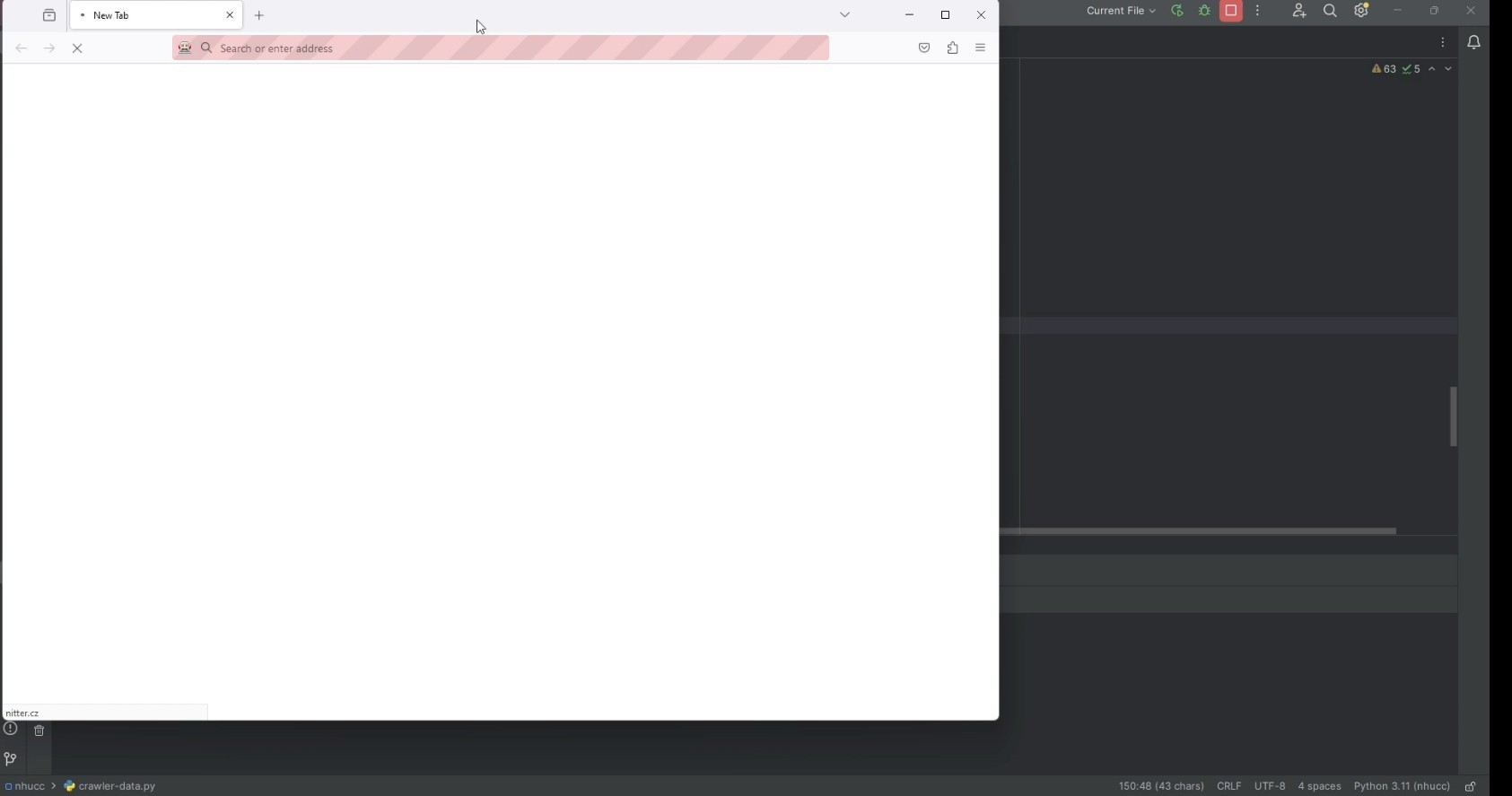
'cache-control': 'max-age=0',

'user-agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/115.0.0.0 '

'Safari/537.36',

}

dr = webdriver.Firefox()



wait = WebDriverWait(dr, 5)

dr.implicitly\_wait(5)

dr.get(url)

no\_bot = dr.find\_element(By.TAG\_NAME, "input")

no\_bot.click()

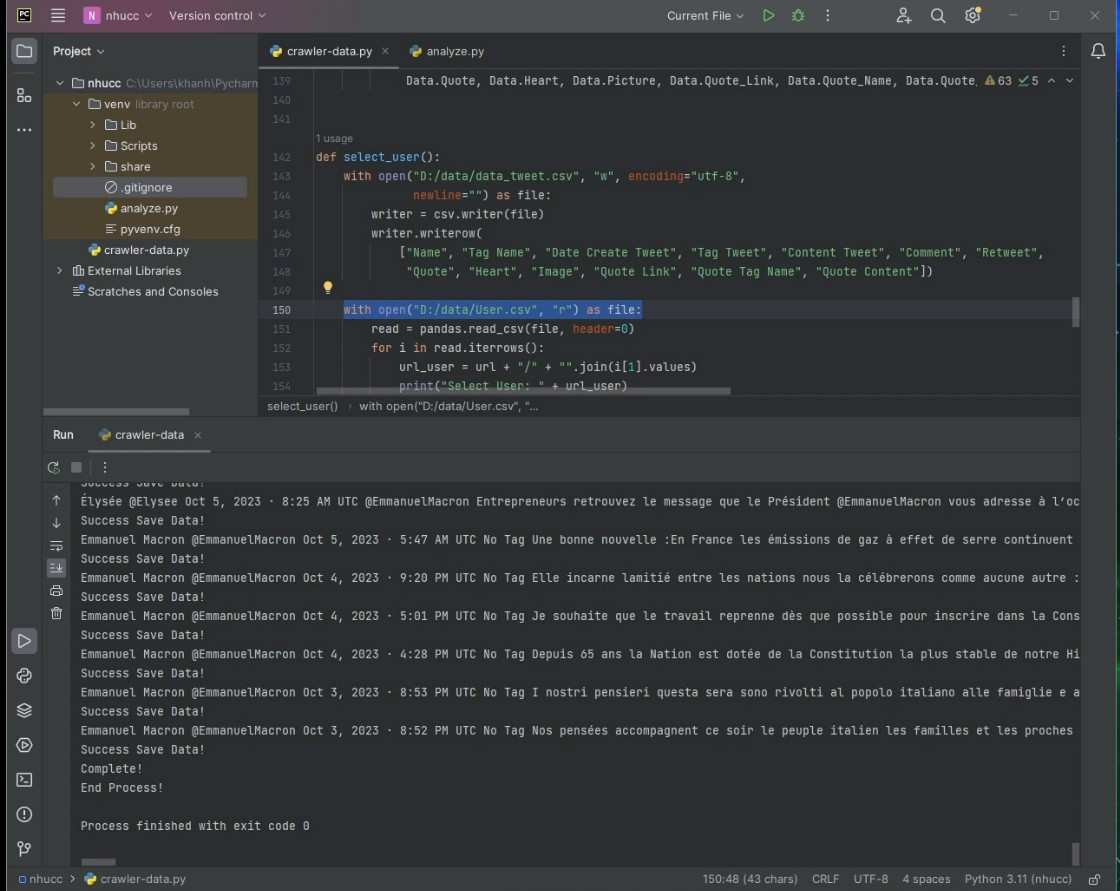
myclient = pymongo.MongoClient("mongodb://localhost:27017/")

mydb = myclient["data\_tweet"]

select\_user()

print("End Process!")

dr.quit()



- Analysis-data.py

import csv

from pyspark.ml.evaluation import MulticlassClassificationEvaluator

from pyspark.ml.classification import LogisticRegression

from pyspark.ml.feature import HashingTF

from pyspark.sql import SparkSession

import re

from pyspark.sql.functions import col, split, udf

from pyspark.sql.types import StringType

def clean\_text(text):

cleaned\_text = re.sub(r"[!@#$%^&\*()+=,.:;'/-]", "", text.replace("\"", "").replace("\n", ""))

return cleaned\_text

def clean\_data():

data = spark.read.csv("D:/data/Reviews.csv", header=True)

data = data.withColumn("Score", col("Score").cast("Int"))

with open("D:/data/Data\_clean\_train.csv", "w", encoding="utf-8", newline="") as file:

writer = csv.writer(file)

writer.writerow(["Text", "label"])

for dt in data.select("Text", "Score").collect():

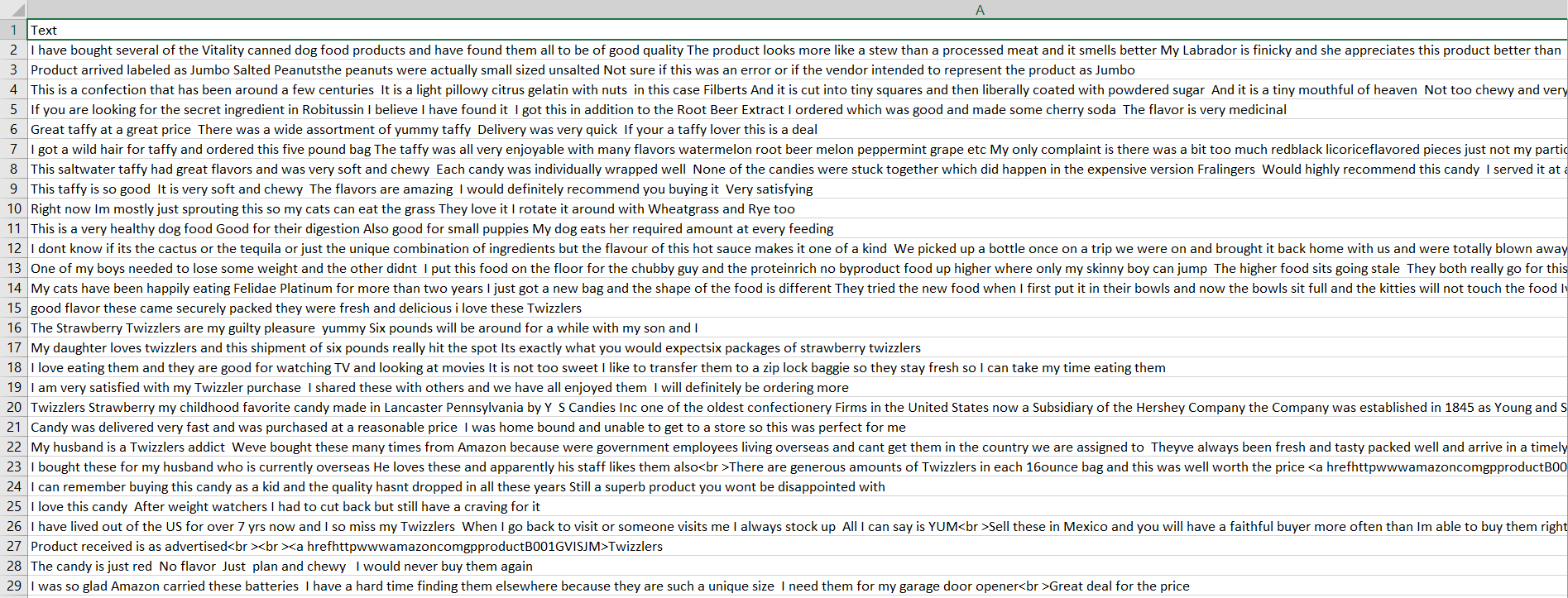
if dt["Text"] is not None:

if dt["Score"] is not None:

with open("D:/data/Data\_clean\_train.csv", "a", encoding="utf-8", newline="") as file:

writer = csv.writer(file)

writer.writerow([clean\_text(dt["Text"]), dt["Score"]])



def sentiment\_data(prediction):

if prediction == 0:

return "Saddest"

if prediction == 1:

return "Sad"

if prediction == 2:

return "Normal"

if prediction == 3:

return "Happy"

if prediction == 4:

return "Happiest"

if prediction >= 5:

return "Very Happy"

else:

return "So Sad"

def train\_data():

data = spark.read.csv("D:/data/Data\_clean\_train.csv", header=True)

data = data.withColumn("label", col("label").cast("Int"))

tokenized\_text = data.withColumn("tokens", split("Text", " "))

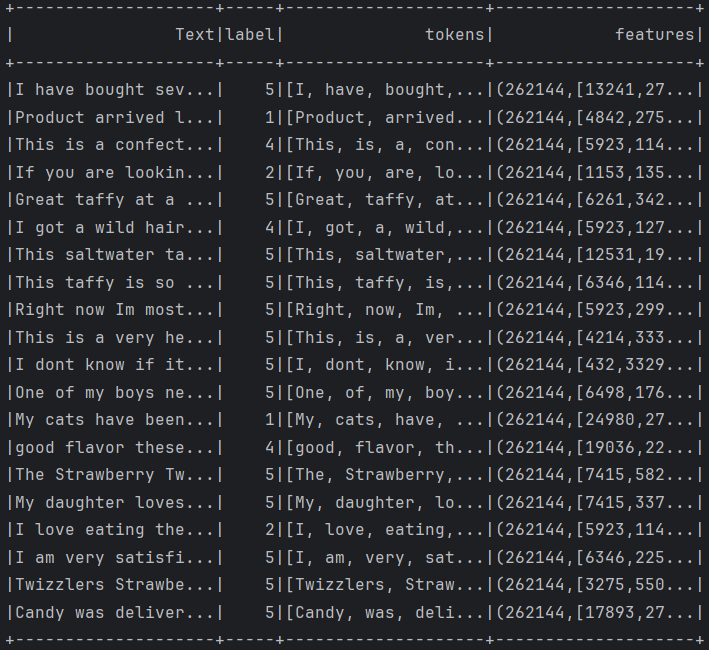
tokenized\_text.show()



hashingTF = HashingTF(inputCol="tokens", outputCol="features")

labeled\_text = hashingTF.transform(tokenized\_text)

labeled\_text.show()



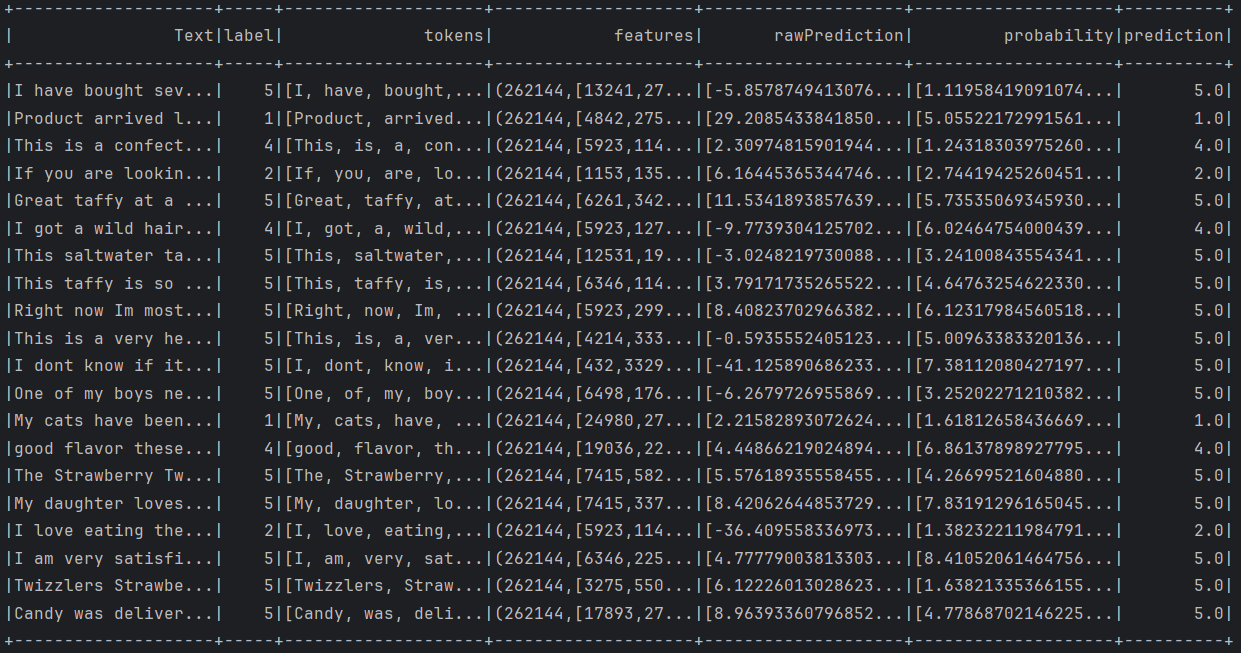
trainData, testData = labeled\_text.randomSplit([0.8, 0.2])

lr = LogisticRegression()

model = lr.fit(labeled\_text)

predictions = model.transform(labeled\_text)

predictions.show()



test\_predictions = model.transform(testData)

evaluator = MulticlassClassificationEvaluator(labelCol="label", predictionCol="prediction", metricName="accuracy")

accuracy = evaluator.evaluate(test\_predictions)

print("Độ chính xác trên tập dữ liệu kiểm tra: {:.2%}".format(accuracy))

return model



spark = SparkSession.builder \

.appName("Analysis\_data") \

.config("spark.driver.memory", "8g") \

.config("spark.shuffle.spill.numElementsForceSpillThreshold", 50000) \

.config("spark.memory.offHeap.enabled", "true") \

.config("spark.memory.offHeap.size", "1g") \

.config("spark.executorEnv.OPENBLAS\_NUM\_THREADS", "1") \

.getOrCreate()

data = spark.read.csv("D:/data/data\_tweet.csv", header=True)

data = data.select("Name", "Content Tweet")

data.show()

tokenized\_text = data.withColumn("tokens", split("Content Tweet", " "))

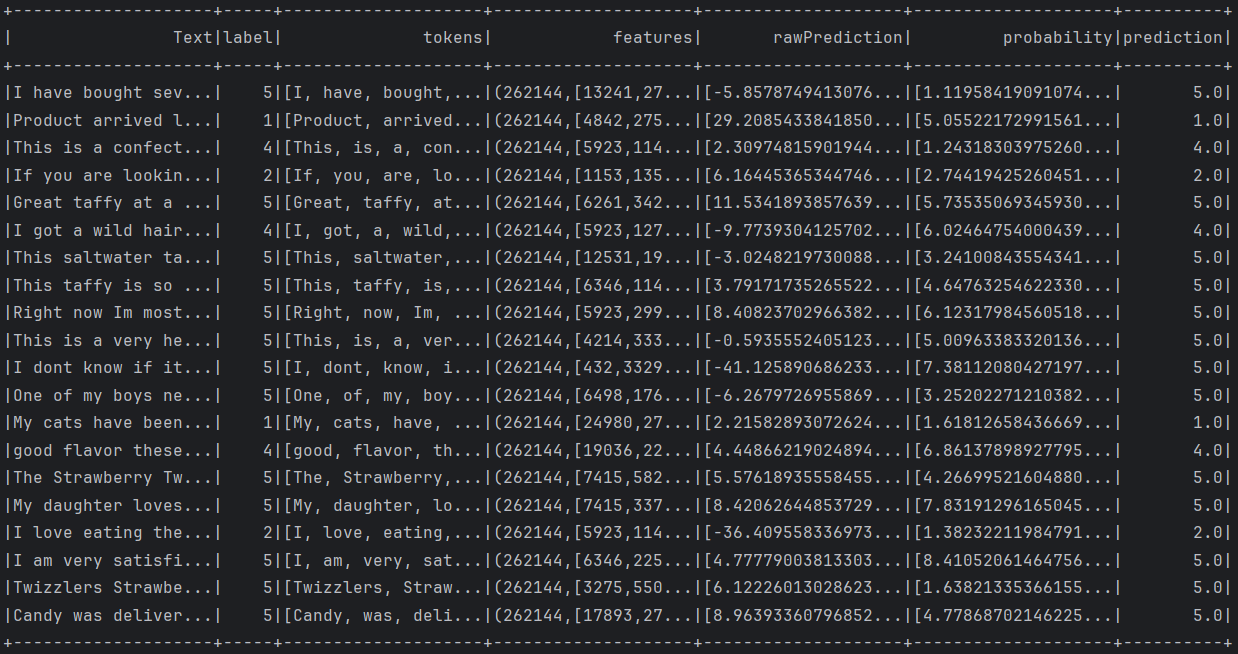
hashingTF = HashingTF(inputCol="tokens", outputCol="features")

labeled\_text = hashingTF.transform(tokenized\_text)

model = train\_data()

predictions = model.transform(labeled\_text)

predictions.show()

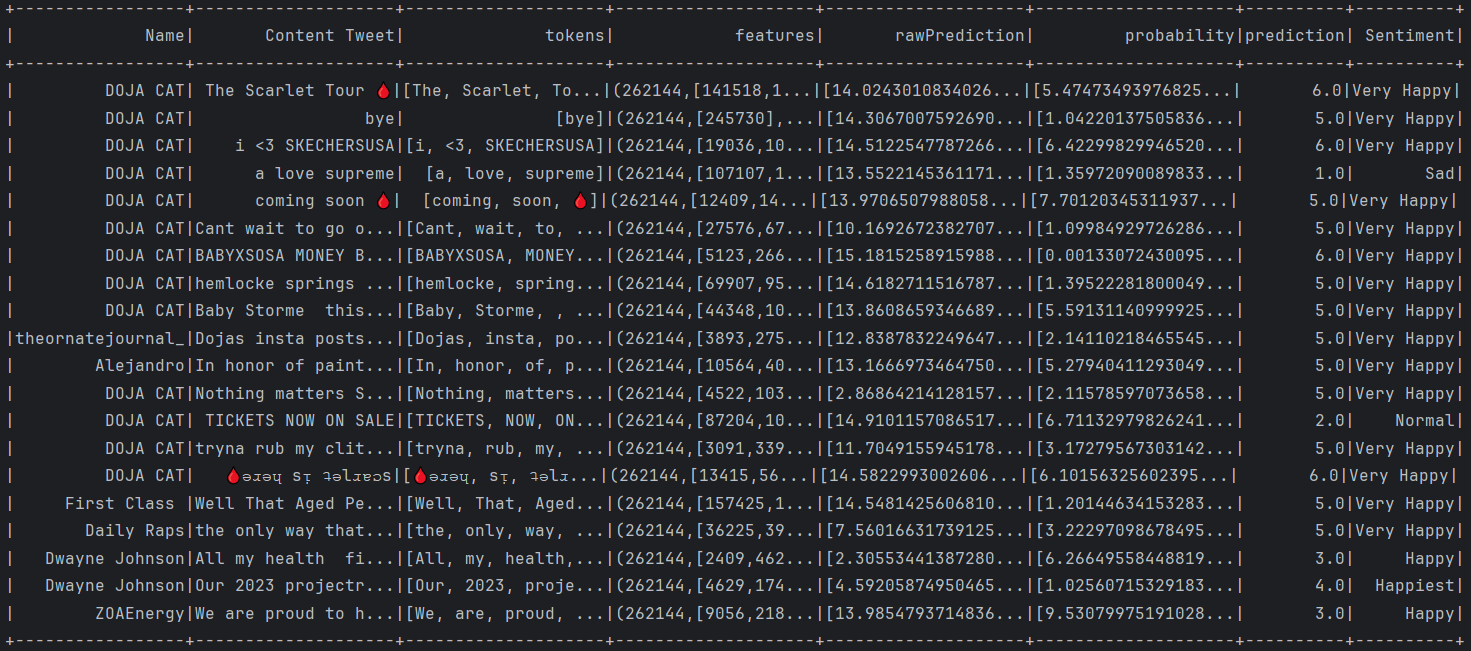


predictions.printSchema()

sentiment\_udf = udf(sentiment\_data, StringType())

predictions\_with\_sentiment = predictions.withColumn("Sentiment", sentiment\_udf(predictions["prediction"]))

predictions\_with\_sentiment.show()



1. **Đánh giá**

- Ưu điểm:

• Tích hợp với Spark: Được xây dựng trên Apache Spark, Machine Learning LogisticRegression có thể tận dụng sức mạnh của Spark để xử lý dữ liệu văn bản lớn.

Các công cụ và mô hình Machine Learning LogisticRegression: Cung cấp các công cụ và mô hình tiện ích cho việc xử lý và phân tích ngôn ngữ tự nhiên.

• Hỗ trợ tiền xử lý văn bản: Cung cấp các công cụ tiền xử lý văn bản để làm sạch và chuẩn hóa dữ liệu văn bản trước khi áp dụng các mô hình học máy.

- Nhược điểm:

• Yêu cầu kiến thức Machine Learning LogisticRegression: Sử dụng Machine Learning LogisticRegression đòi hỏi hiểu biết về ngôn ngữ tự nhiên và Machine Learning LogisticRegression, điều này có thể làm cho nó khó khăn đối với người không có kinh nghiệm trong lĩnh vực này.

• Tài nguyên tính toán: Việc xử lý và phân tích dữ liệu văn bản có thể đòi hỏi tài nguyên tính toán đáng kể, đặc biệt khi làm việc với dữ liệu lớn.

1. **Kết luận**

Phân tích cảm xúc các bài tweet là một lĩnh vực nghiên cứu quan trọng trong xử lý ngôn ngữ tự nhiên. Các kỹ thuật phân tích cảm xúc có thể được sử dụng để hiểu tâm lý của người dùng Twitter, từ đó giúp các doanh nghiệp và tổ chức xây dựng các chiến lược tiếp thị và truyền thông hiệu quả hơn.

Trong những năm gần đây, các kỹ thuật phân tích cảm xúc đã đạt được những tiến bộ đáng kể. Các mô hình phân tích cảm xúc hiện đại có thể đạt độ chính xác cao trong việc phân loại cảm xúc của các bài tweet. Tuy nhiên, vẫn còn một số thách thức cần giải quyết, chẳng hạn như khả năng nhận biết các sắc thái cảm xúc phức tạp và khả năng thích ứng với các thay đổi trong cách sử dụng ngôn ngữ.

Với sự phát triển của công nghệ, phân tích cảm xúc sẽ ngày càng trở nên chính xác và hiệu quả hơn. Các kỹ thuật phân tích cảm xúc sẽ đóng vai trò quan trọng trong việc giúp chúng ta hiểu tâm lý của con người và xây dựng các mối quan hệ hiệu quả hơn.

Mặc dù còn tồn tại những thách thức, phân tích cảm xúc các bài tweet vẫn là một kỹ thuật có tiềm năng ứng dụng rộng rãi. Trong tương lai, phân tích cảm xúc các bài tweet có thể được sử dụng để cải thiện nhiều lĩnh vực khác nhau của cuộc sống, chẳng hạn như truyền thông, tiếp thị và chăm sóc sức khỏe.

TÀI LIỆU THAM KHẢO

- Spark nlp

<https://nlp.johnsnowlabs.com/>

- Spark Download

<https://spark.apache.org/downloads.html>

- Bs4 python

<https://realpython.com/beautiful-soup-web-scraper-python/>

- selenium python

<https://www.geeksforgeeks.org/selenium-python-tutorial/>