TRƯỜNG ĐẠI HỌC CÔNG NGHỆ - ĐẠI HỌC QUỐC GIA HÀ NỘI

VIỆN TRÍ TUỆ NHÂN TẠO

----------***----------

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

ĐỀ TÀI

HỆ THỐNG ĐỀ XUẤT PHIM DỰA TRÊN ITEM COLLABORATIVE FILTERING & HADOOP MAPREDUCE

Nhóm sinh viên thực hiện:

1. Đỗ Xuân Cảnh
2. Trần Hùng Đức
3. Lê Văn Đức

Giảng viên hướng dẫn: TS. Trần Hồng Việt

CN. Đỗ Thu Uyên

Hà Nội, ngày 26 tháng 11 năm 2024

LỜI MỞ ĐẦU

Trong kỷ nguyên số hiện nay, sự bùng nổ dữ liệu đã mang đến cả cơ hội lẫn thách thức mới cho các doanh nghiệp trong việc hiểu và phục vụ khách hàng ngày càng tốt hơn. Mỗi ngày, khối lượng dữ liệu khổng lồ được tạo ra từ các hoạt động trực tuyến và mạng xã hội. Khai thác hiệu quả nguồn dữ liệu này để đưa ra các gợi ý sản phẩm phù hợp không chỉ là xu hướng mà còn là yếu tố cốt lõi trong chiến lược kinh doanh của nhiều công ty.

Dự án phân tích dữ liệu lớn với mục tiêu đề xuất sản phẩm cá nhân hóa được triển khai nhằm tối ưu hóa trải nghiệm mua sắm của khách hàng và thúc đẩy doanh thu. Hệ thống này tận dụng các kỹ thuật phân tích dữ liệu hiện đại, kết hợp với thuật toán máy học tiên tiến để phân tích hành vi, sở thích của người dùng, từ đó đưa ra các đề xuất sản phẩm chính xác, phù hợp với nhu cầu riêng của từng cá nhân.

Dự án đặc biệt tập trung vào việc ứng dụng mô hình học máy để xây dựng một hệ thống gợi ý phim thông minh. Kỳ vọng đặt ra là tạo nên một hệ thống không ngừng học hỏi từ dữ liệu mới, nâng cao độ chính xác của các gợi ý và mang lại giá trị thực tiễn rõ rệt cho cả người dùng lẫn doanh nghiệp. Báo cáo này bao gồm:

Chương 1: Tổng quan về dữ liệu lớn

Chương 2: Xây dựng hệ thống gợi ý phim với thuật toán gợi ý sản phẩm

Chương 3: Thử nghiệm và đánh giá kết quả

Chương 4: Kết luận và hướng phát triển

DANH SÁCH CÔNG VIỆC CỦA TỪNG THÀNH VIÊN

STT	Họ và tên	Mã sinh viên	Công việc
1	Đỗ Xuân Cảnh	22022573	• Tìm hiểu nội dung về Big Data • Thu thập dữ liệu • Code thuật toán và xây dựng website demo • Viết báo cáo
2	Trần Hùng Đức	22022513	• Tìm hiểu về thuật toán Item-based Collaborative Filtering • Tiền xử lý dữ liệu • Làm slide thuyết trình • Code thuật toán và làm giao diện website
3	Lê Văn Đức	22022657	• Tìm hiểu về Item-based Collaborative Filtering kết hợp với Hadoop MapReduce • Làm slide thuyết trình • Code thuật toán và chạy bằng java

MỤC LỤC

CHƯƠNG 1: TỔNG QUAN VỀ DỮ LIỆU LỚN

I. Giới thiệu về Big Data

1.1. Định nghĩa

Big Data là thuật ngữ chỉ việc xử lý các tập hợp dữ liệu khổng lồ và phức tạp, vượt quá khả năng xử lý của các công cụ và phương pháp truyền thống. Dữ liệu lớn bao hàm nhiều thách thức, từ phân tích, thu thập, giám sát, tìm kiếm, chia sẻ, lưu trữ, truyền tải, trực quan hóa, truy vấn, cho đến đảm bảo tính riêng tư. Thuật ngữ này thường nhấn mạnh việc ứng dụng các phương pháp phân tích tiên tiến như phân tích dự đoán, hành vi người dùng, hoặc các kỹ thuật hiện đại khác nhằm khai thác giá trị từ dữ liệu, thay vì chỉ tập trung vào kích thước của tập dữ liệu.

1.2. Đặc trưng của Big Data

- **Volume (Khối lượng):** Đề cập đến lượng dữ liệu khổng lồ cần được xử lý, thường lên đến hàng terabyte (TB) hoặc hơn.
- **Velocity (Tốc độ):** Chỉ tốc độ nhanh chóng trong việc tạo ra và xử lý dữ liệu, thường diễn ra gần như theo thời gian thực.
- **Variety (Đa dạng):** Phản ánh sự phong phú trong các dạng dữ liệu, từ văn bản, hình ảnh, video đến các dữ liệu phi cấu trúc khác.
- **Veracity (Độ tin cậy):** Tập trung vào việc đảm bảo tính chính xác và đáng tin cậy của dữ liệu, yêu cầu các quy trình làm sạch và xác minh hiệu quả.
- **Value (Giá trị):** Big Data mang tiềm năng kinh tế khổng lồ, và việc khai thác đúng cách sẽ giúp tạo ra những thông tin quý giá, thúc đẩy lợi ích kinh doanh và ra quyết định hiệu quả.

Các đặc điểm chính của Big Data được trình bày trong hình 1:

Hình 1: 5 đặc điểm chính của Big Data

1.3. Các nguồn dữ liệu lớn

Dữ liệu lớn có thể đến từ nhiều nguồn khác nhau như:

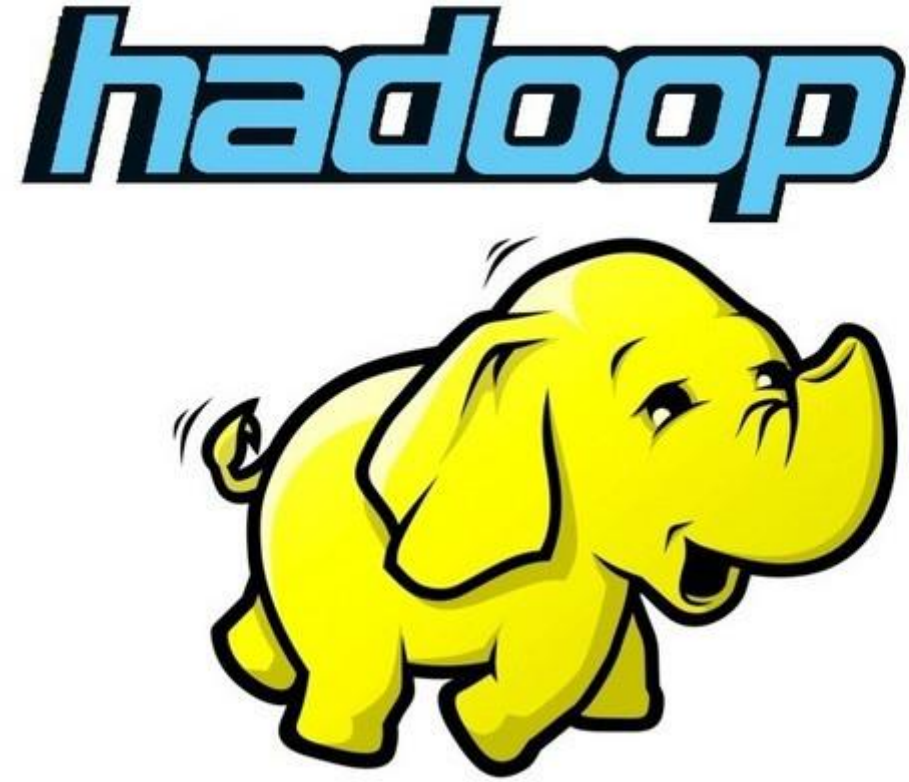
- Dữ liệu hành chính (phát sinh từ chương trình của một tổ chức, có thể là chính phủ hay phi chính phủ). Ví dụ: hồ sơ y tế điện tử ở bệnh viện, hồ sơ bảo hiểm, hồ sơ ngân hàng,...
- Dữ liệu từ hoạt động thương mại (phát sinh từ các giao dịch giữa hai thực thể). Ví dụ: các giao dịch thẻ tín dụng, giao dịch trên mạng, bao gồm cả từ các thiết bị di động
- Dữ liệu từ các thiết bị cảm biến như hình ảnh vệ tinh, cảm biến đường, cảm biến khí hậu
- Dữ liệu từ các thiết bị theo dõi, ví dụ theo dõi dữ liệu từ điện thoại di động, GPS
- Dữ liệu từ các hành vi, ví dụ như tìm kiếm trực tuyến về một sản phẩm, một dịch vụ hay bất kỳ loại thông tin khác, trang xem trực tuyến
- Dữ liệu từ các thông tin ý kiến trên các phương tiện thông tin xã hội.

Tổng quan các nguồn dữ liệu lớn được thể hiện ở hình 2:

Hình 2: Một số nguồn dữ liệu lớn

II. Giới thiệu về Hadoop

Hadoop là một công nghệ phân tán và mã nguồn mở được sử dụng phổ biến để xử lý và lưu trữ khối dữ liệu lớn trên các cụm máy tính phân tán, được thiết kế để xử lý và lưu trữ dữ liệu lớn một cách hiệu quả. Cung cấp khả năng xử lý dữ liệu lớn, mở rộng linh hoạt và chi phí thấp, làm cho nó trở thành một công nghệ không thể thiếu trong các hệ thống xử lý dữ liệu hiện đại. Nó được phát triển để giải quyết các thách thức trong lĩnh vực Big Data mà các công nghệ cũ không thể đáp ứng.



Hình 3: Logo của Hadoop

Hadoop có cấu trúc liên kết master-slave, với một node master và nhiều node slave. Node master gán tác vụ và quản lý tài nguyên, trong khi node slave lưu trữ dữ liệu thực.

Hadoop được xây dựng dựa trên ba phần chính là Hadoop Distributes FileSystem (HDFS), YARN và MapReduce. Trong đó:

- **HDFS (Hadoop Distributed File System):** HDFS là hệ thống file phân tán được sử dụng để lưu trữ dữ liệu. Nó phân chia dữ liệu thành các khối và phân tán chúng trên nhiều máy tính. HDFS có độ tin cậy cao và thích hợp cho việc lưu trữ khối lượng dữ liệu lớn.
- **YARN (Yet Another Resource Negotiator):** YARN là một khung phần mềm quản lý và phân bổ tài nguyên để vận hành các ứng dụng trên Hadoop. Nó cung

cấp các dịch vụ như quản lý tài nguyên, lên lịch và giám sát cho các ứng dụng chạy trên Hadoop.

- **MapReduce:** MapReduce là khung mô hình lập trình để xử lý và tính toán trên số lượng dữ liệu lớn trong môi trường phân tán. Nó hoạt động dựa trên 2 pha chính là Map và Reduce. Pha Map sẽ chia nhỏ dữ liệu thành các cặp key-value, sau đó pha Reduce sẽ nhóm các cặp key-value lại với nhau để tính toán kết quả cuối cùng.

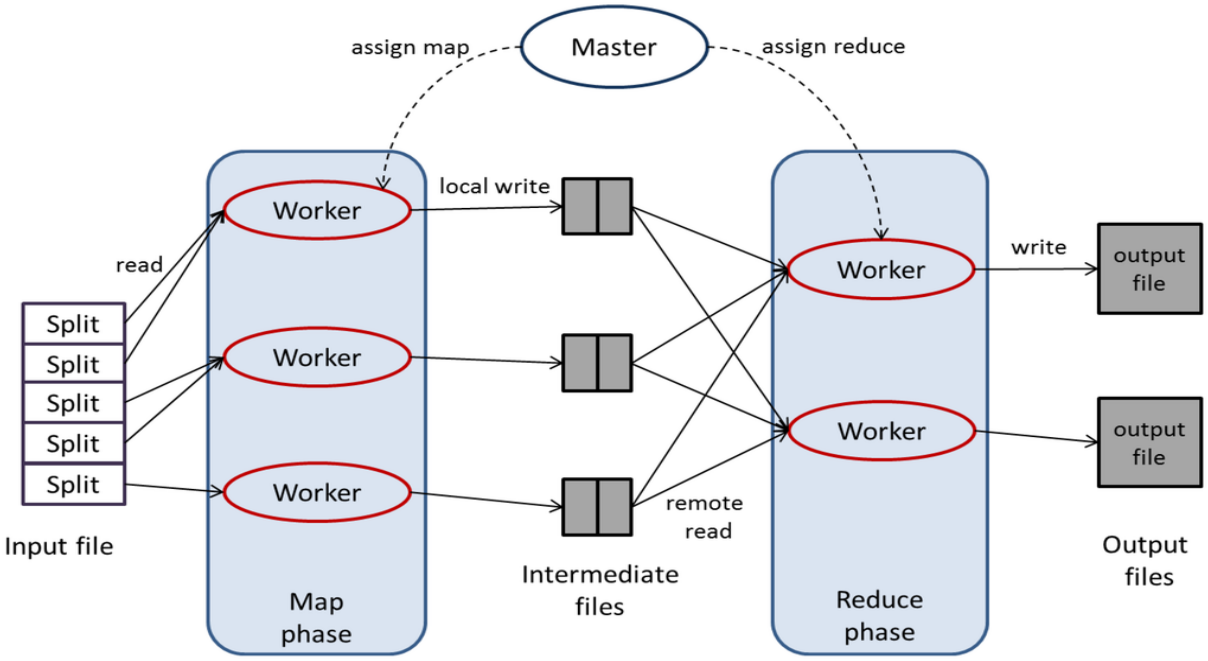
III. Giới thiệu về MapReduce

MapReduce là một framework dùng để viết các ứng dụng xử lý song song một lượng lớn dữ liệu có khả năng chịu lỗi cao xuyên suốt hàng ngàn cluster (cụm) máy tính.

MapReduce thực hiện 2 chức năng chính đó là:

- Map: Sẽ thực hiện đầu tiên, có chức năng tải, phân tích dữ liệu đầu vào và chuyển đổi thành tập dữ liệu theo cặp key/value.
- Reduce: Sẽ nhận kết quả đầu ra từ tác vụ Map, kết hợp dữ liệu lại với nhau thành tập dữ liệu nhỏ hơn, tạo ra kết quả cuối cùng.

Ví dụ về thực hiện một chương trình sử dụng MapReduce:



Hình 4: Thực hiện chương trình MapReduce

CHƯƠNG 2: XÂY DỰNG HỆ THỐNG GỢI Ý PHIM VỚI THUẬT TOÁN GỢI Ý SẢN PHẨM

I. Thuật toán Item-based Collaborative Filtering

1.1. Giới thiệu

Thuật toán Item-based Collaborative Filtering là một nhánh của Collaborative Filtering, tập trung vào việc phân tích mối quan hệ giữa các sản phẩm để tạo ra các đề xuất cho người dùng. Phương pháp này dựa trên quan điểm rằng nếu một người dùng thích hoặc quan tâm đến một sản phẩm, họ có khả năng thích các sản phẩm tương tự, dựa trên hành vi và sở thích của những người dùng khác trong cộng đồng.

Thay vì tìm kiếm người dùng có đặc điểm tương đồng, Item-based Collaborative Filtering tập trung vào việc khai thác sự tương đồng giữa các sản phẩm thông qua lịch sử tương tác của tất cả người dùng. Điều này giúp tạo ra các gợi ý sản phẩm một cách nhanh chóng và hiệu quả, đặc biệt trong các hệ thống có số lượng người dùng lớn.

1.2. Các bước thực hiện

Bước 1: Thu thập dữ liệu

- Thu thập lịch sử tương tác của người dùng với sản phẩm, bao gồm các đánh giá, lượt xem, hoặc hành động khác phản ánh sở thích cá nhân.

Bước 2: Trích xuất đặc trưng

- Xây dựng ma trận đánh giá người dùng và sản phẩm.

Bước 3: Tính toán độ tương đồng

- Đo lường mức độ tương đồng giữa các người dùng dựa trên ma trận đánh giá.
- Sử dụng các phương pháp phổ biến như **cosine similarity** hoặc **correlation coefficient** để xác định mối quan hệ giữa người dùng.

Bước 4: Đưa ra đề xuất

- Tính điểm đề xuất: Dựa trên độ tương đồng giữa các sản phẩm, ước lượng mức độ yêu thích của một người dùng đối với các sản phẩm chưa tương tác.

$$predicted\ rating(u,i) = \frac{\sum_{j \in I}(similarity(i,j).(rating(u,j))}{\sum_{j \in I}|similarity(i,j)|}$$

Với I là tập các sản phẩm đã được người dùng u đánh giá

- Xếp hạng sản phẩm: Sắp xếp các sản phẩm theo điểm đề xuất, từ cao đến thấp.

- Cung cấp gợi ý: Đưa ra danh sách các sản phẩm phù hợp nhất với sở thích cá nhân của từng người dùng.

1.3. Sử dụng MapReduce cho Item Collaborative Filtering

1.3.1. Giai đoạn Map

Xử lý dữ liệu lịch sử tương tác người dùng: Tạo ra các cặp khóa - giá trị, trong đó khóa là ID người dùng, giá trị là (ID sản phẩm, đánh giá)

1.3.2. Giai đoạn Shuffle và Sort

Sắp xếp các cặp khóa - giá trị và nhóm lại dựa trên khóa

1.3.3: Giai đoạn Reduce

Bước 1: Tạo một ma trận đánh giá người dùng - sản phẩm dựa trên dữ liệu nhóm lại từ giai đoạn trước.

Bước 2: So sánh vector đánh giá giữa các người dùng để tính toán độ tương đồng, sử dụng các phương pháp như **cosine similarity** hoặc **Pearson correlation coefficient**.

Bước 3: Dựa trên độ tương đồng và đánh giá của người dùng tương tự, ước tính mức độ yêu thích của một người dùng đối với các sản phẩm chưa được đánh giá.

Bước 4: Sắp xếp các sản phẩm theo điểm đề xuất, từ cao đến thấp.

Bước 5: Xuất cặp khóa - giá trị : user_id - danh sách gợi ý.

II. Ví dụ minh họa cho thuật toán

Để minh họa cho quá trình hoạt động của thuật toán Collaborative Filtering dựa trên Item, chúng ta xem xét một ví dụ đơn giản:

Dữ liệu đầu vào: Ma trận người dùng - sản phẩm với các đánh giá từ 1 - 5. (? là sản phẩm mà người dùng đó chưa đánh giá)

	Sản phẩm 1	**Sản phẩm 2**	**Sản phẩm 3**	**Sản phẩm 4**
Người dùng 1	3	3	?	4
Người dùng 2	3	?	5	2
Người dùng 3	5	?	4	?
Người dùng 4	?	3	?	1

Bước 1: Xây dựng ma trận đồng xuất hiện

Dựa trên dữ liệu đầu vào, chúng ta xây dựng ma trận đồng xuất hiện C như sau: (Ví dụ ta thấy rằng có 2 người cùng đánh giá sản phẩm 1 và 3 là Người dùng 2 và 3 nên C[1][3] = C[3][1] = 2)

	Sản phẩm 1	**Sản phẩm 2**	**Sản phẩm 3**	**Sản phẩm 4**
Sản phẩm 1	3	1	2	2
Sản phẩm 2	1	2	0	2
Sản phẩm 3	2	0	2	1
Sản phẩm 4	2	2	1	3

Bước 2: Chuẩn hóa ma trận

Dựa trên ma trận C, chúng ta chuẩn hóa thành ma trận S như sau: (Ví dụ: S[1][1] = $\frac{C[1][1]}{Tổng\ cột\ 1} = \frac{3}{8} = 0.375$)

	Sản phẩm 1	**Sản phẩm 2**	**Sản phẩm 3**	**Sản phẩm 4**
Sản phẩm 1	0.375	0.2	0.4	0.25
Sản phẩm 2	0.125	0.4	0	0.25
Sản phẩm 3	0.25	0	0.4	0.125
Sản phẩm 4	0.25	0.4	0.2	0.375

Bước 3: Xây dựng ma trận đánh giá

Dựa trên ma trận S và ma trận đầu vào U, ta tính được ma trận R như dưới đây (Ví dụ ta tính đánh giá của người dùng 1 cho sản phẩm 3 là R[1][3] = 3 × 0.4 + 3 × 0 + 0 × 0.4 + 4 × 0.2 = 2.0):

	Sản phẩm 1	**Sản phẩm 2**	**Sản phẩm 3**	**Sản phẩm 4**
Người dùng 1	2.5	3.4	2.0	3.0
Người dùng 2	2.875	1.4	3.6	2.125
Người dùng 3	2.875	1.0	3.6	1.75
Người dùng 4	0.625	1.6	0.2	1.125

III. Item-based Collaborative Filtering với Hadoop MapReduce

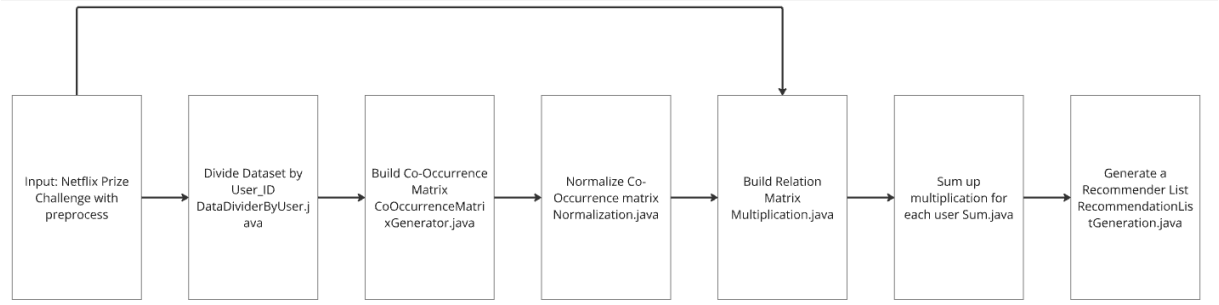
3.1. Ý tưởng MapReduce Item-based Collaborative Filtering

- **Nhiệm vụ:** MapReduce chia bài toán thành các bước nhỏ gọn (Map và Reduce), xử lý dữ liệu theo cách song song, tận dụng sức mạnh của hệ thống phân tán.
- **Ý tưởng chính:** Phân tán quá trình tính toán ma trận đồng xuất hiện và ma trận đánh giá trên nhiều máy tính, từ đó:
    - Rút ngắn thời gian xử lý: Nhờ việc chia nhỏ dữ liệu và xử lý đồng thời.

- Tối ưu hiệu suất: Giảm tải cho từng máy và tăng khả năng xử lý dữ liệu lớn.
- Giải quyết thách thức: Đối phó với kích thước dữ liệu khổng lồ và độ phức tạp trong tính toán khi xây dựng hệ thống gợi ý cá nhân hóa.

- **Kết quả:** Sử dụng MapReduce không chỉ tăng tốc độ xử lý mà còn đảm bảo khả năng mở rộng của hệ thống, giúp xây dựng các hệ thống khuyến nghị hiệu quả ngay cả với tập dữ liệu khổng lồ.

3.2. Sơ đồ luồng hoạt động



Hình 5: Sơ đồ luồng hoạt động của Item-based Collaborative Filtering MapReduce

3.3. Triển khai thuật toán

- **Dữ liệu đầu vào:** Là danh sách các hàng lưu dưới dạng file .txt. Mỗi hàng chứa thông tin về người dùng, sản phẩm và đánh giá của người dùng đó cho sản phẩm đó, cách nhau bởi dấu phẩy, được chuyển sang kiểu key-value làm đầu vào cho thuật toán
- **Triển khai:**
  - Biểu diễn dữ liệu: Dữ liệu lưu trữ dưới dạng list các hàng. Mỗi hàng chứa thông tin về người dùng, sản phẩm và đánh giá của người dùng đó cho sản phẩm đó, cách nhau bởi dấu phẩy.
  - Lưu trữ phân tán dữ liệu: Dữ liệu được chia thành các phần nhỏ và lưu trữ trên nhiều máy tính khác nhau.
  - Trên mỗi máy tính, trong mỗi lần lặp thực hiện đọc vào từng dòng, gửi lại kết quả cho reducer để tính độ lợi thông tin của từng thuộc tính trong từng phần dữ liệu.
  - Thực hiện gọi đệ quy xác định nút gốc và nút lá tương ứng. Cập nhật lại nút, cho đến khi đạt hội tụ sau mỗi vòng lặp.

Dữ liệu cần phân lớp: Là danh sách các hàng lưu trên file .txt. được chuyển sang kiểu key/value làm đầu ra cho thuật toán.

- Mô hình cơ bản của MapReduce:
  - Map (KeyIn, ValIn) → List(KeyInt, ValInt)

    - Reduce (KeyInt, List(ValInt)) → List(KeyOut, ValOut)
- Áp dụng cho thuật toán Item-based Collaborative Filtering:
    - Xây dựng lớp DataDividedByUser
    - Xây dựng lớp CoOccurrenceMatrixGenerator
    - Xây dựng lớp Normalize
    - Xây dựng lớp Multiplication
    - Xây dựng lớp Sum
    - Xây dựng lớp RecommenderListGenerator

Bước 1: Xây dựng lớp DataDividerByUser

- Mục đích: Nhóm dữ liệu theo người dùng
- Mapper:
    - Input: user, movie, rating
    - Output: key là user, value là movie:rating
- Reducer:
    - Input: key là user, value là movie:rating
    - Output: user movie1:rating1, movie2:rating2, ...

Bước 2: Xây dựng lớp CoOccurrenceMatrixGenerator

Mục đích: Đếm số lần mỗi cặp movie được đánh giá bởi cùng 1 người

- Mapper:
    - Input: user movie1:rating1, movie2:rating2, ...
    - Output: key là movie1:movie2, value là 1
- Reducer:
    - Input: key là movie1:movie2, value là 1
    - Output: movie1:movie2 count (trong đó count là số lần cặp movie đó được đánh giá bởi cùng 1 người)

Bước 3: Xây dựng lớp Normalize

Mục đích: Chuẩn hóa từng đơn vị của ma trận đồng xuất hiện bằng cách chia giá trị đó cho tổng của cột tương ứng trong ma trận

- Mapper:
    - Input: movie1:movie2 count
    - Output: key là movie1, value là movie2:count
- Reducer:
    - Input: key là movie1, value là movie2:count
    - Output: movie1 movie2 = relation (với relation là giá trị sau khi xử lý)

Bước 4: Xây dựng lớp Multiplication

Mục đích: Nhân relation với rating tương ứng với mỗi key là movie

- CooccurrenceMapper:
  - Input: movie1 movie2 = relation
  - Output: key là movie1, value là movie2 = relation
- RatingMapper:
  - Input: user, movie, rating
  - Output: key là movie, value là user:rating
- Reducer:
  - Input1: key là movie1, value là movie2 = relation
  - Input2: key là movie, value là user:rating
  - Output: user:movie result (với result = relation × rating)

Bước 5: Xây dựng lớp Sum

Mục đích: Tính tổng các result theo key là user:movie

- Mapper:
  - Input: user:movie result
  - Output: key là user:movie, value là result
- Reducer:
  - Input: key là user:movie, value là result
  - Output: user:movie sum (với sum là tổng các result)

Bước 6: Xây dựng lớp RecommenderListGenerator

Mục đích: Lấy ra 5 movie có giá trị sum lớn nhất với mỗi user được sắp xếp giảm dần

- Mapper
  - Input: user:movie sum
  - Output: key là user, value là movie:sum
- Reducer
  - Input: key là user, value là movie:sum
  - Output: user movie:sum

CHƯƠNG 3: THỬ NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ

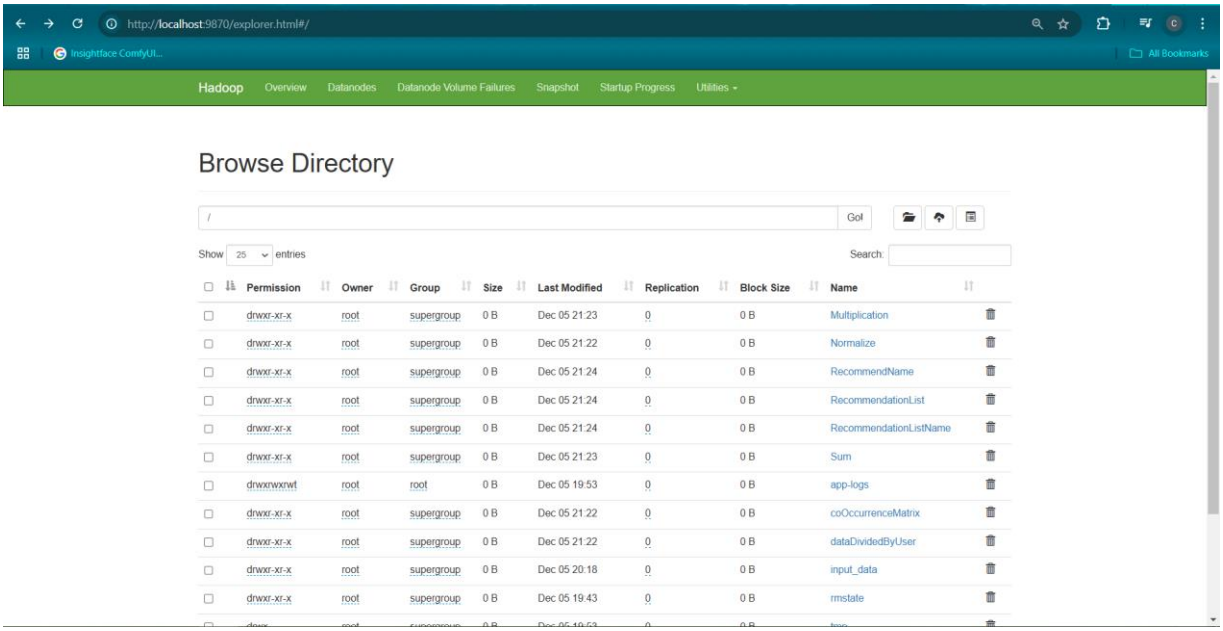
I. Dữ liệu

Sử dụng bộ dữ liệu trên Kaggle có tên Netflix Prize gồm hơn 100 triệu lượt đánh giá từ 480 nghìn khách hàng cho 17.000 phim. Bộ dữ liệu có 4 trường: movie_id, user_id, rating, date.

II. Kết quả

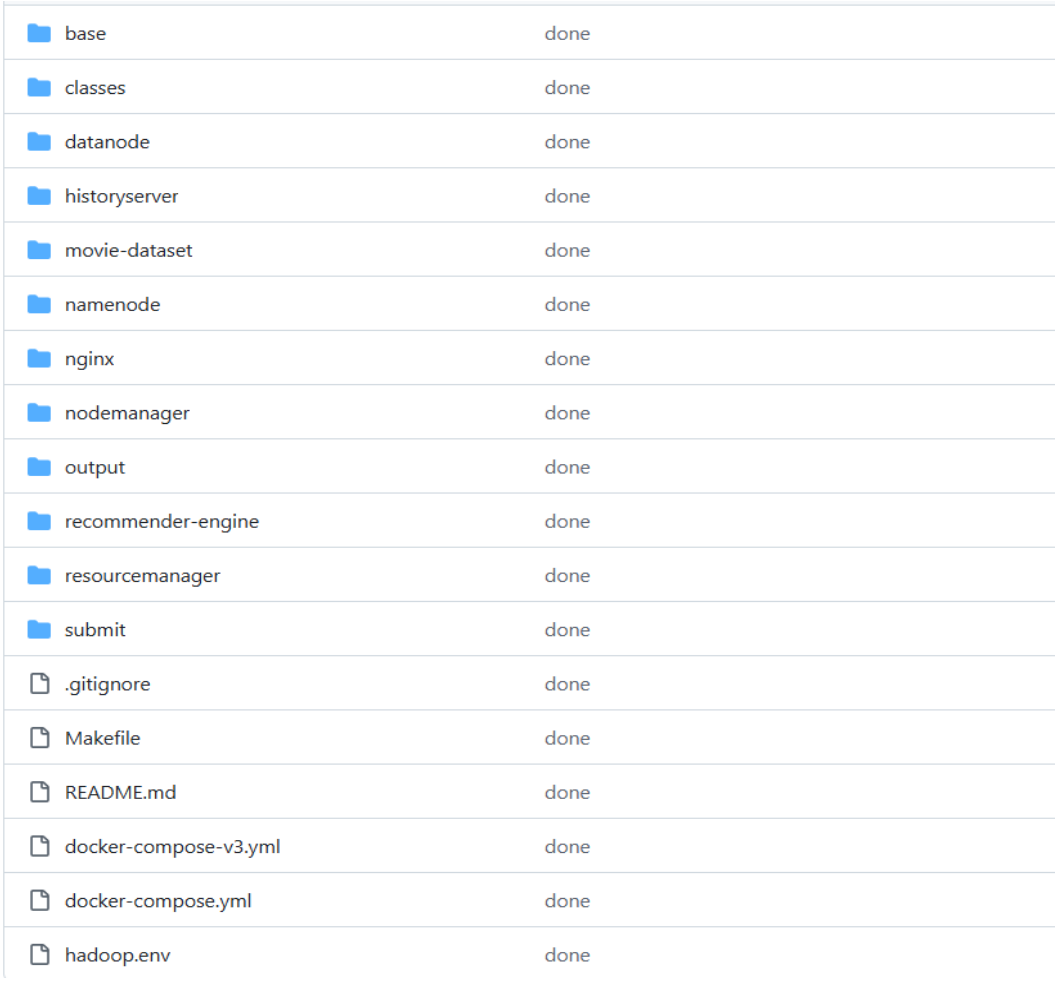
2.1. Cài đặt thuật toán với Hadoop thành công

Nhóm đã sử dụng Docker để cài đặt Hadoop vì tính năng đóng gói của Docker giúp quá trình cài đặt trở nên nhanh chóng và dễ dàng hơn rất nhiều.

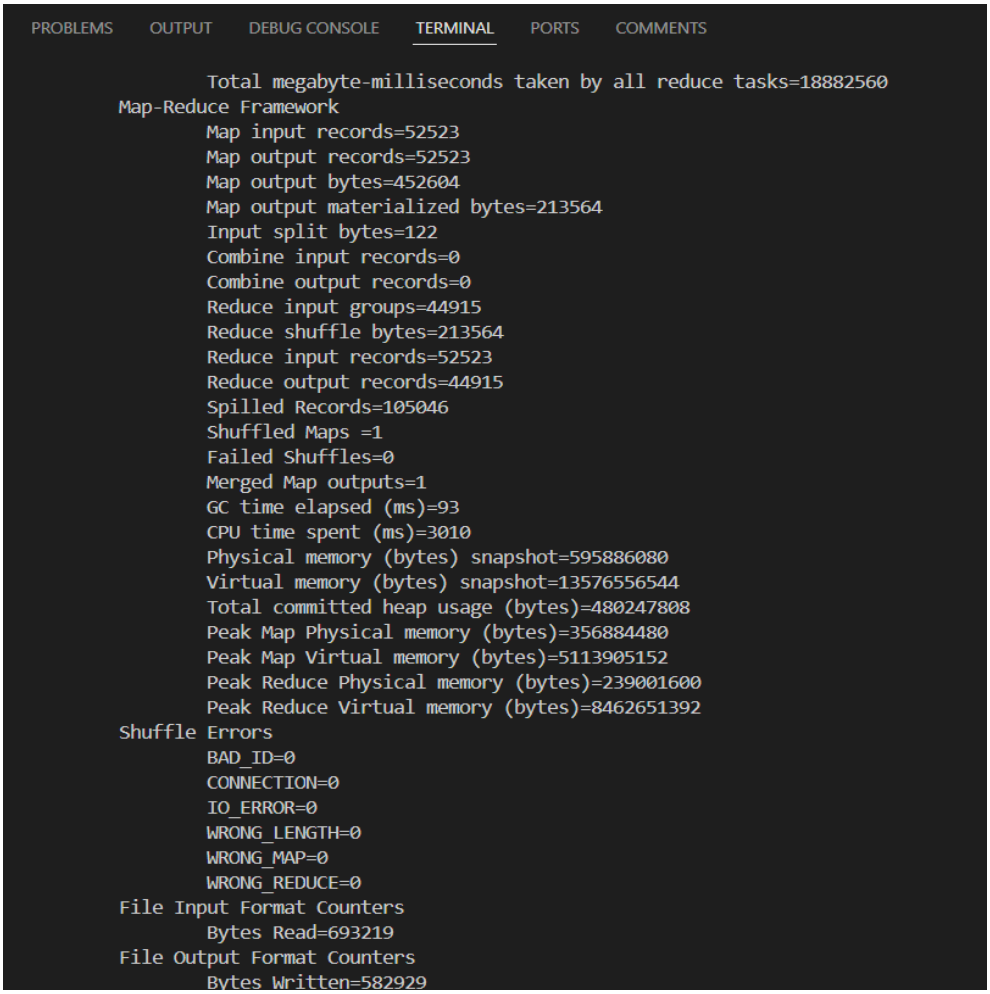


Hình 6: Kết quả chạy thuật toán Item-based Collaborative Filtering với Hadoop MapReduce

2.2. Demo

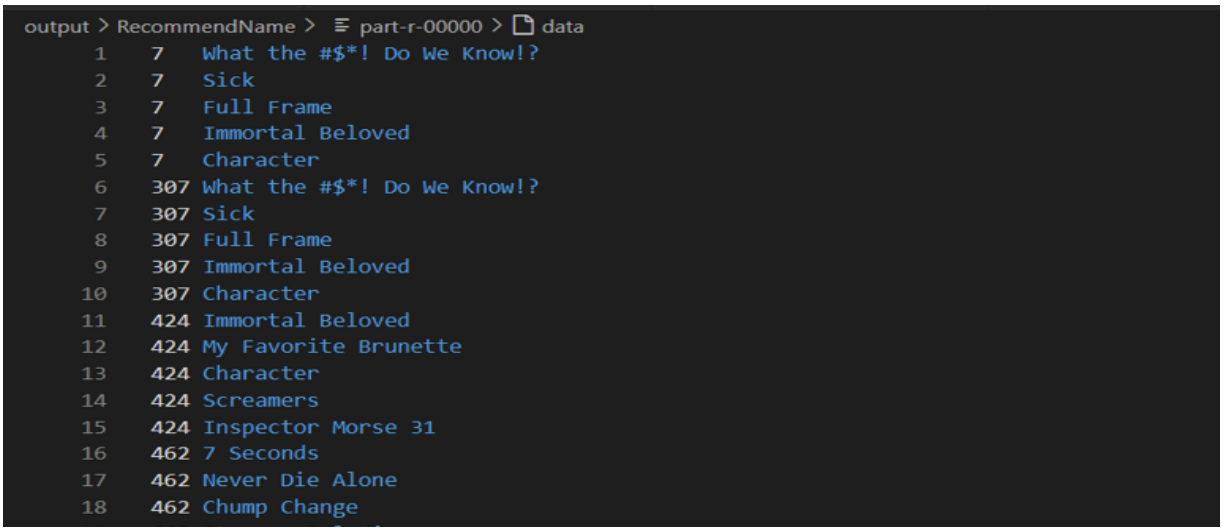


Hình 7: Tổng quan về các file của repository



Hình 8: Terminal sau khi chạy chương trình

Kết quả:



Hình 9: Kết quả sau khi chạy thuật toán

2.3. Phân tích và đánh giá

Ưu điểm:

- Xử lý dữ liệu lớn nhanh chóng nhờ khả năng tính toán song song trên nhiều node trong hệ thống Hadoop.
- Có thể mở rộng dễ dàng khi dữ liệu và số lượng người dùng tăng lên, giúp hệ thống duy trì hiệu suất ổn định.
- Hệ thống đưa ra các gợi ý chính xác dựa trên sự đồng xuất hiện giữa các sản phẩm được người dùng đánh giá tương tự.
- Mã nguồn có thể mở rộng dễ dàng, cho phép thay đổi và cải tiến thuật toán mà không làm gián đoạn hệ thống.
- Dữ liệu được phân tán trên nhiều máy, giúp tăng tính chịu lỗi và khả năng xử lý dữ liệu khối lượng lớn mà không bị gián đoạn.
- Việc chia nhỏ công việc và tính toán song song giúp giảm đáng kể thời gian xử lý và nâng cao hiệu suất.

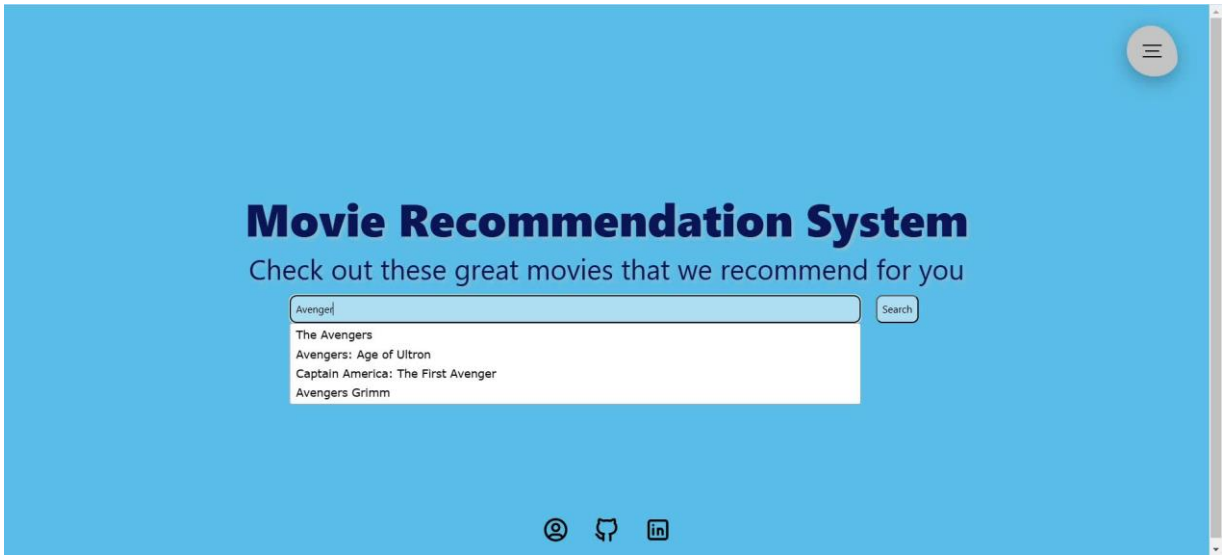
Hạn chế:

- Nếu dữ liệu nhỏ hoặc yêu cầu tính toán không phức tạp, việc sử dụng Hadoop có thể gây lãng phí tài nguyên và làm hệ thống trở nên phức tạp.
- Cấu trúc phân tán và tính toán song song có thể làm tăng độ phức tạp trong việc bảo trì và giám sát hệ thống.

2.4. Demo sản phẩm

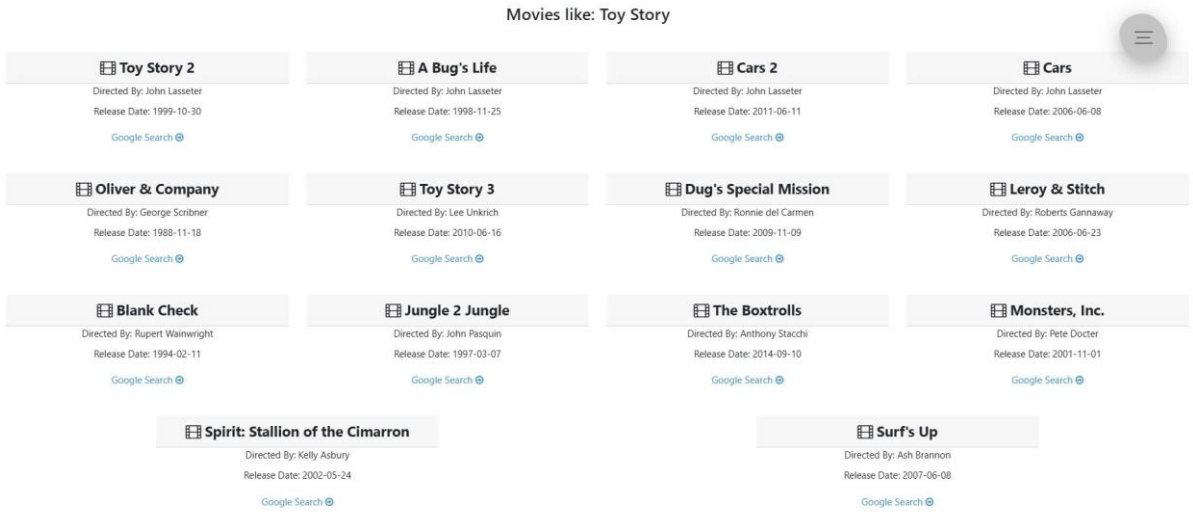
Link tới website: [Movie Recommender System]

Nhóm đã tạo 1 website để gợi ý phim với giao diện như bên dưới:



Hình 11: Giao diện cơ bản

Từ giao diện này ta có thể tìm kiếm các phim tương tự bằng cách nhập tên phim vào ô tìm kiếm sau đó nhấn nút “Search”. Kết quả trả về thu được sẽ là:



Hình 12: Kết quả chạy demo recommendation

Ở đây kết quả trả về sẽ gồm: tên phim, tên đạo diễn, ngày công chiếu và đường link có thể bấm vào để tự động tìm kiếm phim đó trên Google.

Hình 13: Thông tin và liên lạc về nhóm phát triển

CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

I. Kết luận

Big data mang đến cho các tổ chức và doanh nghiệp không chỉ cơ hội mà còn những thách thức và tài nguyên quý giá. Để giải quyết những vấn đề này, mô hình MapReduce đóng vai trò quan trọng, chia công việc xử lý thành những khối nhỏ, phân tán chúng qua các nút tính toán, và sau đó thu thập kết quả một cách hiệu quả. Trong đề tài này, chúng em đã áp dụng mô hình MapReduce trên nền tảng Hadoop – một framework mã nguồn mở – để xây dựng hệ thống gợi ý phim dựa trên thuật toán Item-based Collaborative Filtering.

Hoàn thành dự án **"Hệ thống gợi ý phim dựa trên Item Collaborative Filtering & Hadoop Mapreduce"**, nhóm em đã đạt được một số kết quả sau:

- Nắm vững khái niệm về Big Data, Hadoop và MapReduce.
- Hiểu rõ về thuật toán Item-based Collaborative Filtering và cách thức hoạt động của nó.
- Triển khai thành công ý tưởng và giải pháp sử dụng Hadoop MapReduce trong việc thực hiện thuật toán Item-based Collaborative Filtering.
- Xây dựng sơ đồ luồng hoạt động thuật toán và triển khai chương trình demo thành công.
- Đánh giá và kiểm tra hiệu quả của chương trình.

II. Hướng phát triển

- Áp dụng kiến thức về Big data, apache hadoop, cải tiến và xây dựng ứng dụng phân tích dữ liệu lớn hơn và vào nhiều lĩnh vực khác.
- Trong quá trình hoàn thành bài tập lớn, nhóm em đã cố gắng tìm hiểu và tham khảo các tài liệu liên quan.
- Tuy nhiên, thời gian có hạn nên chúng em sẽ không tránh khỏi những thiếu sót, rất mong nhận được sự đóng góp ý kiến của thầy và các bạn để báo cáo và kỹ năng của chúng em ngày được hoàn thiện hơn.

TÀI LIỆU THAM KHẢO

[1] https://github.com/thviet79/Bigdata_Project_Recommender_System

[2] Content-Based Recommendations Machine Learning cơ bản

[3] Collaborative Filtering Neighborhood-Based Machine Learning cơ bản

[4] https://hadoop.apache.org/

[5] https://github.com/coffee183/Movie-Recommendation-System

[6] Hadoop – Apache Hadoop 3.3.6