**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**

**VIỆN TRÍ TUỆ NHÂN TẠO**

**\_ \_ \_ \_ \*\*\*\_ \_ \_ \_**

**BÁO CÁO MÔN HỌC KỸ THUẬT VÀ CÔNG NGHỆ DỮ LIỆU LỚN**

**ĐỀ TÀI**

**ỨNG DỤNG CỦA HADOOP MAP REDUCE VÀ SPARK TRONG BÀI TOÁN PHÂN TÍCH CẢM XÚC VĂN BẢN**

**Nhóm sinh viên thực hiện:**

1. Hoàng Việt Tùng – 22022663.
2. Vương Ngọc Quân – 22022616.
3. Phạm Văn Trường – 22022564.
4. Trần An Thắng – 22022525.

**Giảng viên hướng dẫn:** TS. Trần Hồng Việt

ThS. Ngô Minh Hương

CN. Lương Sơn Bá

**MỞ ĐẦU**

Trong thế giới dữ liệu ngày nay, phân tích cảm xúc đã trở thành một công cụ quan trọng để hiểu ý kiến, cảm xúc và phản hồi trong nhiều lĩnh vực khác nhau. Khi khối lượng dữ liệu tăng lên nhanh chóng, việc tận dụng các nền tảng mạnh mẽ để xử lý dữ liệu quy mô lớn trở nên cần thiết. Báo cáo này khám phá việc triển khai Hadoop và Spark, hai công nghệ big data hàng đầu, trong bối cảnh phân tích cảm xúc.

Hadoop cung cấp một hệ thống lưu trữ và xử lý phân tán, lý tưởng để xử lý khối lượng lớn dữ liệu phi cấu trúc, trong khi Spark mang đến khả năng tính toán trong bộ nhớ, giúp cải thiện tốc độ và hiệu quả đáng kể. Bằng cách kết hợp hai công nghệ này, dự án hướng đến cung cấp một giải pháp phân tích cảm xúc có khả năng mở rộng và hiệu quả, đồng thời thể hiện tiềm năng của chúng trong việc trích xuất thông tin ý nghĩa từ các tập dữ liệu lớn.

Báo cáo trình bày chi tiết phương pháp, các bước triển khai và đánh giá hiệu suất, đồng thời nêu bật những lợi ích thực tiễn cũng như các thách thức gặp phải khi sử dụng Hadoop và Spark trong bài toán phân tích cảm xúc văn bản.

Báo cáo gồm 4 chương:

Chương 1: Tổng quan về dữ liệu lớn và bài toán phân tích cảm xúc văn bản.

Chương 2: Ứng dụng Hadoop giải quyết bài toán phân tích cảm xúc văn bản

Chương 3: Ứng dụng Apache Spark giải quyết bài toán phân tích cảm xúc văn bản.

Chương 4: Kết luận và hướng phát triển.

# CHƯƠNG 1: TỔNG QUAN VỀ DỮ LIỆU LỚN VÀ BÀI TOÁN PHÂN TÍCH CẢM XÚC VĂN BẢN.

## Định nghĩa.

Theo Wikipedia: Dữ liệu lớn là một thuật ngữ chỉ bộ dữ liệu lớn hoặc phức tạp mà các phương pháp truyền thống không đủ các ứng dụng để xử lý dữ liệu này.

Theo Gartner: Dữ liệu lớn là những nguồn thông tin có đặc điểm chung khối lượng lớn, tốc độ nhanh và dữ liệu định dạng dưới nhiều hình thức khác nhau, do đó muốn khai thác được đòi hỏi phải có giải pháp mới để đưa ra quyết định khám phá và tối ưu hóa quá trình.

Dữ liệu đến từ rất nhiều nguồn khác nhau:

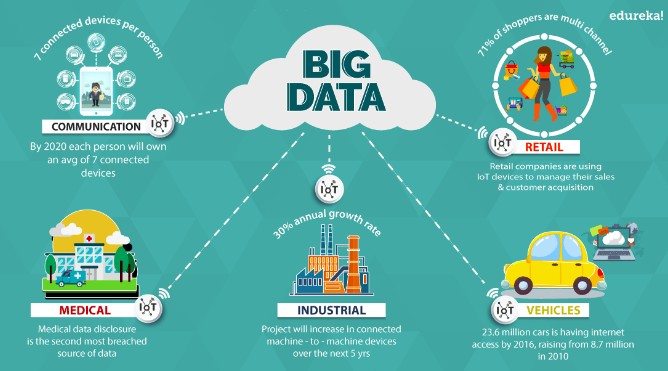


Figure . Hình minh họa nguồn gốc dữ liệu.

Một số lợi ích có thể mang lại như: Cắt giảm chi phí, tiết kiệm thời gian và giúp tối ưu hóa sản phẩm, hỗ trợ con người đưa ra những quyết định đúng và hợp lý hơn.

## Đặc trưng cơ bản của dữ liệu lớn.

1. Khối lượng lớn (Volume): Khối lượng dữ liệu rất lớn và đang ngày càng tăng lên, tính đến 2014 thì có thể lên tới khoảng vài trăm terabyte.
2. Tốc độ (Velocity): Khối lượng dữ liệu gia tăng rất nhanh.
3. Đa dạng (Variety): Ngày nay hơn 80% dữ liệu được sinh ra là dữ liệu phi cấu trúc (tài liệu, blog, hình ảnh, …).
4. Độ tin cậy/ chính xác (Veracity): Bài toán phân tích và loại bỏ dữ liệu thiếu chính xác và nhiễu đang là tính chất quan trọng của BigData.
5. Giá trị (Value): Giá trị thông tin mang lại.

## Tổng quan về Hadoop.

Theo Apache Hadoop: Apache Hadoop là một framework dùng để chạy những ứng dụng trên 1 cluster lớn được xây dựng trên những phần cứng thông thường.

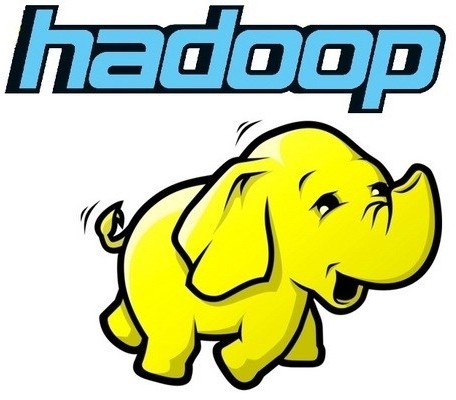


Figure . Biểu tượng của Hadoop.

Các thành phần của Hadoop bao gồm: Core, MapReduce engine, HDFS, Hbase, Hive, Pig, Chukwa,… Tuy nhiên chúng ta sẽ tập trung vào 2 thành phần quan trọng nhất là HDFS và MapReduce.

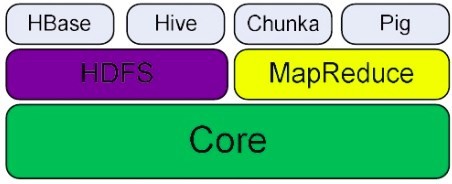


Figure . Thành phần của Hadoop.

Hadoop hiện thực mô hình Map/Reduce, đây là mô hình mà dữ liệu sẽ được chia nhỏ ra thành nhiều phân đoạn khác nhau và các phần này sẽ được chạy song song trên nhiều node khác nhau.

Thêm vào đó, Hadoop cung cấp 1 hệ thông file phân tán (HDFS) cho phép lưu trữ dữ liệu trên nhiều node. Cả Map/Reduce và HDFS đều được thiết kế sao cho framework sẽ tự động quản lý được các lỗi, các hư hỏng về phần cứng của node.

## Tổng quan về MapReduce.

Định nghĩa: Theo Google, MapReduce là mô hình được sử dụng để xử lý tính toán song song và phân tán trên hệ thống phân tán.

Bước 1: Phân rã từ nghiệp vụ chính (do người dùng muốn thể hiện) thành các công việc con để chia từng công việc con này về các máy tính trong hệ thống thực hiện xử lý một cách song song.

Bước 2: Thu thập lại các kết quả.

Ứng dụng của MapReduce:

+ Dữ liệu cần xử lý có kích thước lớn.

+ Các ứng dụng thực hiện xử lý, phân tích dữ liệu, thời gian xử lý đáng kể, có thể tính bằng phút, giờ,…

Thực thi mô hình MapReduce:

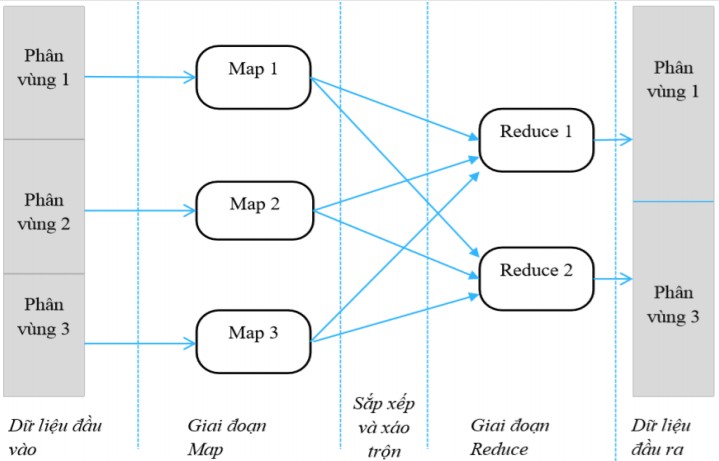


Figure . Thực thi mô hình MapReduce

+ Hàm Map: Hàm Map tiếp nhận mảnh dữ liệu input, rút trích thông tin cần thiết từ từng phần tử (ví dụ: lọc dữ liệu, hoặc trích dữ liệu) tạo kết quả trung gian.

+ Hàm Reduce: Tổng hợp kết quả trung gian, tính toán để cho kết quả cuối cùng.

## Tổng quan về Spark.

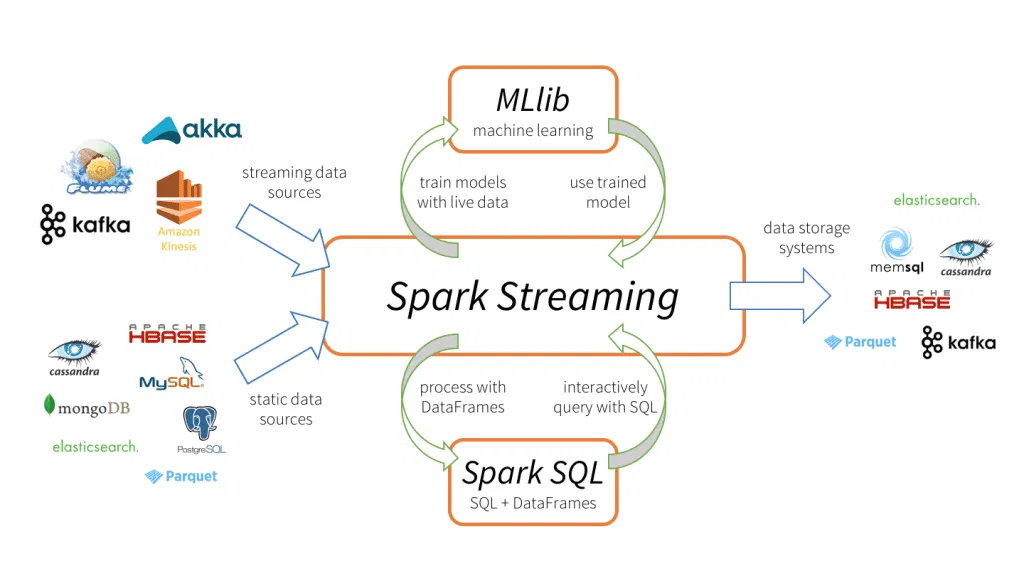


Figure . Tổng quan về Spark.

Apache Spark là framework tính toán cụm mã nguồn mở được thiết kế để cung cấp tốc độ tính toán, khả năng mở rộng (scalability) và khả năng lập trình cho Big Data. Đặc biệt, Spark có khả năng cung cấp quá trình truyền một luồng dữ liệu liên tục (data streaming) với tốc độ cao, cơ sở dữ liệu đồ thị (graph data), học máy (machine learning) và ứng dụng trí tuệ nhân tạo (AI).

Apache Spark được xem là công cụ xử lý dữ liệu nhanh gấp 10 lần so với các công cụ khác. Nó cung cấp một cơ chế tính toán phân tán trên một cụm máy tính (Computer Cluster) với hiệu suất cao hơn so với Apache Hadoop đặc biệt là MapReduce, đồng thời hỗ trợ nhiều ngôn ngữ lập trình phổ biến như Scala, Java, Python,…

Các thành phần của Apache Spark bao gồm: Spark Core, Spark SQL, Spark Streaming, MLlib, GraphX,…

## 1.6 Bài toán phân tích cảm xúc văn bản.

Phân tích quan điểm (sentiment analysis) là một ứng dụng của trí tuệ nhân tạo, nó sử dụng các thuật toán phức tạp để xử lý ngôn ngữ tự nhiên của con người (NLP) và xác định các đặc điểm cảm xúc tiêu cực/tích cực tại một thời điểm thông qua văn bản hoặc lời nói. Các nguồn dữ liệu được phân tích phổ biến như Social media, Blog, Website đánh giá sản phẩm, tổng đài Contact center,…

Hiện nay, nhờ sự tiến bộ của các công nghệ thế hệ mới, các thuật toán phân tích quan điểm ngày càng được nâng cấp với độ chính xác cao, từ đó hỗ trợ trong các sản phẩm thông minh như trợ lý ảo tích hợp trên ô tô, căn hộ… cải thiện chất lượng cuộc sống của con người.

Lợi ích của việc giải quyết bài toán:

* Xác định và trích xuất thông tin hữu ích từ khách hàng:

Việc nhận biết được cảm xúc của người dùng trong cuộc trò chuyện giúp doanh nghiệp phân tích được mức độ quan tâm của khách hàng đối với thương hiệu, sản phẩm/ dịch vụ của mình. Đây là nguồn thông tin có giá trị cao giúp doanh nghiệp điều chỉnh chiến lược về sản phẩm, kinh doanh, marketing và đặc biệt là các dịch vụ tư vấn phù hợp.

* Tự động đánh giá chất lượng của các tư vấn viên hoặc tổng đài viên:

Thông qua cảm xúc của khách hàng trong các cuộc trò chuyện bằng tin nhắn hoặc cuộc gọi, doanh nghiệp dễ dàng xác định được thái độ và sự hài lòng của khách hàng đối với hoạt động tư vấn. Từ đó dễ dàng đánh giá được chất lượng của nhân viên và quản lý hiệu quả hơn.

Ở dự án này, nhóm thực hiện thí nghiệm với bộ dữ liệu bao gồm 1,6 triệu tweets được thu thập từ nền tảng mạng xã hội X. Mỗi tweet đã được đánh nhãn 0 (tiêu cực) hoặc 1 (tích cực).

# CHƯƠNG 2: ỨNG DỤNG HADOOP GIẢI QUYẾT BÀI TOÁN PHÂN TÍCH CẢM XÚC VĂN BẢN.

## 2.1 Giới thiệu thuật toán Naïve Bayes.

Naive Bayes là một thuật toán học máy thuộc nhóm các mô hình xác suất, dựa trên định lý Bayes và giả định mạnh mẽ rằng các đặc trưng (features) là độc lập với nhau. Mặc dù giả định này không thực tế trong nhiều trường hợp, Naive Bayes vẫn hoạt động hiệu quả trong nhiều bài toán phân loại, bao gồm phân loại văn bản.

Đối với bài toán phân loại cảm xúc tweet, Naive Bayes có thể dự đoán một tweet thuộc về một nhãn cảm xúc nhất định (chẳng hạn như **tích cực**, **tiêu cực**, hoặc **trung lập**) dựa trên tần suất và sự xuất hiện của từ ngữ trong dữ liệu huấn luyện.

Ưu điểm của Naive Bayes:

* Hiệu quả tính toán, nhanh chóng trong xử lý dữ liệu lớn.
* Hoạt động tốt trên các tập dữ liệu văn bản.

Nhược điểm:

* Giả định các đặc trưng độc lập không phù hợp với nhiều dữ liệu thực tế.
* Có thể bị ảnh hưởng bởi các từ xuất hiện quá ít hoặc quá thường xuyên.

## 2.2 Triển khai thuật toán Naïve Bayes.

Bước 1: Thu thập dữ liệu.

* Chuẩn bị dữ liệu bao gồm nội dung của các tweet và nhãn của chúng.

Bước 2: Tiền xử lý dữ liệu.

* Loại bỏ các ký tự đặc biệt, liên kết URL và dấu câu.
* Chuyển đổi các từ về dạng chữ thường.
* Áp dụng các kỹ thuật xử lý ngôn ngữ tự nhiên như loại bỏ stop words hoặc stemming (chuyển từ về gốc).

Bước 3: Biểu diễn dữ liệu.

* Chuyển đổi dữ liệu văn bản sang dạng số bằng cách sử dụng các ký thuật như Bag of Words (BoW), TF-IDF,…

Bước 4: Tính toán xác suất.

Sử dụng định lý Bayes:

P(C|X) = P(X|C) . P(C) / P(X)

Trong đó:

P(C|X): Xác suất tweet thuộc lớp C (nhãn cảm xúc) khi biết đặc trưng X.

P(X|C): Xác suất của tập hợp từ X xuất hiện trong lớp C.

P(C): Xác suất tiên nghiệm của lớp C.

P(X): Xác suất cảu từ X trên toàn bộ dữ liệu.

Bước 5: Huấn luyện mô hình:

* Tính toán P(C) và P(X|C) từ dữ liệu huấn luyện.

Bước 6: Dự đoán:

* Đối với một tweet mới, tính xác suất P(C|X) cho từng lớp C, và chọn lớp có xác suất cao.

## 2.3 Ví dụ minh họa thuật toán.

Dữ liệu mẫu:

|  |  |
| --- | --- |
| Tweet | Cảm xúc |
| “Tuyệt vời! Tôi thích nó.” | Tích cực |
| “Thật là tồi tệ.” | Tiêu cực |
| “Rất tốt, tôi rất hài lòng!” | Tích cực |
| “Quá tệ, tôi thất vọng.” | Tiêu cực |

Bước 1: Chuẩn bị dữ liệu.

* Tập dữ liệu gồm 4 tweet và 2 nhãn cảm xúc Tích cực, Tiêu cực.

Bước 2: Tiền xử lý dữ liệu.

* Tweet: “Tuyệt vời! Tôi thích nó.” -> “tuyệt vời tôi thích nói”

Bước 3: Biểu diễn dữ liệu.

* Sử dụng BoW hoặc TF-IDF để biểu diễn dữ liệu

Bước 4: Tính toán xác suất.

* Giả sử ta có:

P(Tích cực) = ½, P(Tiêu cực) = ½

Xác suất P(“tuyệt” | Tích cực) = 1/11 (từ “tuyệt” xuất hiện 2 lần trong các tweet tích cực)

Bước 5: Huấn luyện mô hình.

* Tạo bảng xác suất cho mỗi từ trong từ điển với từng nhãn cảm xúc dựa trên dữ liệu huấn luyện.

Bước 6: Dự đoán cảm xúc.

* Tweet mới: “Sản phẩm rất tuyệt!” -> “sản phẩm rất tuyệt”
* Tính xác suất P(Tích cực|X), P(Tiêu cực|X)
* Chọn nhãn cảm xúc có xác suất cao nhất.

## 2.4 Triển khai thuật toán Naïve Bayes với TF-IDF.

Sử dụng TF-IDF (Term Frequency – Inverse Document Frequency) để chuyển đổi văn bản thành các đặc trưng số học.

TF-IDF gán trọng số cao hơn cho những từ quan tọng trong một tài liệu và xuất hiện ít trong các tài liệu khác.

Với mỗi từ, TF-IDF được tính như sau TF-IDF(t,d) = TF(t,d) . logDF(t) . N

Trong đó:

TF(t,d): Tần suất từ t trong tài liệu d.

N: Tổng số tài liệu.

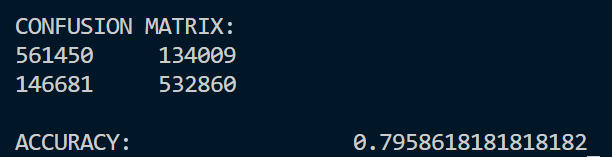
DF(t): Số tài liệu chứa từ t.

Sau khi tính toán TF-IDF cho toàn bộ từ, chọn 75% từ quan trọng nhất dựa trên trọng số TF-IDF của chúng, nhằm giảm số chiều dữ liệu và tập trung vào các đặc trưng quan trọng.

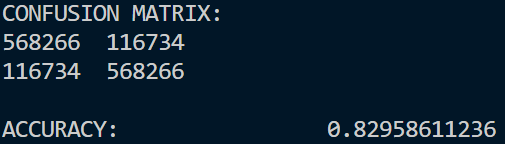
## 2.5 Kết quả triển khai.

Để thực hiện thí nghiệm thuật toán Naïve Bayes với tập dữ liệu đã chuẩn bị ở trên, nhóm sẽ thực hiện lệnh put để đẩy bộ dữ liệu lên hệ thống HDFS, sau đó sử dụng hadoop để Map Reduce hóa bài toán.

Kết quả triển khai thuật toán Naïve Bayes ban đầu:



Kết quả triển khai thuật toán Naïve Bayes kết hợp trọng số TF-IDF:



# CHƯƠNG 3: ỨNG DỤNG APACHE SPARK GIẢI QUYẾT BÀI TOÁN PHÂN TÍCH CẢM XÚC VĂN BẢN.

## 3.1 Đánh giá tổng quan về Spark.

Việc sử dụng Apache Spark để giải quyết bài toán này đem đến cho nhóm những lợi ích sau:

* Xử lý dữ liệu lớn: Khả năng mở rộng xử lý các tập dữ liệu lớn, ở đây là 1,6 triệu tweets.
* Tích hợp dễ dàng: MLlib cung cấp các công cụ học máy sẵn có, giảm công sức cài đặt.
* Hiệu năng cao: Với cơ chế xử lý trong bộ nhớ, Spark giúp giảm thời gian đọc/ghi dữ liệu.

## 3.2 Đánh giá tổng quan thuật toán Naïve Bayes và SVM trong việc xử lý bài toán.

a) Naïve Bayes là một mô hình xác suất dựa trên Định lý Bayes với giả định mạnh mẽ rằng các đặc trưng (features) độc lập với nhau.

Ưu điểm:

* Đơn giản và nhanh: Dễ dàng triển khai và huấn luyện, kể cả trên dữ liệu lớn. Tính toán nhanh, phù hợp với các bài toán thời gian thực.
* Hiệu quả với văn bản: Đặc biệt hiểu quả với bài toán phân loại văn bản, bao gồm sentiment analysis, vì giả định độc lập của các từ thường hoạt động tốt trong các bộ dữ liệu ngôn ngữ tự nhiên.
* Không yêu cầu tài nguyên: Cần ít tài nguyên tính toán hơn so với các mô hình phức tạp như SVM.
* Khả năng xử lý tốt dữ liệu mất cân bằng: Naïve Bayes thường hoạt động tốt trên các tập dữ liệu mà số lượng mẫu giữa các lớp không cân bằng.

Nhược điểm:

* Giả định độc lập: Giả định các đặc trưng độc lập không phải lúc nào cũng đúng trong bài toán ngôn ngữ tự nhiên (ví dụ: từ ngữ thường liên quan đến nhau).
* Hiệu suất thấp hơn khi dữ liệu phức tạp: Với dữ liệu không tuân theo giả định độc lập, độ chính xác của Naïve Bayes có thể giảm.
* Không phù hợp với các biên quyết định phức tạp: Khả năng phân tách dữ liệu không tuyến tính kém hơn so với SVM.

b) Support Vector Machine (SVM) là một mô hình học máy phân loại dựa trên việc tìm kiếm siêu phẳng tối ưu để phân tách dữ liệu:

Ưu điểm:

* Hiệu suất cao: Hoạt động tốt hơn trong các bài toán có dữ liệu phức tạp hoặc biên quyết định không tuyến tính.
* Hỗ trợ không gian đặc trưng cao: SVM hoạt động tốt với nhiều đặc trưng (như trong văn bản, khi số chiều lớn).
* Khả năng tổng quát hóa tốt: Ít bị overfitting hơn, đặc biệt so với các kỹ thuật kernel.
* Thích hợp với sentiment analysis: Thường cho kết quả chính xác cao hơn so với Naïve Bayes khi xử lý dữ liệu có mẫu phức tạp.

Nhược điểm:

* Tốn thời gian và tài nguyên: Quá trình huấn luyện chậm hơn nhiều so với Naïve Bayes, đặc biệt trên các tập dữ liệu lớn.
* Nhạy cảm với tham số: Hiệu suất của SVM phụ thuộc vào việc chọn kernel, hệ số điều chỉnh C, và các tham số khác, điều này có thể phức tạp.
* Dữ liệu mất cân bằng: SVM có thể không hiệu quả nếu không được điều chỉnh trên tập dữ liệu có phân bố lớp không đều.

## 3.3 Triển khai Naïve Bayes với Apache Spark.

Bước 1: Đọc và làm sạch dữ liệu.

* Dữ liệu được đọc vào từ các tệp CSV, sử dụng Spark RDD để xử lý dữ liệu.
* Áp dụng các bước tiền xử lý để làm sạch nội dung tweet.

Bước 2: Chuyển đổi dữ liệu thành DataFrame.

* Dữ liệu được chuyển đổi từ RDD thành DataFrame.

Bước 3: Tokenization và tính TF-IDF.

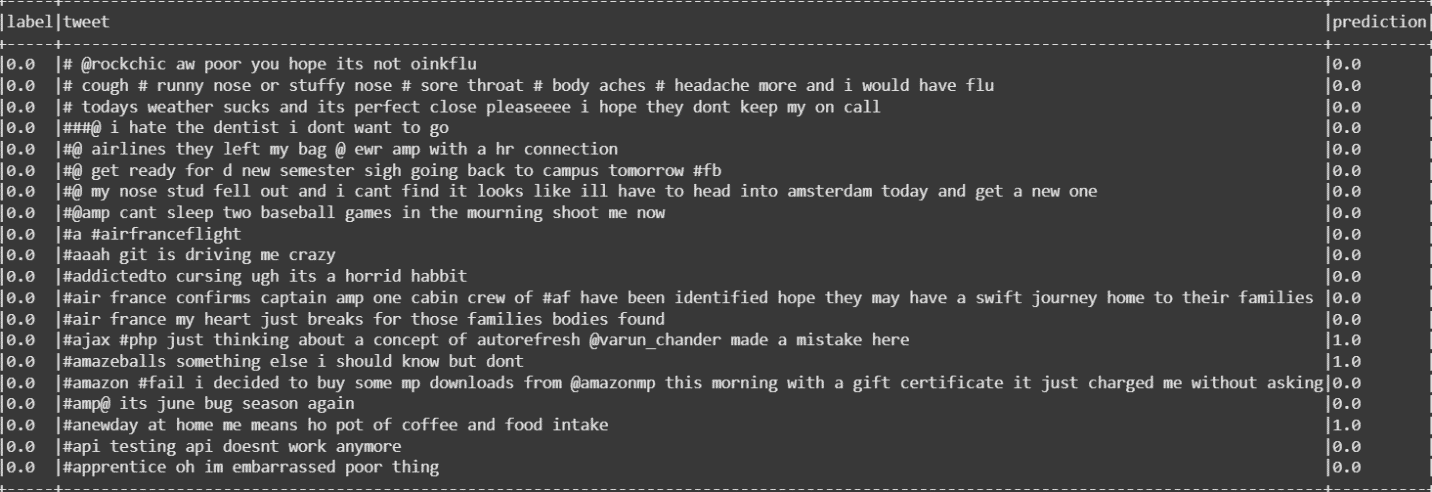
* Sử dụng tokenizer trong Spark Mllib để chuyển nội dung tweet thành danh sách các từ.
* Kết quả: VD: tweet: “this is amazing product” => words: [“this”, “is”, “amazing”, “product”].
* Tính toán TF-IDF.

Bước 4: Huấn luyện mô hình Naïve Bayes.

* Sử dụng NaiveBayes từ thư viện Mllib.

Bước 5: Dự đoán và đánh giá.

Kết quả dự đoán 20 mẫu đầu tiên trong tập dữ liệu kiểm thử:



Confusion matrix:

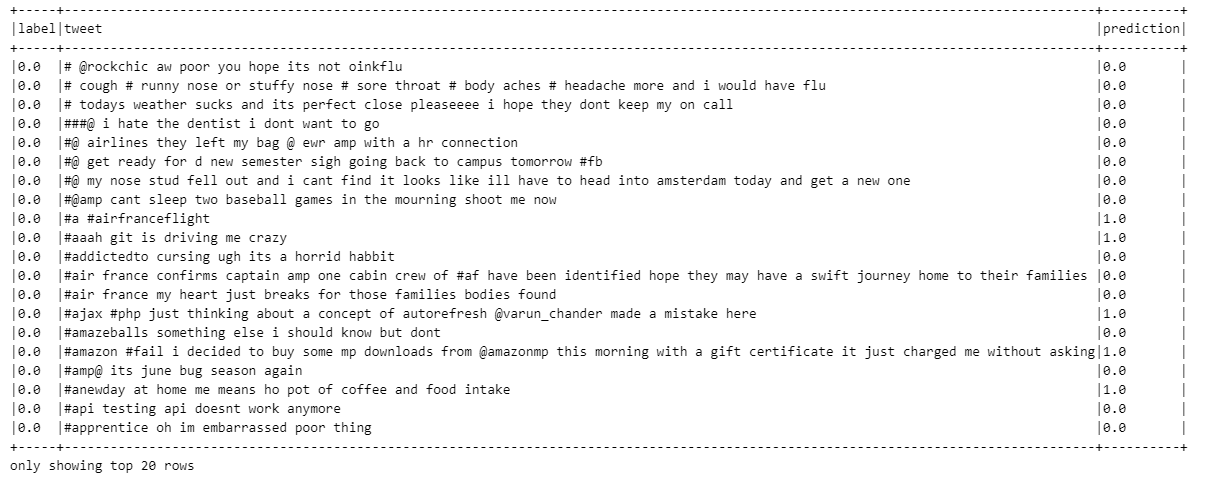


Độ chính xác (accuracy): 0.74. Thời gian thực thi: 570s.

## 3.4 Triển khai SVM với Apache Spark.

Các bước được thực hiện tương tự như trên nhưng thay vì sử dụng NaiveBayes từ thư viện MLlib, ta sử dụng LinearSVC từ pyspark.ml.classification.

Kết quả dự đoán 20 mẫu đầu tiên trong tập dữ liệu kiểm thử:



Độ chính xác (accuracy): 0.78. Thời gian thực thi: 960s.

## 3.4 Triển khai mô hình DistilBERT với Apache Spark.

Dữ liệu được xử lý tương tự như với mô hình Naive Bayes. Tuy nhiên có một vài khác biệt, vì đây là mô hình học sâu nên sẽ không sử dụng TF-IDF mà thay vào đó sử dụng Tokenizer của mô hình đã được pre-train từ trước:

tokenizer = DistilBertTokenizer.from\_pretrained("distilbert-base-uncased-finetuned-sst-2-english")

Vì vậy, tập dữ liệu được biểu diễn dưới dạng Dataset của PyTorch, bao gồm:

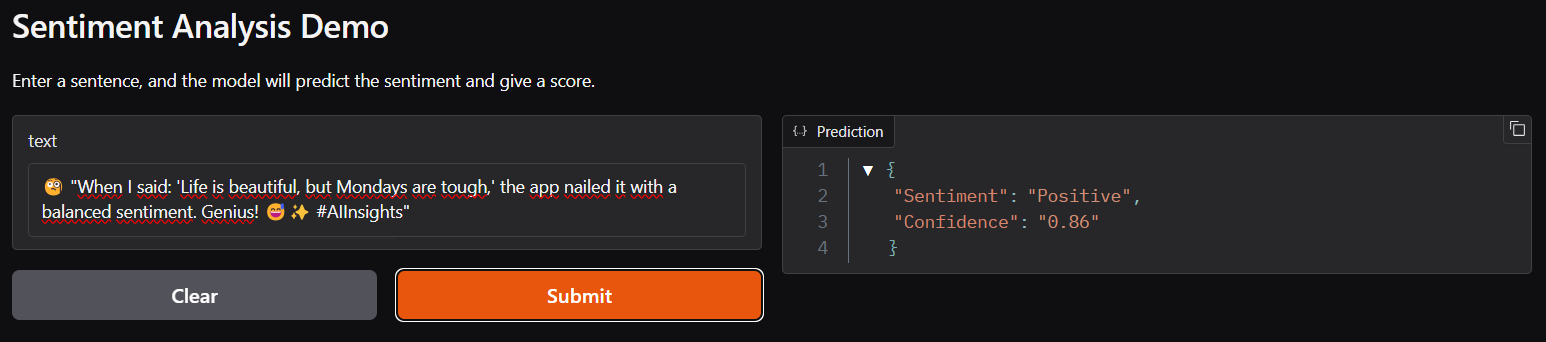
* input\_ids: Chuỗi mã hóa của từ.
* attention\_mask: Mặt nạ đánh dấu từ hợp lệ.
* labels: Nhãn cảm xúc (0 hoặc 1).

Sử dụng mô hình tiền huấn luyện DistilBERT-base-uncased, tinh chỉnh (fine-tune) cho bài toán phân loại cảm xúc với các tham số huấn luyện:

* Số epoch: 3.
* Batch size: 64.
* Learning rate: 2e-5.
* Thực hiện huấn luyện trên GPU của Kaggle, lưu lại mô hình tốt nhất trong quá trình.

Kết quả thử nghiệm:

* Mô hình được đánh giá trên tập kiểm thử với các chỉ số:
* Độ chính xác: 92% (cao hơn so với 74% của Naive Bayes).
* F1 Score: Cải thiện đáng kể, đặc biệt với dữ liệu không cân bằng.
* Kết quả cho thấy DistilBERT vượt trội hơn Naive Bayes và SVM nhờ khả năng học ngữ cảnh và xử lý văn bản dài.

Demo mô hình ở Huggingface Space: [Demo app](https://huggingface.co/spaces/Smrfhdl/bigdataproject)

# CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.

## Kết luận.

Báo cáo đã phân tích hiệu quả của các mô hình học máy, bao gồm **Naive Bayes** và **SVM**, trong bài toán sentiment analysis sử dụng Hadoop và Apache Spark. Mỗi mô hình mang những đặc điểm riêng:

* **Naive Bayes** đơn giản, nhanh chóng và phù hợp với dữ liệu có tính độc lập cao giữa các đặc trưng.
* **SVM** vượt trội hơn trong việc xử lý dữ liệu phức tạp nhờ khả năng tìm kiếm biên quyết định tối ưu, tuy nhiên đòi hỏi tài nguyên và thời gian huấn luyện lớn hơn.

Kết quả thực nghiệm cho thấy, việc lựa chọn mô hình phù hợp cần dựa trên đặc điểm dữ liệu và yêu cầu cụ thể của bài toán.

## Hướng phát triển.

Để nâng cao hiệu quả của sentiment analysis trong tương lai, một số định hướng phát triển có thể được cân nhắc:

1. **Tăng cường xử lý ngôn ngữ tự nhiên (NLP)**: Sử dụng các kỹ thuật tiên tiến như **word embeddings** (Word2Vec, GloVe) hoặc các transformer models (e.g., **RoBERTa**, **GPT**) để cải thiện khả năng biểu diễn ngữ nghĩa của từ.
2. **Cải thiện khả năng xử lý dữ liệu không cân bằng**: Áp dụng các phương pháp điều chỉnh trọng số (class weights) hoặc tạo mẫu dữ liệu (oversampling/undersampling) để tối ưu hóa hiệu suất trên dữ liệu lệch nhãn.
3. **Mở rộng ứng dụng thực tiễn**: Tích hợp mô hình sentiment analysis vào các hệ thống thương mại điện tử, mạng xã hội hoặc chatbot để phân tích ý kiến khách hàng theo thời gian thực.
4. **Tối ưu hóa hiệu năng mô hình**: Khai thác các framework hiện đại để triển khai mô hình hiệu quả hơn trên các hệ thống phân tán (e.g., Spark MLlib, TensorFlow).
5. **Nghiên cứu trên dữ liệu đa ngôn ngữ**: Phát triển các mô hình có khả năng xử lý sentiment analysis cho nhiều ngôn ngữ, đặc biệt là các ngôn ngữ ít tài nguyên như tiếng Việt.

Bằng cách tiếp tục tối ưu và mở rộng ứng dụng, sentiment analysis sẽ đóng vai trò ngày càng quan trọng trong các lĩnh vực nghiên cứu và công nghiệp.