**TRƯỜNG ĐẠI HỌC THỦY LỢI**

**KHOA CÔNG NGHỆ THÔNG TIN**

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


**BÁO CÁO BÀI TẬP LỚN**

**HỌC PHẦN PHÂN TÍCH DỮ LIỆU LỚN**

**ỨNG DỤNG HADOOP MAPREDUCE VÀ THUẬT TOÁN K-MEANS TRONG BÀI TOÁN PHÂN CỤM DỮ LIỆU KHÁCH HÀNG**

Nhóm sinh viên thực hiện: **Nhóm 18**

*Nguyễn Đình Tư – 62THNB – 2051063803*

*Nguyễn Thế Quang – 62THNB – 2051063773*

*Đào Thị Kiều Trang – 62THNB – 2051063436*

*Đinh Việt Anh – 62THNB – 2051063878*

**Giảng viên hướng dẫn: Nguyễn Đắc Hiếu**

**Hà Nội, Tháng 11 Năm 2023**

**MỤC LỤC**

[I. CHƯƠNG 1: TỔNG QUAN, GIỚI THIỆU BÀI TOÁN 3](#_Toc150519444)

[1.1. Giới thiệu bài toán 3](#_Toc150519445)

[1.2. Tổng quan các kỹ thuật, phương pháp liên quan đến bài toán 3](#_Toc150519446)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 3](#_Toc150519447)

[2.1. Tổng quan về dữ liệu lớn 3](#_Toc150519448)

[2.2. Tổng quan về Apache Hadoop 4](#_Toc150519449)

[2.3. Tổng quan về Mapreduce 5](#_Toc150519450)

[2.4. Tổng quan về thuật toán K-means 6](#_Toc150519451)

[2.4.1. Thuật toán K-means 6](#_Toc150519452)

[2.4.2. MapReduce hóa thuật toán K-means: 7](#_Toc150519453)

[CHƯƠNG 3: KẾT QUẢ VÀ ĐÁNH GIÁ MÔ HÌNH 10](#_Toc150519454)

[3.1. Chi tiết bài toán 10](#_Toc150519455)

[3.1.1. Phân tích dữ liệu thô 10](#_Toc150519456)

[3.1.2. Tiền xử lý dữ liệu 11](#_Toc150519457)

[3.1.3. Xác định số lượng cụm K 12](#_Toc150519458)

[3.1.4. Sử dụng MongoDB để lưu trữ dữ liệu 14](#_Toc150519459)

[3.1.5. Xây dựng các API bằng Node.js để thao tác với MongoDB 16](#_Toc150519460)

[3.1.6. Sử dụng Hadoop MapReduce để phân cụm dữ liệu 17](#_Toc150519461)

[3.2. Kết quả bài toán 18](#_Toc150519462)

[3.2.1. Demo cài đặt chương trình 18](#_Toc150519463)

[3.2.2. Kết quả 19](#_Toc150519464)

[3.2.3. Demo ứng dụng dự đoán nhóm khách hàng 21](#_Toc150519465)

[CHƯƠNG 4: KẾT LUẬN 22](#_Toc150519466)

[TÀI LIỆU THAM KHẢO 23](#_Toc150519467)

# CHƯƠNG 1: TỔNG QUAN, GIỚI THIỆU BÀI TOÁN

## Giới thiệu bài toán

Bài toán tập trung vào việc ứng dụng kỹ thuật phân tích dữ liệu sử dụng Hadoop MapReduce kết hợp với thuật toán phân cụm K-means trong bài toán phân cụm dữ liệu khách hàng.

Bộ dữ liệu được sử dụng cho bài toán chứa các thông tin đa dạng về chỉ số những đặc điểm của khách hàng. Mục tiêu chính của bài toán là thực hiện phân cụm dữ liệu để xác định nhóm phân khúc khách hàng dựa trên các đặc trưng liên quan đến việc mua sắm của họ.

Kết quả của bài toán sẽ là việc phân loại các khách hàng vào các nhóm tương ứng với đặc điểm mua sắm của họ dựa trên sự tương tự về đặc trưng của khách hàng. Điều này có thể giúp cung cấp thông tin quan trọng về xu hướng và mối liên quan giữa các đặc điểm khách hàng khác nhau và tình hình mua sắm. Từ đó có thể đưa ra các đề xuất về biện pháp về chăm sóc khách hàng và giúp cửa hàng hiểu hơn về khách hàng của mình.

## Tổng quan các kỹ thuật, phương pháp liên quan đến bài toán

* **K-means Clustering**: là một thuật toán phân cụm không giám sát, sử dụng để phân chia các nhóm khách hàng vào các phân khúc dựa trên các đặc điểm khách hàng như độ tuổi, thu nhập, quy mô gia đình …
* **MapReduce**: là một mô hình lập trình và hệ thống xử lý dữ liệu song song phục vụ cho việc xử lý các tập dữ liệu lớn. Bằng cách chia nhỏ nhiệm vụ tính toán thành các phần nhỏ và thực hiện trên nhiều node khác nhau.
* **HDFS (Hadoop Distributed File System)**: là hệ thống lưu trữ dữ liệu phân tán của Hadoop**.**
* **MongoDB**: là một hệ quản trị cơ sở dữ liệu không quan hệ, sử dụng để lưu trữ dữ liệu của khách hàng, và lưu trữ thông tin của các cụm, phục vụ cho ứng dụng dự đoán khách hàng thuộc nhóm đặc điểm về phân khúc khách hàng nào.
* **Node.js**: là môi trường thực thi Javascript phía máy chủ, sử dụng để xây dựng ứng dụng và các API để thao tác với cơ sở dữ liệu MongoDB.
* **Python**: sử dụng cho việc tiền xử lý dữ liệu và xác định số lượng cụm.

# CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

## 2.1. Tổng quan về dữ liệu lớn

* **Theo wikipedia**: Dữ liệu lớn (Big data) là một thuật ngữ cho việc xử lý một tập hợp dữ liệu rất lớn và phức tạp mà các ứng dụng xử lý dữ liệu truyền thống không xử lý được.



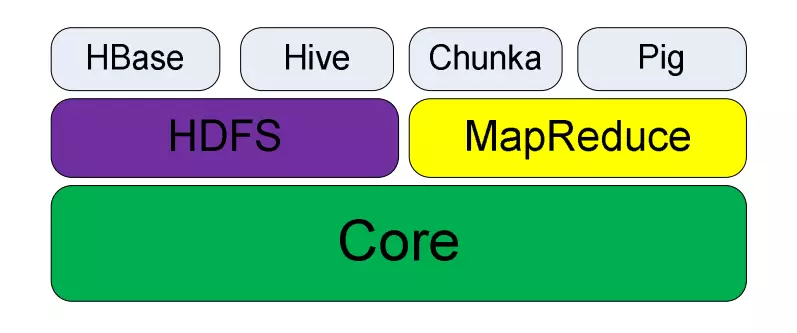
Dữ liệu lớn thường có năm đặc điểm chính:

* **Khối lượng (Volume):** Dữ liệu lớn đề cập đến khối lượng thông tin khổng lồ, thường là hàng tỷ hoặc triệu tỷ bản ghi dữ liệu.
* **Tốc độ (Velocity):** Dữ liệu lớn được tạo ra và cập nhật với tốc độ nhanh chóng từ nhiều nguồn khác nhau, đặc biệt trong thời gian thực.
* **Đa dạng (Variety):** Dữ liệu lớn có đa dạng về nguồn gốc và định dạng, bao gồm văn bản, hình ảnh, video, dữ liệu cảm biến, và nhiều loại dữ liệu khác.
* **Chính xác (Veracity):** Chính xác của dữ liệu là yếu tố quan trọng, đảm bảo rằng dữ liệu không bị nhiễu, thiếu hoặc chứa thông tin sai lệch.
* **Giá Trị (Value):** Giá trị thông tin là tính chất quan trọng nhất của công nghệ Big Data

## 2.2. Tổng quan về 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. 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.

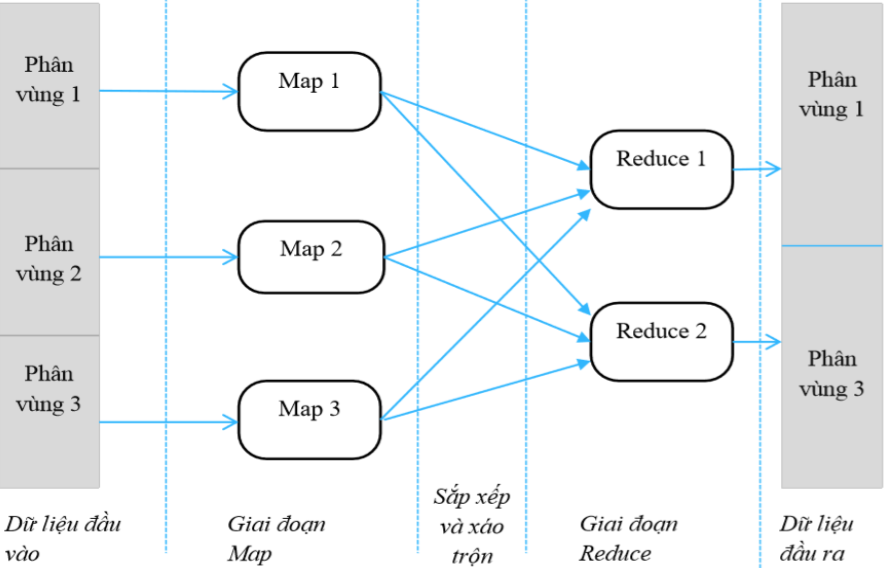




* Các thành phần chính của Apache Hadoop bao gồm: HDFS, MapReduce, Hbase, Hive, Pig, Chunka,…Tuy nhiên hai thành phần chính và quan trọng nhất là HDFS và MapReduce
* Hadoop Distributed File System (HDFS): là hệ thống lưu trữ dữ liệu phân tán của Hadoop, chia nhỏ dữ liệu thành các khối và lưu trữ trên nhiều node khác nhau. Điều này đảm bảo tính sẵn sàng, độ tin cậy và khả năng mở rộng của dữ liệu lớn.
* MapReduce: là mô hình lập trình và hệ thống xử lý dữ liệu song song trong Hadoop. Qua hai bước "map" và "reduce", nó phân tách và xử lý dữ liệu lớn trên nhiều máy tính, giúp tối ưu hóa quá trình xử lý và phân tích dữ liệu.

## 2.3. Tổng quan về Mapreduce

* MapReduce là mô hình được thiết kế độc quyền bởi Google, nó có khả năng lập trình xử lý các tập dữ liệu lớn song song và phân tán thuật toán trên 1 cụm máy tính.
* Mô hình MapReduce chia quá trình xử lý dữ liệu thành hai phần chính: pha ‘Map’ và pha ‘Reduce’



* Quy trình thực hiện công việc
* B1: Chia dữ liệu đầu vào thành các mảnh dữ liệu
* B2: Thực hiện công việc Map trên từng mảnh dữ liệu đầu  
  vào => Xử lý song song các mảnh dữ liệu trên nhiều máy  
  tính trong cụm
* B3: Tổng hợp kết quả trung gian (Sắp xếp, trộn)
* B4: Sau khi tất cả công việc Map hoàn thành, thực hiện  
  công việc Reduce trên từng mảnh dữ liệu trung gian => Thực hiện song song các mảnh dữ liệu trung gian  
  trên nhiều máy tính trong cụm
* B5: Tổng hợp kết quả hàm Reduce để cho kết quả cuối  
  cùng

## 2.4. Tổng quan về thuật toán K-means

### **2.4.1. Thuật toán K-means**

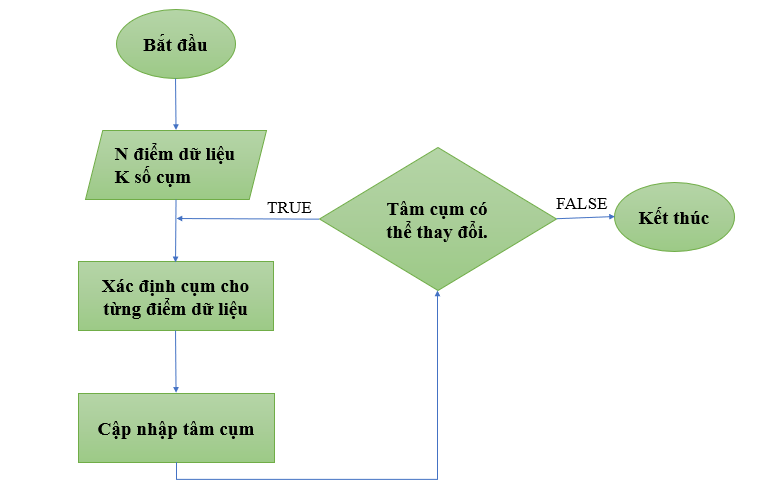
Thuật toán K-means là một trong các thuật toán phân cụm đơn giản và điển hình nhất, do MacQueen đề xuất trong lĩnh vực thống kê năm 1967. Mục đích của thuật toán là sinh ra *k* cụm dữ liệu từ một tập dữ liệu ban đầu gồm *n* đối tượng trong không gian *p* chiều.

**Phát biểu bài toán phân cụm:**

* Input: *n* đối tượng và số các cụm *k*
* Output: các cụm *Ci (i = 1…k)* sao cho hàm tiêu chuẩn E đạt giá trị tối thiểu

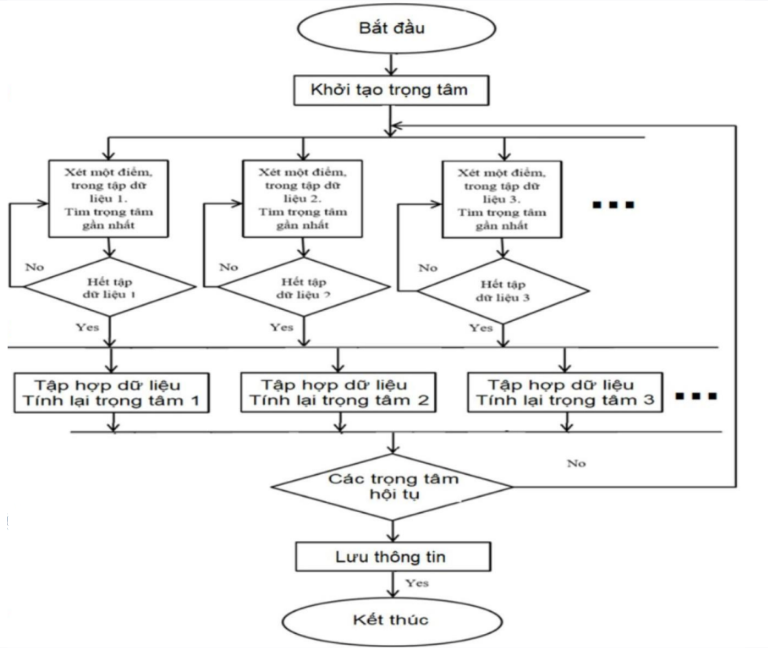
**Các bước của thuật toán K-mean: [3]**

* Bước 1: Khởi tạo tâm cụm
  + Chọn *k* đối tượng *mj (j=1…k)* là trọng tâm ban đầu của *k* cụm từ tập dữ liệu
* Bước 2: Tính toán khoảng cách và gán cụm
  + Với mỗi điểm dữ liệu, tính toán khoảng cách từ nó tới mỗi tâm cụm đã được khởi tạo, sau đó tìm tâm cụm gần nhất đối với mỗi đối tượng
  + Công thức tính khoảng cách Euclidean giữa hai điểm A (a1, a2, ..., an) và B (b1, b2, ..., bn) trong không gian n chiều là:
* Bước 3: Cập nhật lại trọng tâm
  + Với mỗi *j=1,…,k* , cập nhật trọng tâm cụm mj bằng cách xác định trung bình cộng của các vector đối tượng dữ liệu
* Bước 4: Kiểm tra điều kiện dừng
  + Lặp lại các bước 2 và 3 cho đế khi các tâm cụm không thay đổi



### **2.4.2.** **MapReduce hóa thuật toán K-means:**

* MapReduce hóa thuật toán K-means giúp tận dụng khả năng xử lý song song của MapReduce để thực hiện thuật toán K-means trên dữ liệu lớn.



Hình 2.6 Lưu đồ MapReduce hóa thuật toán K-means

**Các bước thực hiện:**

* Biểu diễn dữ liệu
  + Dữ liệu lưu trữ dưới dạng list các hàng
  + Mỗi hàng là list giá trị, là các thành phần của vector biểu diễn cho một điểm
* Lưu trữ phân tán dữ liệu:
  + Do các điểm được tính toán độc lập với nhau → có thể lưu trữ các phần của dữ liệu trên nhiều máy khác nhau để có thể xử lý song song và tăng tốc tính toán
* Trong mỗi vòng lặp
  + B1: tính khoảng cách của mỗi điểm trong phần dữ liệu của nó với các trọng tâm
  + B2: kiểm tra xem điểm đó gần trọng tâm nào nhất
  + B3: gom các điểm thuộc cùng một cụm để tính lại trọng tâm sau mỗi vòng lặp

**Giải pháp MapReduce hóa K-means**

* Dữ liệu cần phân cụm là danh sách các hàng (có thể lưu trên file txt) được chuyển sang kiểu key-value làm đầu vào 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 K-means:
  + Xây dựng hàm Map\_K-means
  + Xây dựng hàm Reduce\_K-means

**Xây dựng hàm Map\_K-means**

* Đầu vào: cặp key-value biểu diễn tọa độ của một điểm
  + keyIn là giá trị byte offset của dòng
  + valIn là vector biểu diễn tọa độ của một điểm
* Xử lý:
  + Tính khoảng cách của điểm với các trọng tâm (chưa phải là trọng tâm cần tìm)
  + Chuyển về cụm có tâm gần nhất
* Đầu ra: cặp key-value trung gian
  + keyInt là tâm gần nhất (trọng tâm hoặc chỉ số tâm)
  + valInt là tọa độ điểm thuộc cụm có trọng tâm là keyInt

**Xây dựng hàm Reduce\_K-means**

* Trước khi hàm reduce thực hiện
  + Kết quả của hàm map được trộn lại
  + Các cặp cùng keyInt được gom thành một nhóm
* Đầu vào:
  + keyInt được chuyển từ hàm map
  + list(valInt) là list các điểm vallInt thuộc về cụm thứ keyInt
* Xử lý:
  + Tính trung bình cộng từng thành phần của các điểm cùng cụm
  + Cập nhật lại trọng tâm của cụm đó
* Đầu ra:
  + keyOut là tâm mới
  + valOut là danh sách các điểm

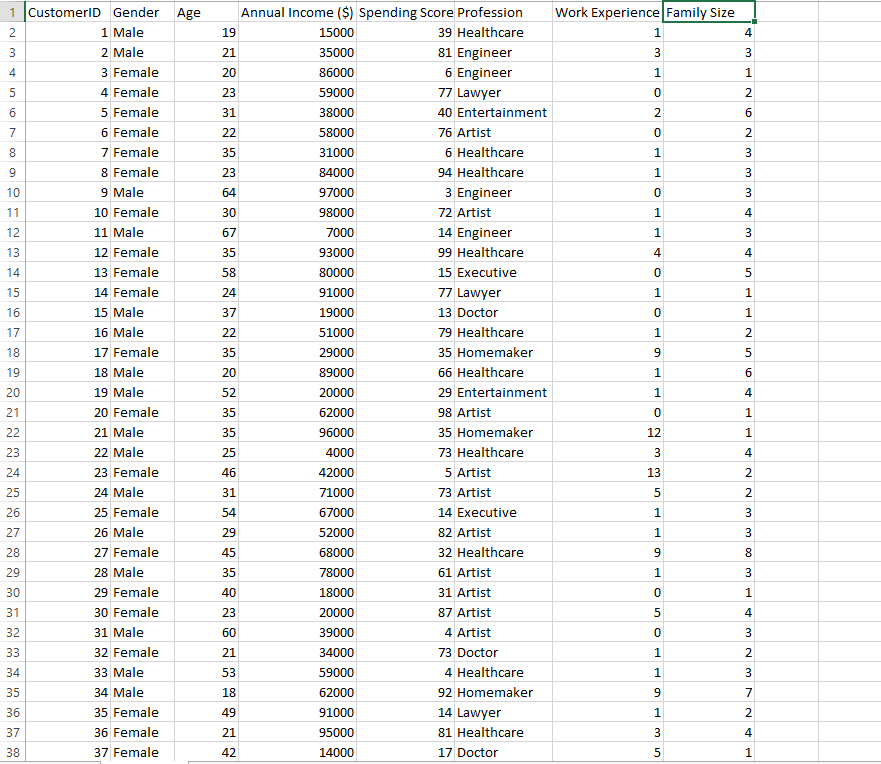
# CHƯƠNG 3: KẾT QUẢ VÀ ĐÁNH GIÁ MÔ HÌNH

## 3.1. Chi tiết bài toán

### **3.1.1. Phân tích dữ liệu thô**

Nguồn dữ liệu thô: [Shop Customer Clustering](https://www.kaggle.com/datasets/datascientistanna/customers-dataset), bộ dữ liệu thô về khách hàng dùng cho việc phân cụm.

* Số lượng thuộc tính: 8
* Số lượng bản ghi: 2000



### **3.1.2. Tiền xử lý dữ liệu**

**3.1.2.1 . Làm sạch dữ liệu**

* 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 dữ liệu không nhất quán (inconsistent data).

Đọc file dữ liệu và thực hiện làm sạch dữ liệu bằng Python:

* Xóa những thuộc tính không cần thiết: những thuộc tính không cần thiết như ‘CustomerID’, ‘Gender’ sẽ được xóa bỏ
* Xử lý thuộc tính ‘Profession’ bị thiếu giá trị: thay thế bằng giá trị xuất hiện thường xuyên nhất của thuộc tính đó, sử dụng **mode()** của thư viện pandas

**3.1.2.2. Biến đổi dữ liệu**

Biến đổi dữ liệu là quá trình thay đổi cấu trúc hoặc nội dung của dữ liệu để tạo ra các biến mới hoặc thay đổi dạng của dữ liệu hiện có. Mục tiêu của bước này là cải thiện tính phù hợp của dữ liệu với mục tiêu phân tích hoặc mô hình hóa.

Đọc file dữ liệu và thực hiện biến đổi dữ liệu bằng Python:

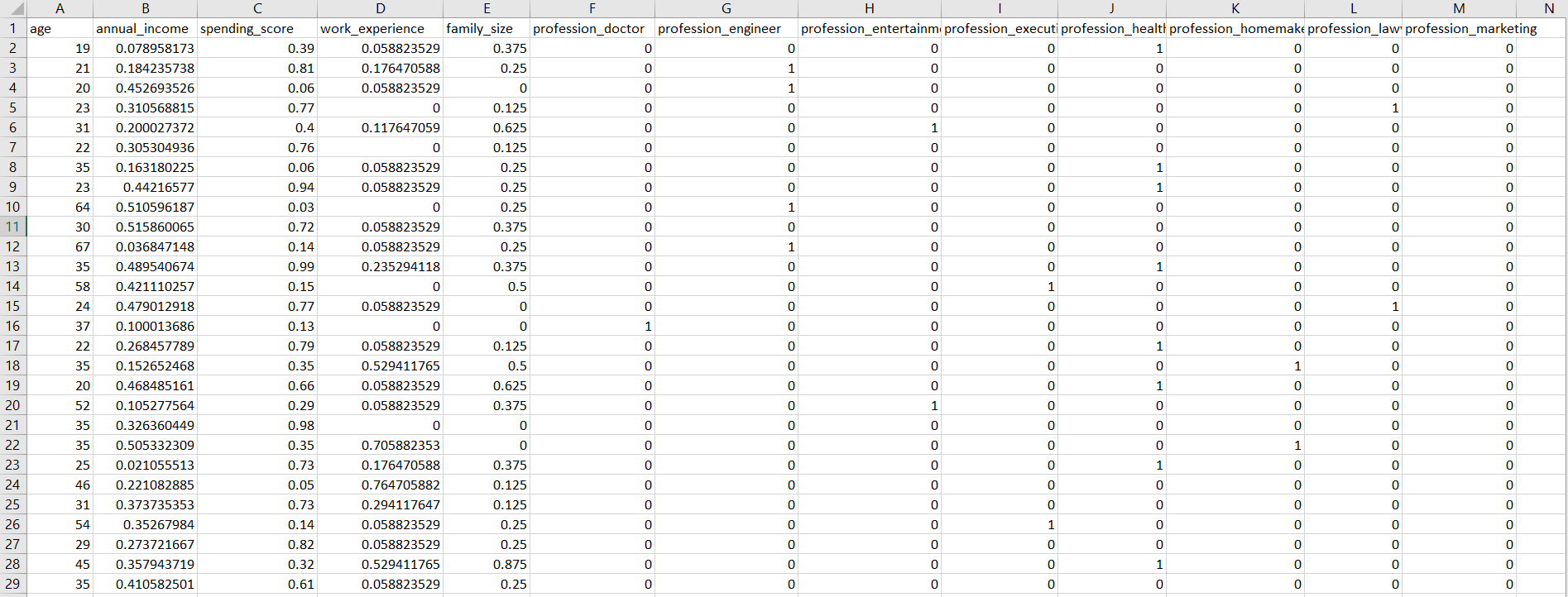
* Biến đổi dữ liệu bằng phương pháp Min-Max Scaling: đưa các giá trị dữ liệu về khoảng giá trị cụ thể, thường là trong khoảng từ 0 đến 1.
* Công thức:

Trong đó:

* v = [minA, maxA] là giá trị cũ
* v’ = [0,1] là giá trị mới

Ví dụ:

* Thuộc tính ‘Family\_size’: có v = [1, 9]
* Chuẩn hóa về giá trị: v’ = [0, 1]
* Với v = 4
  + - v’ = (4 - 1) / (9 - 1) = 0.375
* Biến đổi dữ liệu bằng phương pháp One-hot encoding: biến các giá trị cố định ở dạng ký tự của thuộc tính ‘Profession’ về dạng giá trị số, sử dụng get\_dummies() trong thư viện Pandas



* Các thuộc tính của bộ dữ liệu sau khi đã chuẩn hóa

|  |  |  |
| --- | --- | --- |
| **STT** | **Thuộc tính** | **Mô tả thuộc tính** |
| 1 | age | Tuổi |
| 2 | annual\_income | Thu nhập hàng năm |
| 3 | spending\_score | Điểm chi tiêu – do cửa hàng ấn định dưa trên tính chất chi tiêu và thái độ hành vi của khách hàng |
| 4 | work\_experience | Kinh nghiệm làm việc |
| 5 | family\_size | Quy mô gia đình |
| 6 | profession\_doctor | Nghề nghiệp là bác sĩ |
| 7 | profession\_engineer | Nghề nghiệp là kĩ sư |
| 8 | profession\_entertainment | Nghề nghiệp là giải trí |
| 9 | profession\_executive | Nghề nghiệp là quản lý |
| 10 | profession\_healthcare | Nghề nghiệp là điều dưỡng |
| 11 | profession\_homemaker | Nghề nghiệp là nội trợ |
| 12 | profession\_lawyer | Nghề nghiệp là luật sư |
| 13 | profession\_marketing | Nghề nghiệp là marketing |

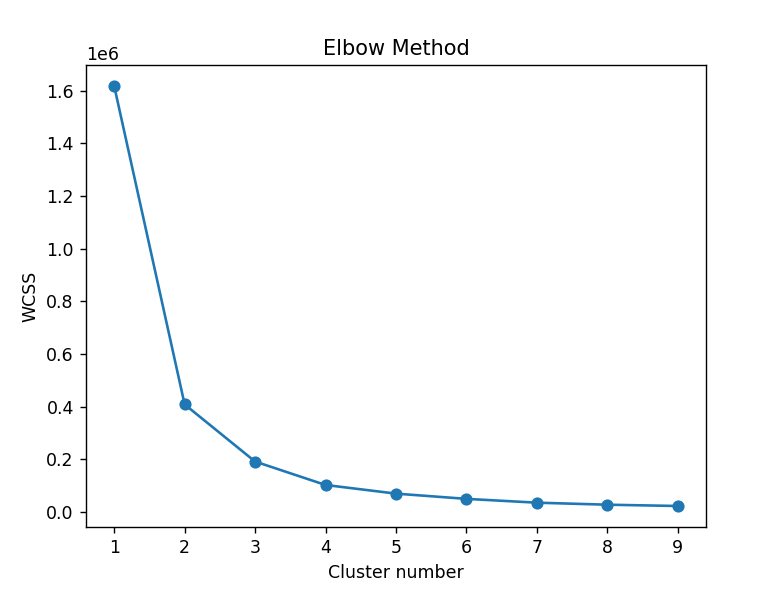
### **3.1.3. Xác định số lượng cụm K**

Để xác định số lượng cụm K, cần sử dụng một biểu đồ được tạo bằng Elbow Method (phương pháp khuỷu tay) – là một phương pháp thường được sử dụng để chọn số lượng tốt nhất của các cụm trong thuật toán phân cụm. Phương pháp này giúp xác định số lượng cụm tối ưu dựa trên sự thay đổi của tổng bình phương khoảng cách (Within-Cluster Sum of Squares - WCSS) giữa các điểm dữ liệu và trung tâm của cụm.

Cách thức thực hoạt động của Elbow Method như sau:

* Thực hiện thuật toán KMeans với một loạt số lượng cụm khác nhau (thường từ 1 đến một giới hạn tối đa đã xác định từ trước).
* Đối với mỗi số lượng cụm, tính giá trị WCSS (tổng bình phương khoảng cách giữa các điểm dữ liệu và trung tâm cụm tương ứng).
* Vẽ biểu đồ Elbow Method với trục x là số lượng cụm và trục y là giá trị WCSS. Đường cong trên biểu đồ thường có dạng "cổ điển" như cùi chỏ, giống như đoạn của một cánh tay và một bàn tay đang giơ lên.
* Xác định điểm "bẻ" của đường cong (nơi mà đường cong thay đổi hướng từ việc giảm dần sang giảm chậm hơn). Điểm này thường gọi là "đỉnh của cùi chỏ" (elbow point).
* Số lượng cụm tốt nhất sẽ là số lượng cụm tương ứng với điểm "bẻ" trên biểu đồ Elbow Method.

Ý nghĩa của Elbow Method là khi số lượng cụm tăng, WCSS thường giảm vì các điểm dữ liệu được gom vào cụm một cách gần hơn với trung tâm cụm. Tuy nhiên, tại một điểm nào đó, việc tăng số lượng cụm không còn gây ra sự giảm đáng kể trong WCSS, và đây chính là điểm mà đường cong bẻ. Điều này xảy ra vì thêm một cụm mới không còn đáng giá trong việc giảm tổng bình phương khoảng cách nữa, và sự thay đổi giữa số lượng cụm đã đạt một mức ổn định.

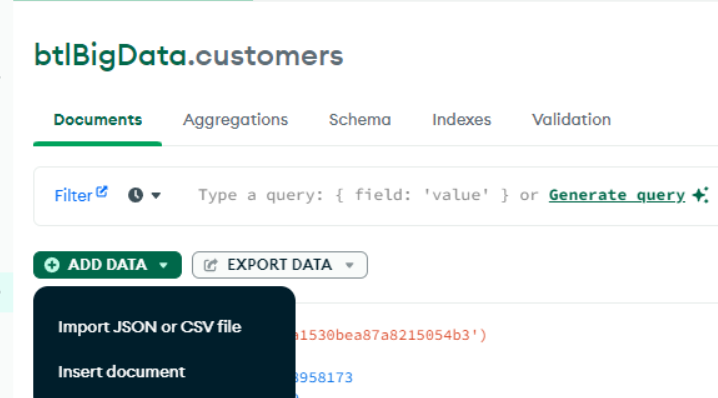


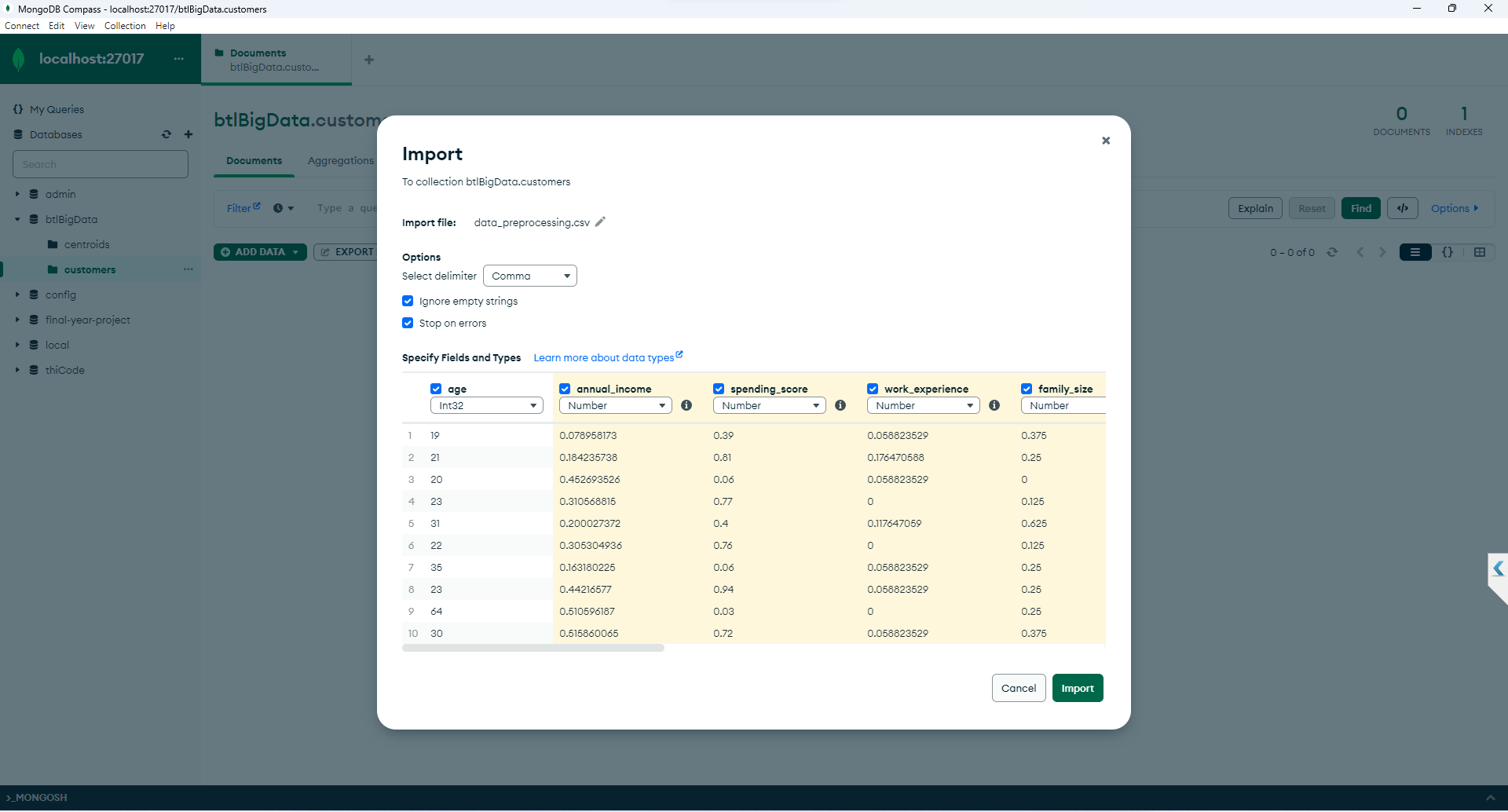
Từ biểu đồ trên, nhận thấy rằng tại số tượng cụm bằng 3 thì biểu đồ thay đổi hướng từ việc giảm dần sang giảm chậm hơn. Kết luận rằng số lượng cụm tối ưu cho dữ liệu là 3 cụm

### **3.1.4. Sử dụng MongoDB để lưu trữ dữ liệu**

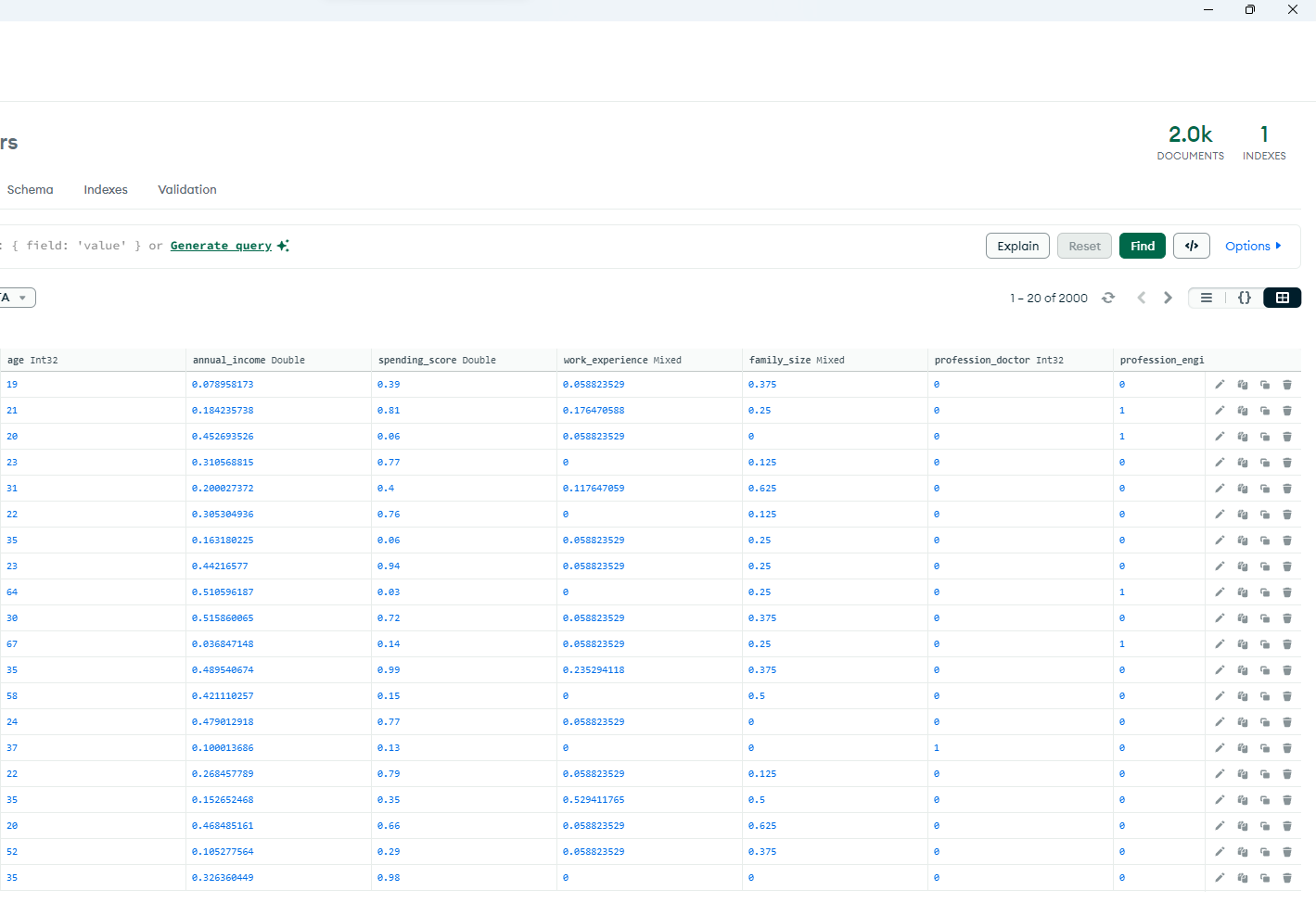
Dữ liệu sau khi đã được chuẩn hóa, thực hiện import file csv vào trong MongoDB:

* Chọn ‘Import JSON or CSV file’ trong MongoDB Compass
* Chọn file CSV cần import
* Kiểm tra lại kiểu dữ liệu của các trường dữ liệu
* Ấn ‘Import’ và kiểm tra dữ liệu đã được import đầy đủ hay chưa





Dữ liệu sau khi được import vào MongoDB:



### **3.1.5. Xây dựng các API bằng Node.js để thao tác với MongoDB**

* Lấy dữ liệu của tất cả khách hàng:
  + URL: <http://127.0.0.1:5000/api/customers>
  + Phương thức: GET
  + Mô tả: API cho phép truy cập dữ liệu của tất cả các khách hàng trong MongoDB để phục vụ cho việc phân cụm dữ liệu khách hàng bằng Hadoop MapReduce
  + Phản hồi: trả về tất cả dữ liệu của khách hàng ở dạng chuỗi
* Lưu tâm cụm vào MongoDB
  + URL: <http://127.0.0.1:5000/api/centroids>
  + Phương thức: POST
  + Body: dữ liệu của các tâm cụm dạng chuỗi
  + Mô tả: API cho phép lưu trữ các tâm cụm sau khi đã thực hiện phân cụm bằng thuật toán K-means vào MongoDB
  + Phản hồi: trả về dữ liệu của các tâm cụm nếu lưu thành công
* Dự đoán nhóm tiêu dùng của khách hàng:
  + URL: <http://127.0.0.1:5000/api/centroid-predict>
  + Phương thức: POST
  + Body: dữ liệu về các thuộc tính điểm tích lũy mua hàng của một khách hàng dạng JSON
  + Mô tả: API thực hiện chuẩn hóa dữ liệu đầu vào, sau đo tính toán khoảng cách từ điểm dữ liệu đầu tới các tâm cụm xem tâm cụm nào gần nhất, sau đó thực hiện lưu dữ liệu vào MongoDB
  + Phản hồi: trả về dữ liệu của khách hàng đã được chuẩn hóa, dữ liệu của tâm cụm gần nhất và những đặc trưng về mua sắm của tâm cụm đó

### **3.1.6. Sử dụng Hadoop MapReduce để phân cụm dữ liệu**

**3.1.6.1. Xây dựng lớp CallAPI**

* Hàm *fetchAndWriteDataToHDFS(conf, inputFilePath):* thực hiện gọi đến API lấy dữ liệu của khách hàng, lưu kết quả phản hồi của API là dữ liệu của các khách hàng vào file input trên HDFS
* Hàm *saveCentroidsToMongoDB(centroids):* thực hiện gửi dữ liệu của các tâm cụm đến API lưu tâm cụm vào MongoDB và trả về thông báo.

**3.1.6.2. Xây dựng lớp PointWritable**

* Lớp PointWritable để khởi tạo các đối tượng điểm dữ liệu bao gồm các thuộc tính
  + attibutes: mảng chứa các giá trị thuộc tính của mỗi điểm dữ liệu
  + dim: số chiều của mỗi điểm dữ liệu, tức là số lượng thuộc tính trong mỗi điểm
  + nPoints: đếm số lượng điểm dữ liệu được gộp lại với nhau, để tính tổng của tất cả các cụm để tính ra tâm cụm mới
* Cung cấp các phương thức để đọc và ghi dữ liệu của các điểm dữ liệu, tính tổng các giá trị thuộc tính, tính khoảng cách dữ điểm hiện tại và điểm p dựa vào các thuộc tính của chúng, tính trung bình của các thuộc tính trong dữ liệu dựa trên số lượng điểm đã gộp

**3.1.6.3. Xây dựng lớp Kmapper**

* Thực hiện việc tính toán khoảng cách từ điểm dữ liệu đến tất cả các tâm cụm, sau đó xác định cụm gần nhất.
* Đầu vào: key là vị trí dòng của điểm trong tập tin đầu vào. value là dòng dữ liệu chứa thông tin của một điểm, được biểu diễn dưới dạng chuỗi văn bản.
* Đầu ra: centroidId là id của tâm cụm gần nhất cho điểm. pointInput là điểm dữ liệu với thông tin của các thuộc tính và số lượng điểm trong cụm.

**3.1.6.4. Xây dựng lớp KCombiner**

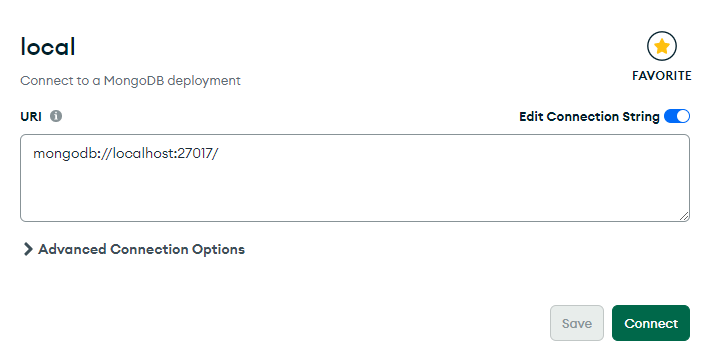
* Thực hiện tính toán tổng các điểm dữ liệu thuộc cùng một cụm và ghi thông tin điểm vào tập tin đầu ra, để lưu trữ các điểm thuộc cùng một cụm vào các tập tin riêng biệt.
* Đầu vào : centroidId là khóa đại diện cho id của tâm cụm, và points là danh sách các điểm thuộc tâm cụm.
* Đầu ra : centroidId vẫn là khóa đại diện cho id của tâm cụm, và ptSum là tổng của các điểm trong cùng một cụm sau khi đã kết hợp.

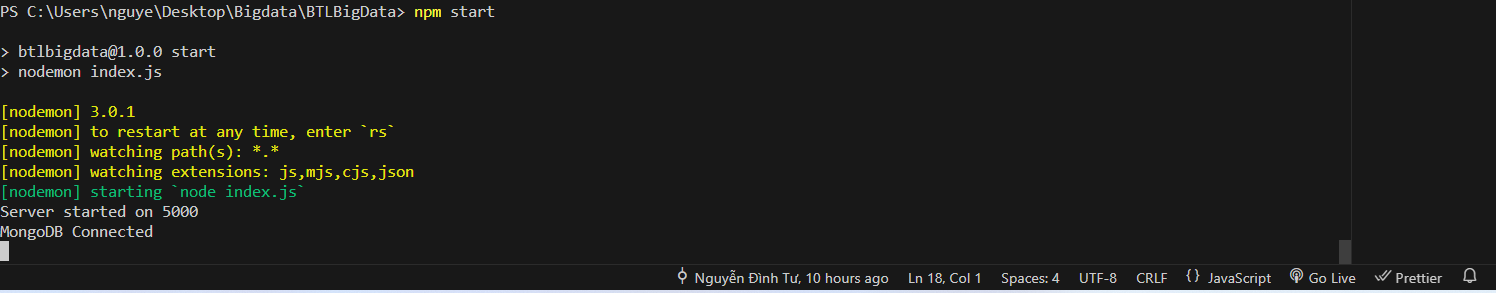
**3.1.6.5. Xây dựng lớp KReducer**

* Tính toán trung bình của các điểm trong cùng một cụm và tìm ra các điểm tâm cụm mới.
* Đầu vào: centroidId là khóa đại diện id của tâm cụm. partialSums là danh sách các điểm thuộc cụm tương ứng đã được tổng hợp từ bước KCombiner.
* Đầu ra: newCentroidId là khóa đại diện id của tâm cụm mới. newCentroidValue là giá trị của tâm cụm mới.

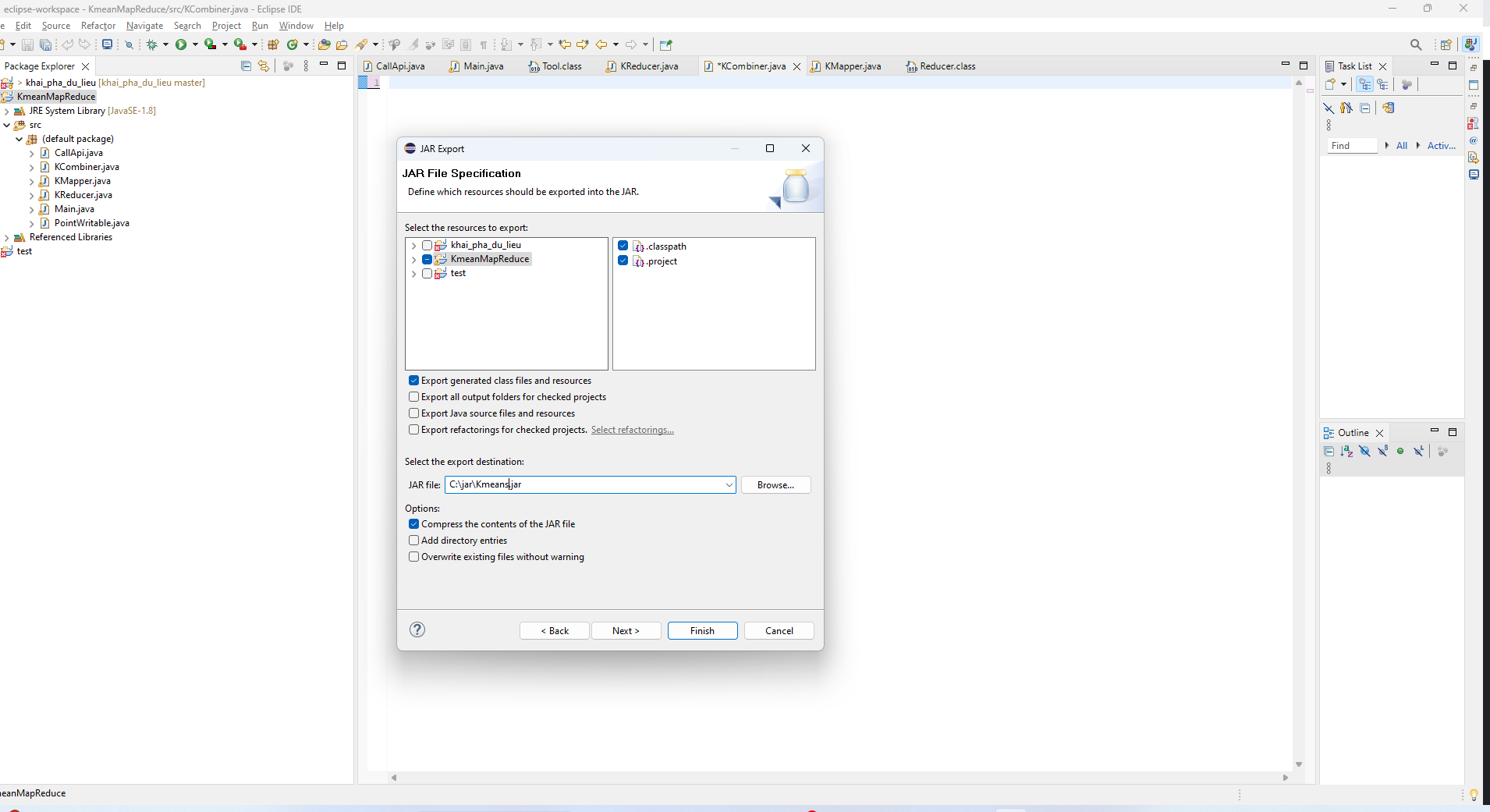
## 3.2. Kết quả bài toán

### **3.2.1. Demo cài đặt chương trình**

* Kết nối tới MongoDB:
* Di chuyển vào thư mục chứa các API thao tác với MongoDB và giao diện của ứng dụng dự đoán nhóm khách hàng, thực hiện câu lệnh ‘*npm start’*  để khởi chạy các API và kết nối đến MongoDB:



* Thực hiện build project thành file jar:

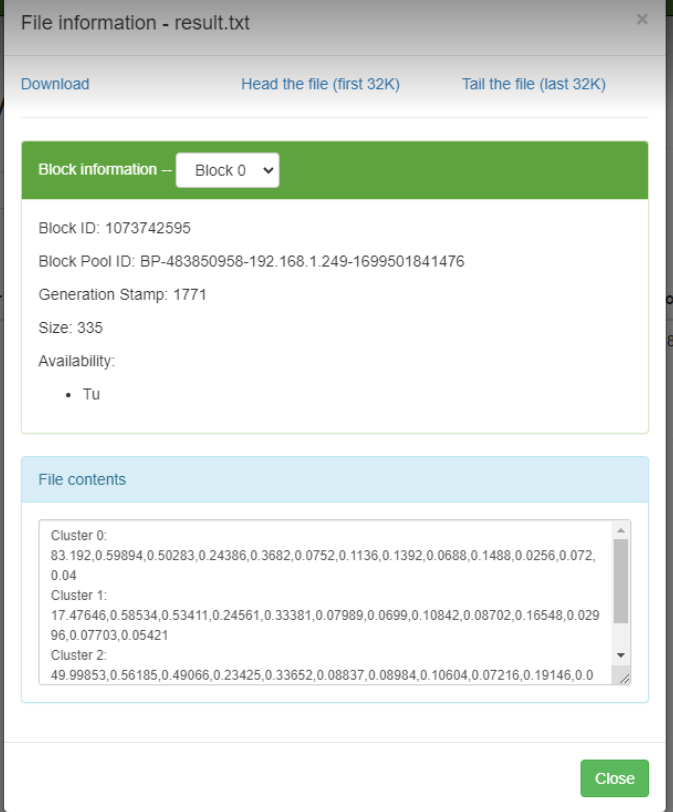


* Chạy file jar bằng câu lệnh

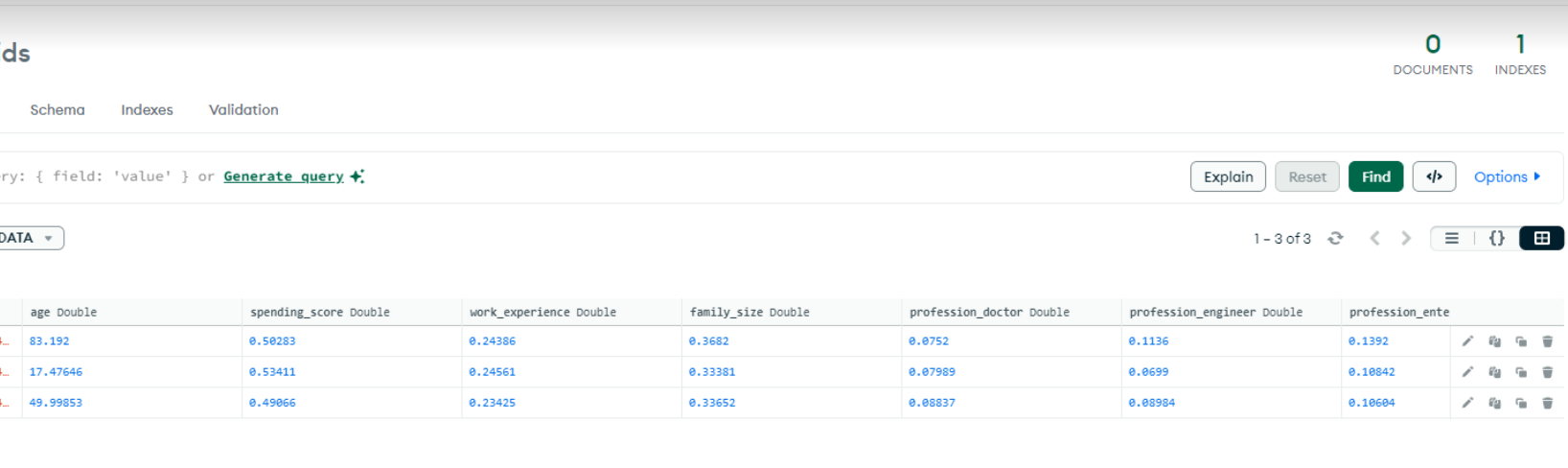


### **3.2.2. Kết quả**

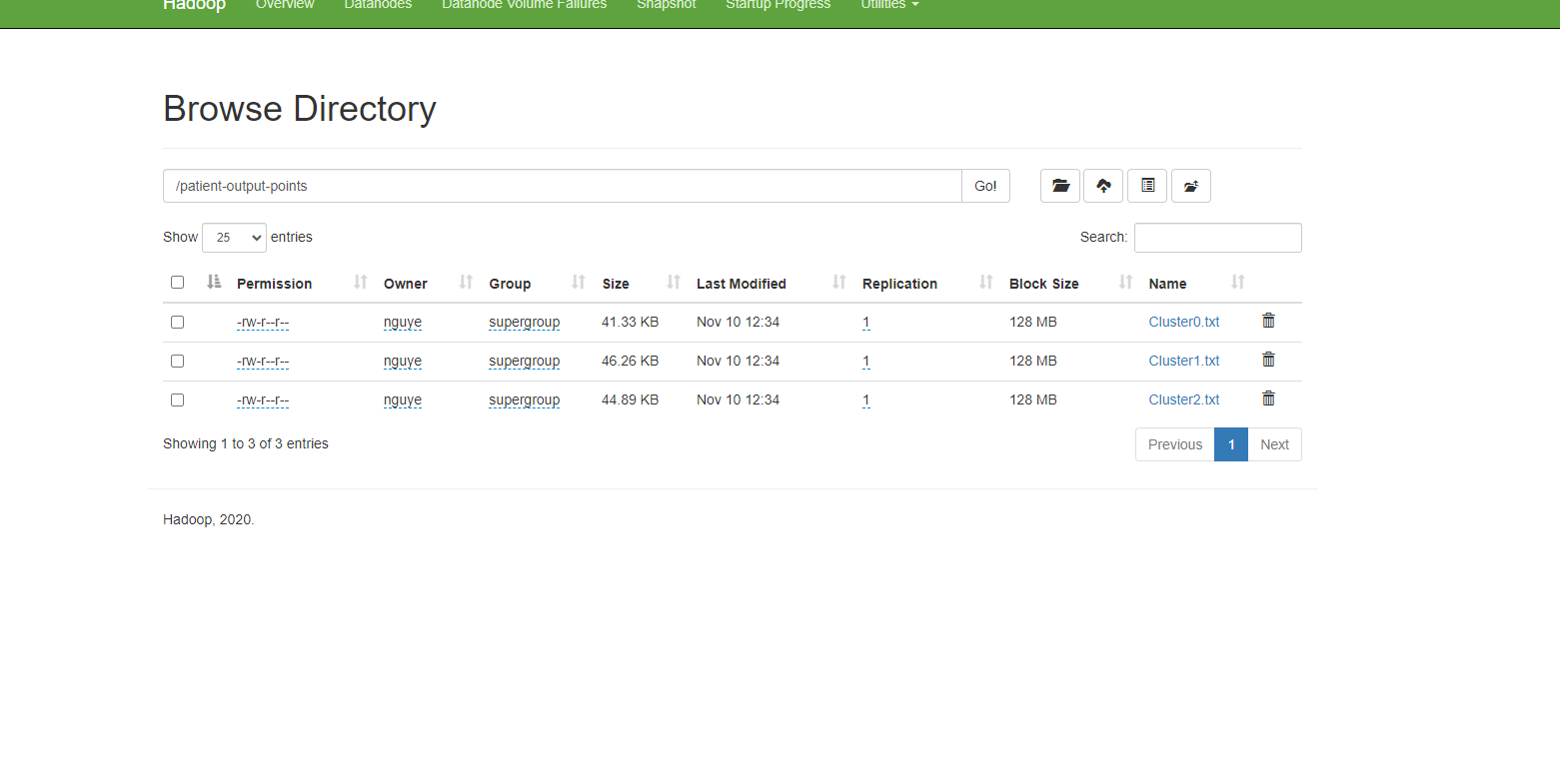
* Kết quả in ra file *result.txt* 3 tâm cụm cuối cùng và dữ liệu của 3 tâm cụm được lưu vào trong MongoDB

****

* Dữ liệu 3 tâm cụm được lưu vào MongoDB:

****

* In ra 3 file kết quả, mỗi file chứa dữ liệu các điểm thuộc mỗi tâm cụm



### **3.2.3. Demo ứng dụng dự đoán nhóm khách hàng**

Từ dữ liệu của 3 tâm cụm đã tìm thấy, có thể nhận xét những đặc điểm của mỗi cụm như sau:

* Cụm 0: nhóm ‘**Già**’, có đặc điểm như độ tuổi trung bình cao, điểm chi tiêu trung bình, có mức thu nhập cao, kinh nghiệm làm việc thấp
* Cụm 1: nhóm ‘**Trẻ**’, có đặc điểm như độ tuổi trung bình thấp, điểm chi tiêu trung bình cao, có mức thu nhập trung bình, kinh nghiệm làm việc cao
* Cụm 2: nhóm ‘**Trung niên’**, có đặc điểm như độ tuổi trung bình, điểm chi tiêu trung bình thấp, có mức thu nhập thấp, kinh nghiệm làm việc trung bình

Từ những phân tích đặc điểm dữ liệu của các cụm trên, xây dựng ứng dụng dự đoán nhóm phân khúc khách hàng:

* Khởi chạy [localhost:5000](localhost:3000)
* Nhập dữ liệu của một khách hàng và ấn nút ‘Dự Đoán’
* Dữ liệu sau đó sẽ được chuẩn hóa theo đúng dạng đã sử dụng để phân cụm
* Sau đó thực hiện truy vấn đến 3 tâm cụm trong MongoDB rồi thực hiện tính toán khoảng cách từ điểm dữ liệu nhập vào tới 3 tâm cụm
* Tìm ra tâm cụm gần nhất và hiển thị những đặc trưng của tâm cụm đó lên màn hình



# CHƯƠNG 4: KẾT LUẬN

Trong bối cảnh của sự phát triển mạnh mẽ của thông tin và công nghệ thông

tin, việc xử lý khối lượng lớn dữ liệu để trích xuất thông tin và tri thức có giá trị đang

trở thành một yếu tố quan trọng trong mọi lĩnh vực. Sự kết hợp giữa khả năng tự

động, tốc độ và độ chính xác trong việc trích xuất tri thức từ dữ liệu đã trở thành mục

tiêu hàng đầu, và trong ngữ cảnh này, phân tích dữ liệu lớn đóng vai trò quan trọng.

Lĩnh vực kinh doanh buôn bán cũng không nằm ngoài xu hướng này. Phân tích dữ liệu lớn trong kinh doanh buôn bán giúp chúng ta rút ra tri thức liên quan đến tiềm năng khách hàng và những nhu cầu thiết yếu của người mua hàng. Tuy việc áp dụng phân tích dữ liệu trong kinh doanh buôn bán còn đối mặt với một số thách thức như việc lựa chọn thuật toán phù hợp, yêu cầu lượng dữ liệu lớn về khách hàng, và tốc độ xử lý tính toán. Tuy nhiên, điều này đã góp phần hỗ trợ quá trình phân loại và đánh giá tiềm năng khách hàng trở nên nhanh chóng hơn và cung cấp thêm độ chính xác cho các doanh nghiệp, các nhà kinh doanh trong việc phân loại và đnahs giá tiềm năng khách hàng.

Tổng kết lại, khai phá dữ liệu đóng vai trò quan trọng trong việc trích xuất tri thức từ dữ liệu khách hàng, góp phần cải thiện quá trình đánh giá tiềm năng khách hàng và tạo điều kiện thuận lợi cho các doanh nghiệp và các nhà kinh doanh trong việc đánh giá và tìm hiểu khách hàng để xây dựng và phát triển mô hình kinh doanh.

TÀI LIỆU THAM KHẢO

* Slide bài giảng môn Phân tích dữ liệu lớn - TS. Nguyễn Tu Trung, TS. Trần Mạnh Tuấn - Trường ĐH Thủy Lợi
* Tập dữ liệu khách hàng: [Shop Customer Clustering | Kaggle](https://www.kaggle.com/code/utkarshsaxenadn/shop-customer-clustering/input)
* Viblo.asia: <https://viblo.asia/p/gioi-thieu-ve-framework-ma-nguon-mo-apache-hadoop-1ZnbRlNlG2Xo>