TRƯỜNG ĐẠI HỌC THỦY LỢI

KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO MÔN HỌC PHÂN TÍCH DỮ LIỆU LỚN

ĐỀ TÀI

Phân tích dữ liệu khách hàng của cửa hàng bán lẻ trực tuyến sử dụng thuật toán Kmeans trên Hadoop

Nhóm sinh viên thực hiện: 07

1. Đinh Tất Hiển - 1851061360
2. Phạm Hoàng Minh-1851061718
3. Đoàn Hữu Mạnh-1851061960
4. Dương Minh Tiến-175A071296

Giảng viên hướng dẫn: TS. Tạ Quang Chiểu

HÀ NỘI, 05/2022

MỤC LỤC

MỞ ĐẦU

Công nghệ big data đã đạt đến đỉnh cao trong việc thực hiện các chức năng của nó. Trong tháng 8/2015 big data đã vượt ra khỏi bảng xếp hạng những công nghệ mới nổi Cycle Hype của Gartner và tạo ra một tiếng vang lớn cho *xu hướng công nghệ* của thế giới. Big data chứa trong mình rất nhiều thông tin quý giá mà nếu mà trích xuất thành công, nó sẽ giúp rất nhiều trong nhiều lĩnh vực như y tế, giao thông, giáo dục, …

Chính vì thế những framework giúp việc xử lý BIGDATA cũng đang ngày càng được xử lý và phát triển mạnh. Một trong những công nghệ cốt lõi cho việc lưu trữ và truy cập số lượng lớn dữ liệu là Hadoop - một framework giúp lưu trữ và xử lý Big data áp dụng MapReduce.

Từ đó, chúng em đã chọn đề tài: "**Phân tích dữ liệu khách hàng của cửa hàng bán lẻ trực tuyến sử dụng thuật toán Kmeans trên Hadoop**" để làm báo kết thúc môn học của mình.

Báo cáo gồm 4 chương:

Chương 1: Tổng quan về dữ liệu lớn.

Chương 2: Phân cụm dữ liệu bằng thuật toán Kmeans.

Chương 3: MapReduce thuật toán Kmeans trong phân cụm dữ liệu.

Chương 4: Kết luận và hướng phát triển.

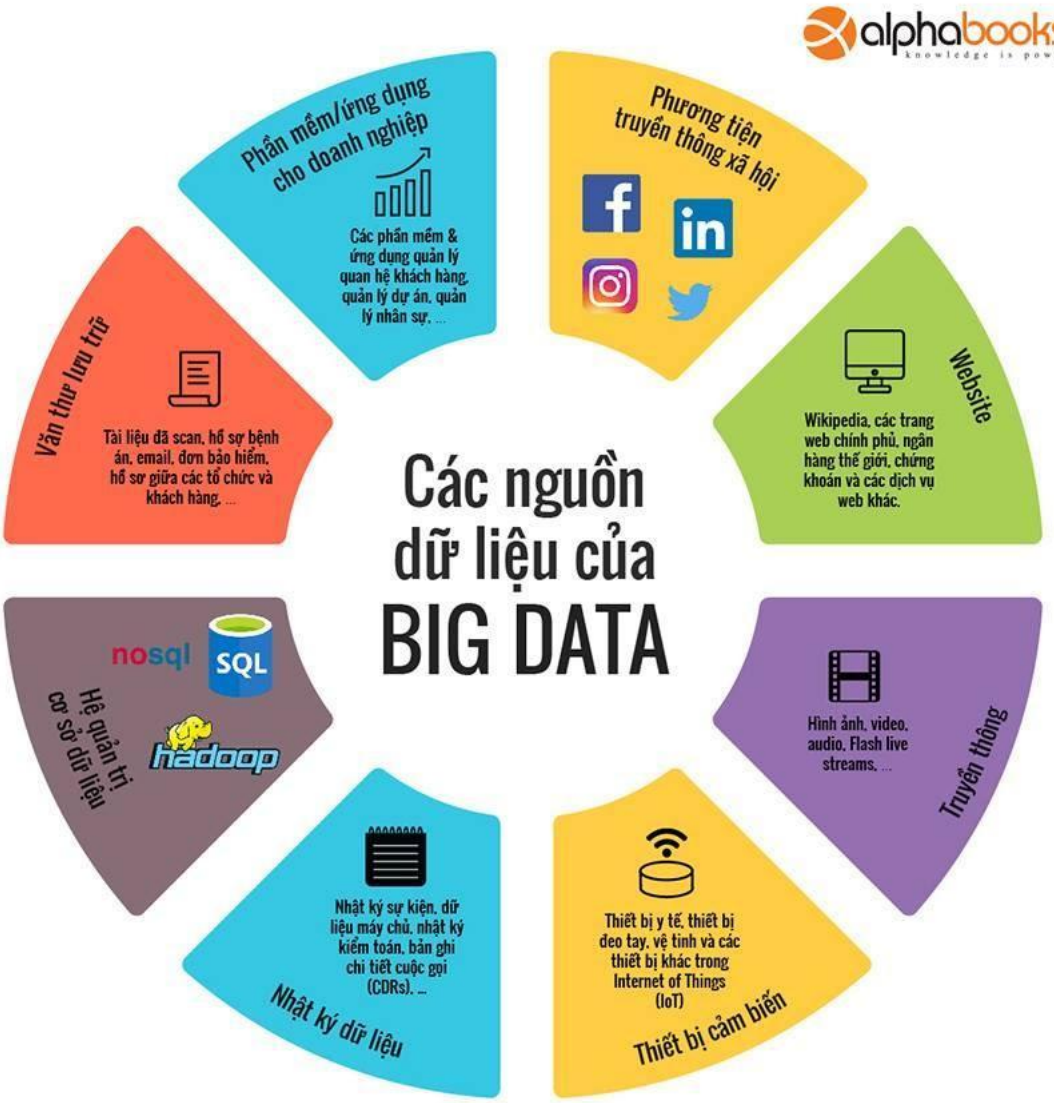
CHƯƠNG 1: TỔNG QUAN VỀ DỮ LIỆU LỚN

1.1 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.

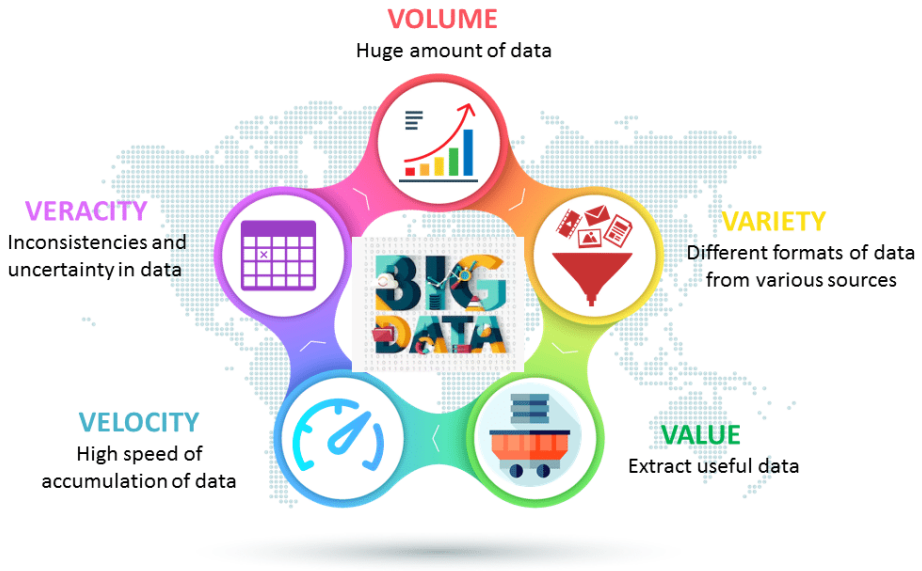
Theo Gartner : Dữ liệu lớn là những nguồn thông tin có đặc điểm chung khối lượng lớn, tốc độ nhanh và dữ liệu định dạng dưới nhiều hình thức khác nhau,do đó muốn khai thác được phải đòi hỏi phải có hình thức mới để đưa ra quyết định khám phá và tối ưu hóa quy trình.

Dữ liệu đến từ rất nhiều nguồn khác nhau:



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 dúng và hợp lý hơn.

1.2 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 trong 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 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.

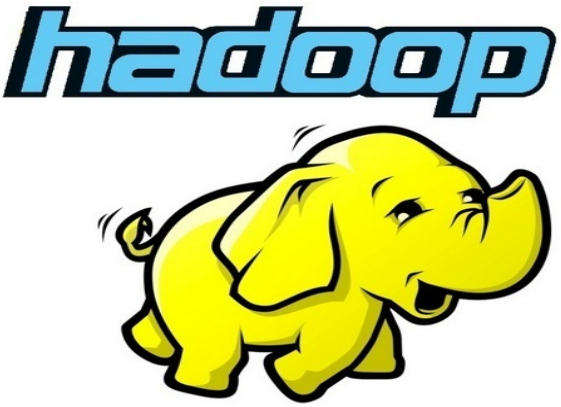
1.3 Ứng dụng của dữ liệu lớn.

Dữ liệu lớn đã được ứng dụng trong nhiều lĩnh vực:

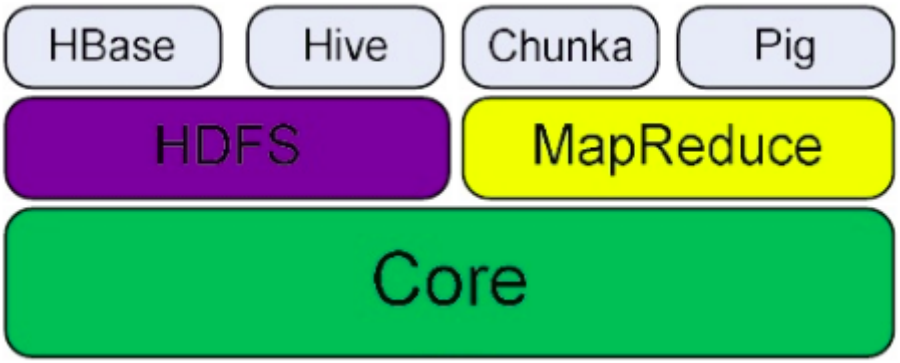
- Hoạt động chính trị
- Giao thông
- Y tế
- Thể thao
- Tài chính
- Thương mại
- Thống kê...

1.4 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.



+ Các thành phần của hadoop: Core, MapReduce engine, HDFS, HBase, Hive, Pig, Chukwa,.. Tuy nhiên *tập trung* vào 2 thành phần quan trọng nhất: HDFS và MapReduce.



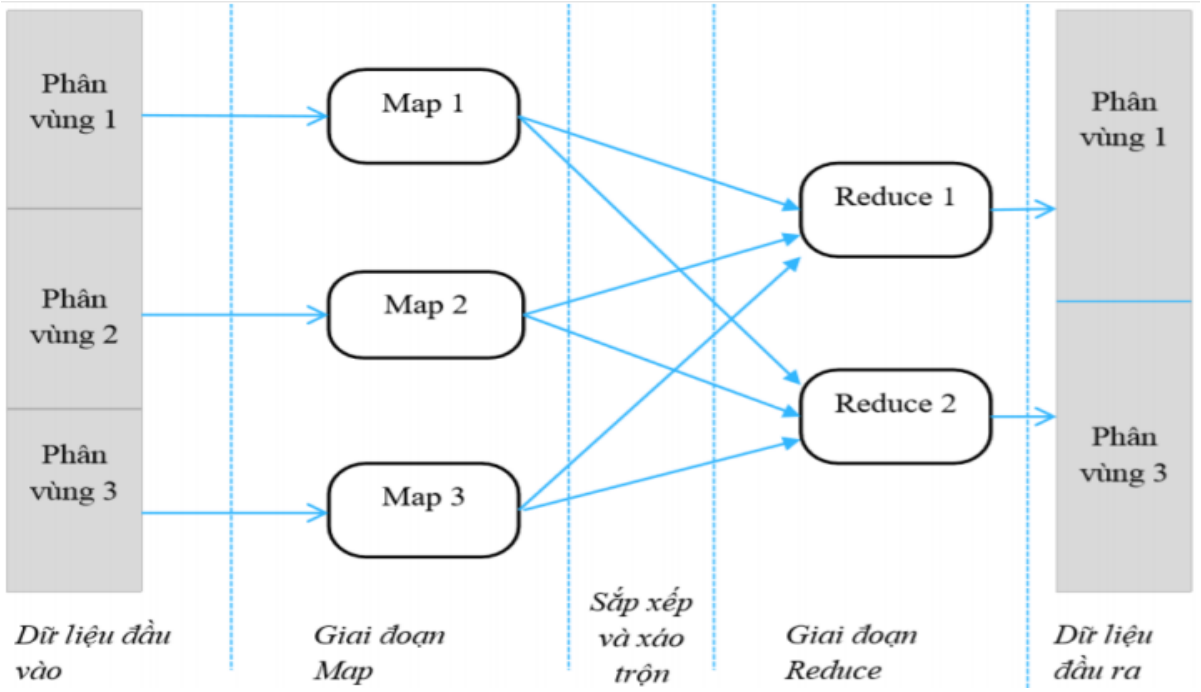
+ Hadoop hiện thực mô hình Map/Reduce, đây là mô hình mà ứng dụng sẽ được chia nhỏ ra thành nhiều phân đoạn khác nhau, và các phần này sẽ được chạy song song trên nhiều node khác nhau.

+ Thêm vào đó, Hadoop cung cấp 1 hệ thống file phân tán (HDFS) cho phép lưu trữ dữ liệu lên 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 các node.

=> *Kết luận:* Là một framework cho phép phát triển các ứng dụng phân tán. Viết bằng java.

- ***Tổng quan về MapReduce:***

MapReduce là mô hình lập trình được sử dụng để tính toán tập dữ liệu lớn. Một tiến trình xử lý MapReduce cơ bản có thể tính toán đến terabytes hoặc petabyte dữ liệu trên hệ thống được kết nối thành cụm các nodes. Dữ liệu được chia thành các mảnh nhỏ rồi đưa vào các nodes độc lập, vì vậy số lượng và kích thước của các mảnh phụ thuộc vào số nodes được kết nối trong mạng.



Các bước Map và Reduce được thiết kế tách biệt, riêng rẻ và hoàn toàn độc lập. Mỗi bước Map và Reduce được thực hiện song song trên các cặp dữ liệu (key, value). Do đó, chương trình được chia thành hai giai đoạn riêng biệt là Map và Reduce. Bộ ánh xạ (Mapper): xử lý tập dữ liệu đầu vào dưới dạng (key, value) và tạo ra tập dữ liệu trung gian là cặp (key, value). Tập dữ liệu này được tổ chức cho hoạt động của Reduce. Bộ ánh xạ Map thực hiện các bước như sau:

- Bước 1: Ánh xạ cho mỗi nhóm dữ liệu đầu vào dưới dạng (key, value).
- Bước 2: Thực thi việc Map, xử lý cặp (key, value) để tạo (key, value) mới, công việc này được gọi là chia nhóm, tức là tạo các giá trị liên quan tương ứng với cùng các từ khóa.
- Bước 3: Đầu ra của bộ ánh xạ được lưu trữ và định vị cho mỗi bộ Reducer. Tổng các khối và số công việc reduce là như nhau.

Bộ Reducer: Trộn tất cả các phần tử value có cùng key trong tập dữ liệu trung gian do Map tạo ra để tạo thành tập trị nhỏ hơn và quá trình này được lặp lại cho tất cả các giá trị key khác nhau. Việc truyền dữ liệu được thực hiện giữa Map và Reduce.

Lập trình viên cần cài đặt chính xác (key, value), MapReduce sẽ gom cụm một cách tự động và chính xác các values với cùng key. Các bước của bộ Reducer:

- Bước 1 (Trộn): Đầu vào của Reducer là đầu ra của Mapper. Giai đoạn này, MapReduce sẽ gán khối liên quan cho bộ Reducer.
- Bước 2 (Sắp xếp): Giai đoạn này, đầu vào của bộ Reducer được gom theo key (do đầu ra của bộ ánh xạ khác nhau có thể có cùng key). Hai giai đoạn Trộn và sắp xếp được đồng bộ hóa.
- Bước 3: Tạo bộ so sánh để nhóm các keys trung gian lần hai nếu quy luật gom nhóm khác với trước đó.

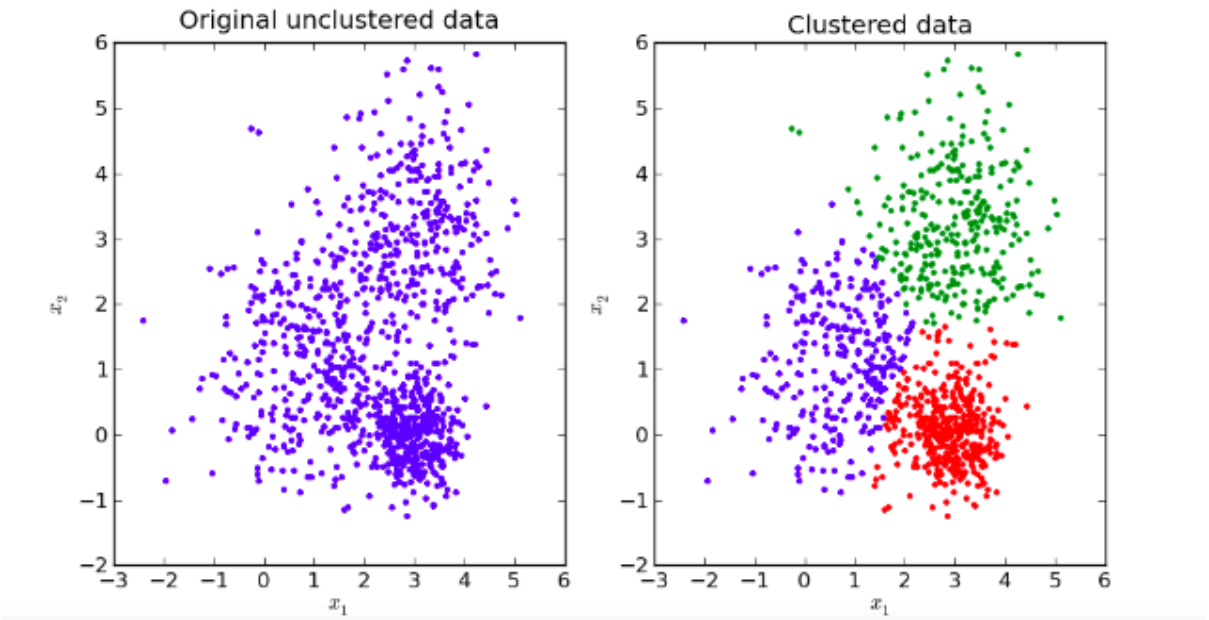
Trong một tiến trình của MapReduce, các công việc Map chạy song song, các công việc Reduce chạy song song. Tuy nhiên, các công việc Map và Reduce được thực hiện tuần tự, tức là Map phải hoàn thành và chuyển dữ liệu cho Reduce. Dữ liệu đầu vào và đầu ra được lưu trữ trong hệ thống file

CHƯƠNG 2: PHÂN CỤM DỮ LIỆU BẰNG THUẬT TOÁN KMEANS

2.1 Giới thiệu về kỹ thuật phân cụm

2.1.1 Khái niệm

Phân cụm dữ liệu là bài toán gom nhóm các đối tượng dữ liệu vào thành từng cụm (cluster) sao cho các đối tượng trong cùng một cụm có sự tương đồng theo một tiêu chí nào đó.



Đặc điểm

- Số cụm dữ liệu không được biết trước
- Có nhiều cách tiếp cận, mỗi cách lại có vài kỹ thuật
- Các kỹ thuật khác nhau thường mang lại kết quả khác nhau.

Các độ đo khoảng cách

Tính chất của độ đo khoảng cách:

- Tính không âm (non-negative): d(x,y) >=0 và d(x, y) = 0 khi và chỉ khi x trùng y.
- Tính đối xứng (symmetric): d(x, y) = d(y, x)
- Tính tam giác (triangle inequality): d(x, y) + d(y, z) >= d(x, z)

Độ đo Euclid chuẩn và độ đo Manhattan

- Cho hai điểm x = (x1, x2, ..., xm) và y = (y1, y2, ..., ym)
- Độ đo Euclid được xác định theo công thức

$$d(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^{m} |x_i - y_i|^r \right)^{\frac{1}{r}}$$

- Độ đo Euclid chuẩn (r = 2)

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{m} (x_i - y_i)^2}$$

- Độ đo Manhattan

$$d(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{m} |x_i - y_i|$$

Độ đo Cosine

- Cho hai vectơ x = (x1, x2, ..., xm) và y = (y1, y2, ..., ym)
- Độ đo Cosine được tính như sau
- Trong không gian dương:
    - Thoả mãn cả 3 tính chất
    - Giá trị nằm trong khoảng [0, 1]

Độ đo Hamming

- Được sử dụng khi các vectơ ở dạng logic (true/false, 0/1)
- Khoảng cách giữa hai vectơ được xác định là số chiều mà ở đó các giá trị tương ứng của hai vectơ là khác nhau.
- Thỏa mãn cả 3 tính chất
- VD: v1(0, 1, 0, 1, 0) và v2 (1, 1, 0, 1, 0) vậy d(v1, v2) = 1

Độ đo Jaccard

- x, y là hai tập hợp
- Chỉ số Jaccard

$$J(\mathbf{x}, \mathbf{y}) = \frac{|\mathbf{x} \cap \mathbf{y}|}{|\mathbf{x} \cup \mathbf{y}|}$$

- Độ đo Jaccard

$$d(\mathbf{x}, \mathbf{y}) = 1 - J(\mathbf{x}, \mathbf{y})$$

- Thỏa mãn cả 3 tính chất

Độ đo Kullback-Leibler (KL)

- Cho x = (x1, x2, ..., xm) và y = (y1, y2, ..., ym) là hai phân phối xác xuất rời rạc.
- Độ đo KL được tính như sau:

$$D_{KL}(\mathbf{x}||\mathbf{y}) = \sum_{i=1}^{m} x_i log \frac{x_i}{y_i}$$

Trong đó không xét những vị trí có xi = 0 hoặc yi = 0.

- KL không thoải mãn tính chất đối xứng, tức DKL(x||y) có thể khác DKL(y||x)
- Đó đó, có thể tính độ đo dựa trên KL như sau:

$$d(\mathbf{x},\mathbf{y}) = \frac{D_{KL}(\mathbf{x}||\mathbf{y}) + D_{KL}(\mathbf{y}||\mathbf{x})}{2}$$

2.1.2 Phương pháp nghiên cứu

Trong bài báo này chúng tôi sử dụng phương pháp phân cụm phổ biến đó là phương pháp K-means.

Phân cụm K-means (MacQueen, 1967) là thuật toán học máy không được giám sát được sử dụng để phân nhóm các đối tượng đã cho vào k cụm, trong đó k được chỉ định trước. Trong phân cụm K-means, mỗi cụm được biểu diễn bằng tâm của nó (centroid) tương ứng với trung bình của các điểm được gán cho cụm .

Ý tưởng chính của thuật toán K-means là xác định các cụm sao cho total within-cluster variation là nhỏ nhất với định nghĩa total within-cluster variation như sau:

$$tot.withiness = \sum_{k=1}^{k} \mathrm{W}(C_k) = \sum_{k=1}^{k} \sum_{x_i \in C_k} (x_i - \mu_k)^2$$

Trong đó: xi là điểm dữ liệu thuộc cụm Ck

μk là giá trị trung bình của các điểm trong cụm Ck .

2.2 Triển khai thuật toán phân cụm Kmeans

Giải thuật được mô tả như sau:

Input: số cụm k và n tài liệu.

Output: k cụm.

➢ **Các bước triển khai:**

1. Chọn ngẫu nhiên K tâm (centroid) cho K cụm (cluster). Mỗi cụm được đại diện bằng các tâm của cụm.
2. Tính khoảng cách giữa các đối tượng (objects) đến K tâm (thường dùng khoảng cách Euclidean)
3. Nhóm các đối tượng vào nhóm gần nhất
4. Xác định lại tâm mới cho các nhóm
5. Thực hiện lại bước 2 cho đến khi không có sự thay đổi nhóm nào của các đối tượng

Tuy nhiên, giải thuật này tồn tại một số nhược điểm như việc tính khoảng cách từ một phần tử đến tâm và việc tính toán và điều chỉnh tâm các cụm được thực hiện lặp lại sau mỗi bước gán một phần tử cho một cụm dẫn đến tiêu tốn nhiều tài nguyên hệ thống và thời lượng chạy chương trình sẽ lâu. Do vậy, giải thuật này phù hợp với lượng dữ liệu vừa và nhỏ, để triển khai cho tập dữ liệu lớn thì nó cần được cải thiện và thực hiện trên mô hình phù hợp hơn để hạn chế các nhược điểm nói trên

CHƯƠNG 3: ỨNG DỤNG MAP REDUCE KMEANS TRONG PHÂN CỤM DỮ LIỆU

3.1 Ý tưởng MapReduce Kmeans.

Giải thuật K-Means dựa trên mô hình MapReduce làm việc như sau:

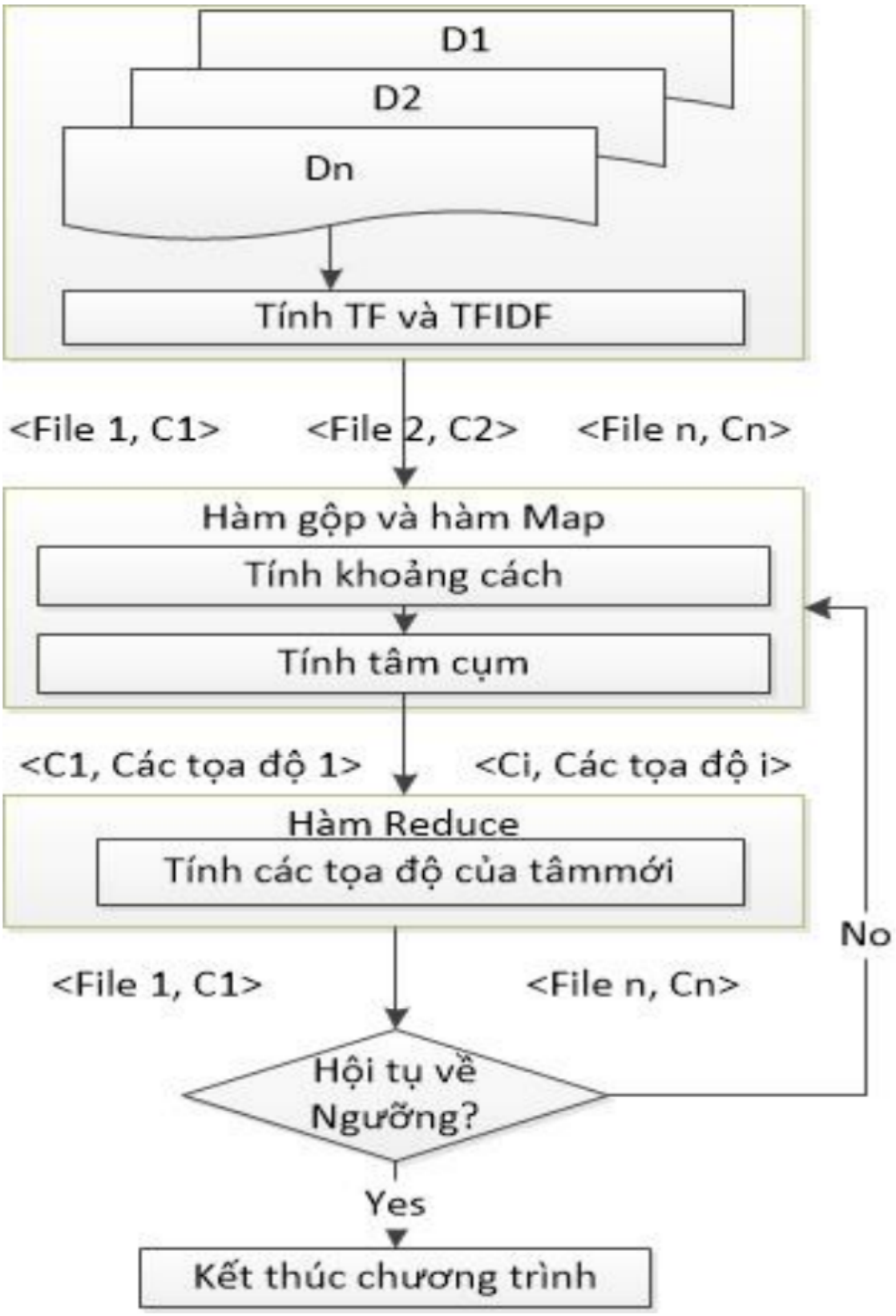
Bước 1: Giai đoạn đầu là tiền xử lý tài liệu. Chia tập tài liệu D thành m tập con. Có hai công việc MapReduce trong giai đoạn này, đầu tiên là phải tính toán các tham số cho bước tiếp theo như đếm từ, tính TF,.., tiếp theo là tính TFIDF (con số thể hiện mức độ quan trọng của từ trong một tài liệu trên tập các tài liệu) của mỗi tài liệu. Kết thúc giai đoạn này tài liệu được đánh chỉ số cũng như vector trọng số của nó cũng đã hoàn chỉnh, đã chọn được k tài liệu làm tâm, xác định kích thước mảnh dữ liệu, ngưỡng hội tụ để chương trình kết thúc.

Bước 2: Giai đoạn thứ hai là hàm map, đọc dữ liệu đầu vào và tính khoảng cách tới mỗi tâm. Với mỗi tài liệu, nó tạo ra cặp <key (chỉ số cụm), value(tọa độ của tài liệu)>. Rất nhiều dữ liệu được tạo ra trong giai đoạn này, hàm gộp có thể được sử dụng để giảm kích thước trước khi gửi đến Reduce. Hàm trộn được mô tả như sau: Hàm trộn tính trị trung bình của các tọa độ cho mỗi id cụm, cùng với số tài liệu. Tất cả dữ liệu cùng cụm hiện tại được gửi tới một reducer.

Bước 3: Giai đoạn thứ 3 là hàm reduce, hàm này tính tọa độ mới cho tâm các cụm. Dữ liệu đầu ra này được ghi vào tập tin của cụm bao gồm: số lần lặp, id cụm, các tọa độ của tâm cụm, kích thước của cụm.

Bước 4: Cuối cùng các tọa độ của cụm mới được so sánh với các tọa độ ban đầu. Nếu hàm điều kiện hội tụ thì chương trình kết thúc và ta tìm được các cụm. Nếu không, sử dụng các tâm của cụm mới thực hiện và lặp lại từ bước 2 đến bước 4.

3.2 Lưu đồ của thuật toán MapReduce Kmeans.



3.3 Giải pháp MapReduce hóa Kmeans

3.3.1 Tổng quan bài toán.

Phân tích khách hàng là một nhánh cực kỳ quan trọng trong việc phân tích dữ liệu kinh doanh. Tìm hiểu hành vi, ghi nhận thói quen mua sắm, nắm bắt sở thích khách hàng v.v... luôn được các doanh nghiệp đầu tư bài bản nhằm tạo ra lợi thế cạnh tranh lâu dài. Nhóm khách hàng của một công ty thường đa dạng về thành phần, khác nhau về độ tuổi v.v... từ đó dẫn đến tâm lý mua sắm rất khác nhau. Do đó, các doanh nghiệp thường phải phân chia khách hàng ra thành các nhóm có những đặc điểm tương tự nhau, từ đó đưa ra các chiến lược sản xuất, tiếp thị sản phẩm nhằm đáp ứng tốt hơn nhu cầu mua sắm, tăng doanh thu công ty. Có nhiều cách để phân chia hay phân cụm khách hàng. Trước đây, bộ phận marketing phân cụm chủ yếu dựa vào các thông tin truyền thống như:

- Nhân khẩu học (bao gồm độ tuổi, giới tính, thu nhập và giáo dục)
- Tâm lý học (như tầng lớp xã hội, lối sống và đặc điểm cá tính)
- Dữ liệu hành vi (bao gồm thói quen chi tiêu)
- Thông tin địa lý (thị trấn, quận, thành phố, tiểu bang, quốc gia cư trú).

Trong bài báo này, chúng em nghiên cứu bài toán phân khúc khách hàng thông qua phương pháp MapReduce hoá K-means. Mục đích của việc phân cụm là tìm ra các phân khúc thị trường có ý nghĩa.

3.2.2 Phân tích dữ liệu thô.

Nguồn dữ liệu thô: Một cửa hàng bán lẻ trực tuyến tại Anh.

https://archive.ics.uci.edu/ml/datasets/online+retail

+ *Hiểu dữ liệu:* Tập dữ liệu xuyên quốc gia chứa tất cả các giao dịch xảy ra từ ngày 01/12/2010 đến 09/12/2011

+ *Dữ liệu gồm:* Dữ liệu bao gồm 541909 bản ghi cùng 8 thuộc tính về giao dịch của khách hàng, hình dưới đây hiển thị 5 dòng đầu và 5 dòng cuối của dữ liệu sử dụng python

```
retail.head()
```

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	12/1/2010 8:26	2.55	17850.0	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	12/1/2010 8:26	3.39	17850.0	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	12/1/2010 8:26	2.75	17850.0	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	12/1/2010 8:26	3.39	17850.0	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	12/1/2010 8:26	3.39	17850.0	United Kingdom

```
retail.tail()
```

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country	Amount
541904	581587	22613	PACK OF 20 SPACEBOY NAPKINS	12	12/9/2011 12:50	0.85	12680.0	France	10.20
541905	581587	22899	CHILDREN'S APRON DOLLY GIRL	6	12/9/2011 12:50	2.10	12680.0	France	12.60
541906	581587	23254	CHILDRENS CUTLERY DOLLY GIRL	4	12/9/2011 12:50	4.15	12680.0	France	16.60
541907	581587	23255	CHILDRENS CUTLERY CIRCUS PARADE	4	12/9/2011 12:50	4.15	12680.0	France	16.60
541908	581587	22138	BAKING SET 9 PIECE RETROSPOT	3	12/9/2011 12:50	4.95	12680.0	France	14.85

```
retail.shape
```

```
(541909, 8)
```

Mô tả các thuộc tính:

STT	Thuộc tính	Ý nghĩa thuộc tính
1	InvoiceNo	Số hoá đơn
2	StockCode	Mã sản phẩm
3	Description	Tên sản phẩm
4	Quantity	Số lượng của mỗi sản phẩm (mặt hàng) trên mỗi giao dịch
5	InvoiceDate	Thời gian của từng hoá đơn (Ngày, giờ)

6	UnitPrice	Giá trên từng sản phẩm
7	CustomerID	Mã khách hàng
8	Country	Tên nước

3.2.2.1 Làm sạch dữ liệu.

Là quá trình nhận dạng dữ liệu đã có để tiến hành xử lý các dữ liệu bị thiếu (missing data) xử lý dữ liệu bị nhiễu (noisy data) và không nhất quán.

(1) Xử lý dữ liệu bị thiếu (missing data)

(2) Xử lý dữ liệu nhiễu , không nhất quán (inconsistent data).

Thực hiện:

- Xử lý bằng python:
  + Thống kê số lượng giá trị bị thiếu trong dữ liệu, có 2 thuộc tính bị thiếu đó là Description thiếu 1454 giá trị (chiếm 0.27%) và CustomerID thiếu 135080 giá trị chiếm 24.93%

```
InvoiceNo          0    InvoiceNo       0.00
StockCode          0    StockCode       0.00
Description     1454    Description     0.27
Quantity           0    Quantity        0.00
InvoiceDate        0    InvoiceDate     0.00
UnitPrice          0    UnitPrice       0.00
CustomerID    135080    CustomerID     24.93
Country            0    Country         0.00
```

  + Xóa những hàng có giá trị bị thiếu: Sau khi xóa, dữ liệu còn lại 40 6829 dòng và 8 cột

```
retail.dropna(inplace=True)
```

```
retail.shape
```

```
(406829, 8)
```

3.2.2.2 Tích hợp dữ liệu.

Là quá trình trộn dữ liệu từ các nguồn khác nhau vào một kho dữ liệu sẵn sàng cho quá trình khai phá dữ liệu.

(1) Tích hợp lược đồ và so trùng đối tượng.

(2) Vấn đề dư thừa.

(3) Phát hiện và xử lý mâu thuẫn giá trị dữ liệu.

=> Dữ liệu lấy từ một nguồn nên không cần thực hiện quá trình này.

3.2.2.3 Biến đổi dữ liệu.

Là quá trình biến đổi hay kết hợp dữ liệu vào những dạng thích hợp cho quá trình khai phá dữ liệu.

Các chiến lược thu giảm:

+ Làm trơn dữ liệu.
+ Kết hợp dữ liệu.
+ Tổng quát hóa dữ liệu.
+ Chuẩn hóa dữ liệu.
+ Xây dựng thuộc tính đặc tính.
    - Xây dựng thuộc tính đặc tính

Chúng tôi sẽ phân tích Khách hàng dựa trên 3 yếu tố dưới đây:

R (Recency): Số ngày kể từ lần mua cuối cùng, để xem mức độ thường xuyên mua hàng của khách hàng, tính bằng cách lấy ngày xa nhất trừ đi ngày mua hàng của khách hàng đó

F (Frequency): Số lượng giao dịch, để tìm ra khách hàng tiềm năng, tính bằng cách đếm số lượng giao dịch dựa vào CustomerID

M (Monetary): Tổng số lượng giao dịch (doanh thu đóng góp), sẽ được tính bằng cách tính tổng số tiền mà mỗi CustomerID đã mua

Thuộc tính : Monetary

```
retail['Monetary'] = retail['Quantity']*retail['UnitPrice']
rfm_m = retail.groupby("CustomerID")['Monetary'].sum()
rfm_m = rfm_m.reset_index()
rfm_m.head()
```

	CustomerID	Monetary
0	12346.0	0.00
1	12347.0	4310.00
2	12348.0	1797.24
3	12349.0	1757.55
4	12350.0	334.40

Thuộc tính : Frequency

```
rfm_f = retail.groupby("CustomerID")['InvoiceNo'].count()
rfm_f = rfm_f.reset_index()
rfm_f.columns = ['CustomerID', "Frequency"]
rfm_f.head()
```

	CustomerID	Frequency
0	12346.0	2
1	12347.0	182
2	12348.0	31
3	12349.0	73
4	12350.0	17

Gộp 2 thuộc tính vào 1 bản

```
rfm = pd.merge(rfm_m, rfm_f, on='CustomerID', how='inner')
rfm.head()
```

	CustomerID	Monetary	Frequency
0	12346.0	0.00	2
1	12347.0	4310.00	182
2	12348.0	1797.24	31
3	12349.0	1757.55	73
4	12350.0	334.40	17

Thuộc tính: Recency

Chuyển đổi sang datetime thành kiểu dữ liệu thích hợp: vì khi đọc dữ liệu bằng pandas nhận thấy cột InvoiceDate có thuộc tính kiểu string, ta cần đưa dữ liệu về kiểu Timestamp

```
type(retail.InvoiceDate[0])
```

```
str
```

```
retail['InvoiceDate'] = pd.to_datetime(retail['InvoiceDate'], format='%m/%d/%Y %H:%M')
```

```
type(retail.InvoiceDate[0])
```

```
pandas._libs.tslibs.timestamps.Timestamp
```

Tìm ngày giao dịch cuối cùng được ghi trong bản ghi:

```
max_date = max(retail['InvoiceDate'])
max_date
```

```
Timestamp('2011-12-09 12:50:00')
```

Tính toán sự khác biệt giữa ngày giao dịch cuối cùng được lưu trong bản ghi và ngày giao dịch của mỗi khách hàng có ID là CustomerID

```
retail['Diff'] = max_date - retail['InvoiceDate']
retail.head()
```

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country	Monetary	Diff
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	United Kingdom	15.30	373 days 04:24:00
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom	20.34	373 days 04:24:00
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	United Kingdom	22.00	373 days 04:24:00
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom	20.34	373 days 04:24:00
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom	20.34	373 days 04:24:00

Tính toán ngày giao dịch cuối cùng để nhận được lần truy cập gần đây nhất của mỗi khách hàng

```
rfm_p = retail.groupby('CustomerID')['Diff'].min().reset_index()
rfm_p.head()
```

	CustomerID	Diff
0	12346.0	325 days 02:33:00
1	12347.0	1 days 20:58:00
2	12348.0	74 days 23:37:00
3	12349.0	18 days 02:59:00
4	12350.0	309 days 20:49:00

Chỉ trích xuất số ngày

```
rfm_p['Diff'] = rfm_p['Diff'].dt.days
rfm_p.head()
```

	CustomerID	Diff
0	12346.0	325
1	12347.0	1
2	12348.0	74
3	12349.0	18
4	12350.0	309

Hợp nhất các khung dữ liệu để có được khung dữ liệu cuối cùng, sau đó lưu lại dưới dạng csv.

```
rfm = pd.merge(rfm, rfm_p, on='CustomerID', how='inner')
rfm.columns = ['CustomerID','Monetary', 'Frequency', 'Recency']
rfm = rfm.drop("CustomerID", 'columns')
```

```
rfm.to_csv('rfm.csv', index = None)
```

```
rfm = pd.read_csv("rfm.csv")
rfm.head()
```

	Monetary	Frequency	Recency
0	0.00	2	325
1	4310.00	182	1
2	1797.24	31	74
3	1757.55	73	18
4	334.40	17	309

Sau khi tiền xử lý chúng em đã tạo ra một tệp dữ liệu mới để tiến hành MapReduce hoá Kmeans, dữ liệu sau khi chuyển thành file txt:

```
Monetary,Frequency,Recency
0.0,2,325
4310.0,182,1
1797.24,31,74
1757.55,73,18
334.4,17,309
1545.41,95,35
89.0,4,203
1079.4,58,231
459.4,13,213
2811.43,59,22
6207.67,131,32
1168.06,19,1
6245.53,254,7
2662.06,129,51
189.9,10,286
5154.58,274,2
552.0,23,109
1313.1,85,7
320.69,23,290
```

➢ **Triển khai:**

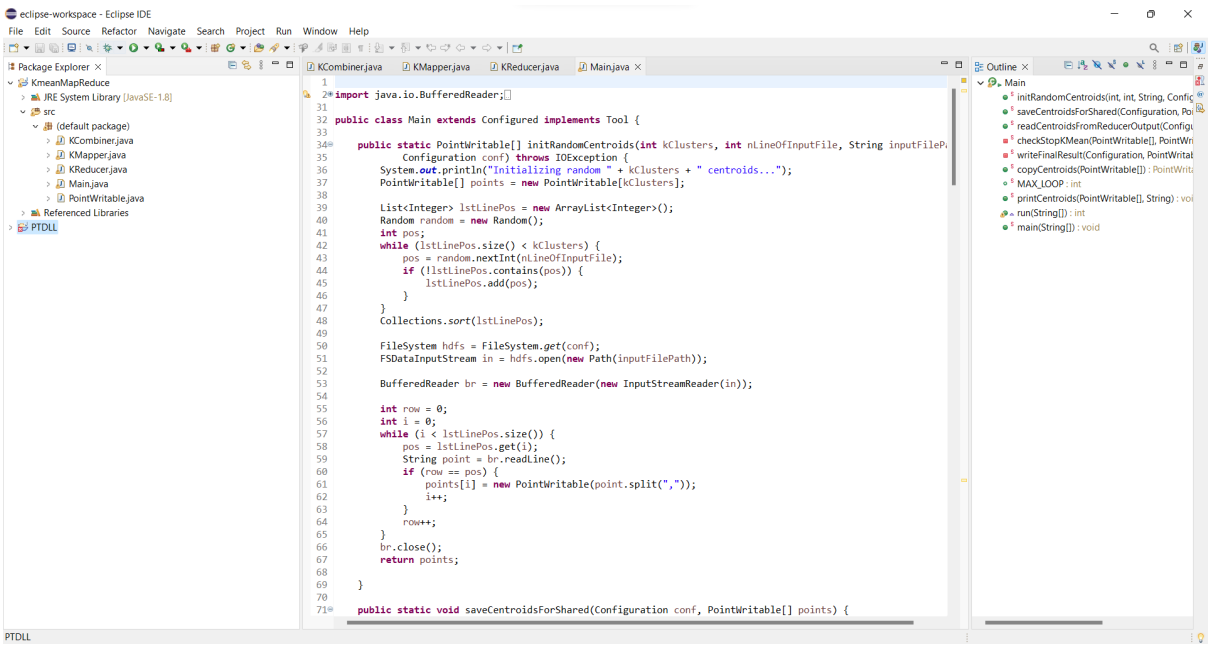
- Bước 1: Từ kết quả trên, tiến hành chọn k=3 , vector ngẫu nhiên làm tâm khởi điểm.
- Bước 2: Dữ liệu đầu vào cho hàm Map là 4372 dữ liệu. Tính khoảng cách từ các vector (2000) đến các tâm (3), việc tính toán này được phân phối để thực hiện song song. Kết quả bước này là 3 cặp (key, value) tương ứng (chỉ_số_cụm, tọa_độ_các_tài_liệu).
- Bước 3: Tính lại tọa độ các tâm. Dữ liệu đầu vào cho hàm Reduce là 3 cặp (key, value) do Map chuyển qua, dữ liệu được phân chia để thực hiện song song trên cả 2 node sau đó gộp lại. Kết quả hàm này ghi lại số lần lặp, chỉ số cụm, tọa độ của cụm, kích thước cụm.
- Bước 4: So sánh tọa độ cụm mới với tọa độ cụm trước đó. Nếu điều kiện hội tụ thỏa mãn (ngưỡng hội tụ =0.) thì dừng chương trình và cho kết quả 20 cụm là 20 tập tin chứa tọa độ của các vector của cụm. Ngược lại, điều kiện hội tụ chưa thỏa mãn thì lặp lại quá trình Map và Reduce từ bước 2 đến bước 4.

=> *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.

3.3 Demo chương trình cài đặt.

3.3. 1 Demo Chương trình demo.

(1) Cấu trúc thư mục của project:



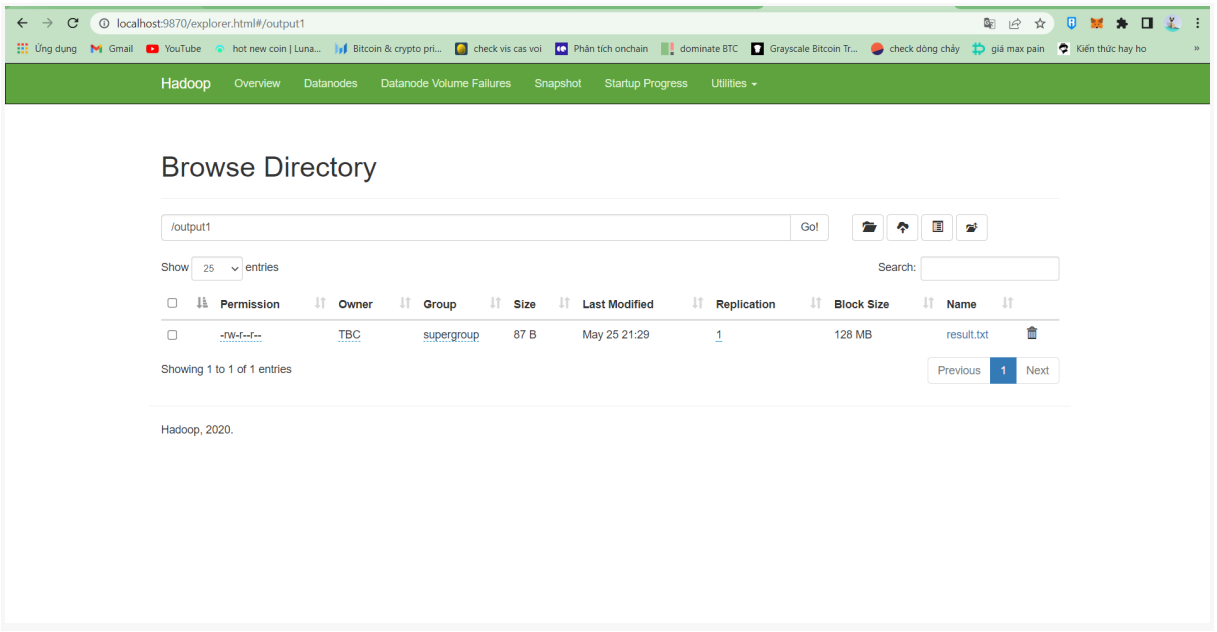
(2) Cách chạy chương trình Kmeans Hadoop:

```
Select Administrator: Command Prompt
C:\WINDOWS\system32>start-all
This script is Deprecated. Instead use start-dfs.cmd and start-yarn.cmd
starting yarn daemons

C:\WINDOWS\system32>cd C:\input1

C:\input1> hadoop fs -put rfm.csv /input1
put: `rfm.csv': No such file or directory

C:\input1> hadoop fs -put rfm.txt  /input1

C:\input1>hadoop fs -put centroid.txt /input1

C:\input1>hadoop jar KMeans.jar Main -Din /input1/rfm.txt  -Dlines 30 -Dresult result.txt -Dmaxloop 50 -Dk 3 -Dthresh 0.
1 -DNumReduceTask 2 -Dout /k-output
---------------INPUT PARAMETERS---------------
inputFilePath:/input1/rfm.txt
outputFolderPath:/k-output
outputFileName:result.txt
maxloop:50
numLineOfInputFile:30
nClusters:3
threshold:0.1
NumReduceTask:2
--------------- STATR ---------------
Initializing random 3 centroids...
=> CURRENT CENTROIDS:
centroids(init)[0]=> :1797.24,31.0,74.0
centroids(init)[1]=> :334.4,17.0,309.0
centroids(init)[2]=> :320.69,23.0,290.0
---------------------------------
```

(3) Kết quả chương trình

```
C:\input1>hdfs dfs -cat /output1/result.txt
5614.8926,284.55453,25.59861
807.8949,59.58454,99.59124
21474.836,868.8197,10.01639

C:\input1>
```

(4) File kết quả đầu ra: Là1 file txt có tên là result.txt



CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

4.1 Kết luận.

Từ kết quả nghiên cứu các thuật toán K-Means, thuật toán ban đầu không thể thực hiện với tập dữ liệu lớn hàng terabyte. Do đó, chúng tôi đề xuất giải pháp phân cụm dữ liệu văn bản lớn trên mô hình MapReduce, với kỹ thuật này hiệu suất, thời lượng và tính ổn định của việc phân cụm dữ liệu lớn được cải thiện và hiệu quả hơn nhiều so với giải thuật K-Means ban đầu. Kết quả thực nghiệm cũng cho thấy tính hiệu quả của việc phát triển giải thuật K-Means trên mô hình MapReduce so với thuật toán K-Means ban đầu.

4.2 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. Tài liệu tiếng việt.

- Nguyễn Hà Nam, Nguyễn Chí Thành, Hà Quang Thụy, Giáo trình Khai phá dữ liệu, 2016, Nhà xuất bản Đại học Quốc Gia Hà Nội
- TS. Nguyễn Huy Đức, TS.Đặng Thị Thu Hiền, Bài giảng Khai phá dữ liệu.
- TS. Ta Quang Chieu, Bài giảng Phân tích dữ liệu lớn

- https://archive.ics.uci.edu/ml/datasets/online+retail

- https://machinelearningcoban.com/2017/01/01/kmeans/

2. Tài liệu tiếng anh.

- Dolnicar S, Grn B, Leisch F. Market Segmentation. Market Seg- mentation Analysis: Understanding It, Doing It, and Making It Useful. Springer; 2018. p. 11–22. 2.

- Kassambara A. Practical guide to cluster analysis in R: unsuper- vised machine learning. In: STHDA; 2017. .

- Kanungo T, Mount DM, Netanyahu NS, Piatko CD, Silverman R, Wu A, et al. An efficient k-means clustering algorithm: Analysis and implementation. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2002;7:881–92.

- Khan SS, Ahmad A. Ahmad AJPrl. Cluster center initialization algorithm for K-means clustering. Pattern Recognition Letters. 2004;25(11):1293–302.

- A density-based algorithm for discovering clusters in large spa- tial databases with noise. In: Ester M, Kriegel HP, Sander J, Xu X, editors. Proceedings of the SecondInternational Conference on Knowledge Discovery andData Mining (KDD-96). AAAI Press; 1996. p. 226–231.