**TRƯỜNG ĐẠI HỌC THỦY LỢI**

**KHOA CÔNG NGHỆ THÔNG TIN**

**- - - 🙞 🕮** **🙜 - - -**

**BÁO CÁO PHÂN TÍCH DỮ LIỆU LỚN:**

**Đề tài: Áp dụng mô hình MapReduce vào thuật toán Kmeans để phân cụm rượu vang**

Giảng viên hướng dẫn: Nguyễn Đắc Phương Thảo

Nhóm sinh viên thực hiện: Nhóm 16

1. Trần Thị Thu Nguyệt
2. Nguyễn Hồng Nhật
3. Phạm Minh Quân
4. Hoàng Đức Nhuận
5. Đinh Quang Thao

**Hà Nội, năm 2024**

Mục lục

[I. Mô tả bài toán 3](#_Toc187782492)

[1. Khái quát vấn đề: 3](#_Toc187782493)

[2. Phạm bi dự án: 3](#_Toc187782494)

[II. Mô tả bộ dữ liệu 4](#_Toc187782495)

[III. Một số kiến thức cơ bản 5](#_Toc187782496)

[1. Tiền xử lý dữ liệu 5](#_Toc187782497)

[2. Mô hình MapReduce 6](#_Toc187782498)

[3. Thuật toán Kmeans 7](#_Toc187782499)

[IV. MAPREDUCE HOÁ KMEANS 9](#_Toc187782500)

[3.1. MapReduce hoá để huấn luyện 9](#_Toc187782501)

[V. Thực Nghiệm 12](#_Toc187782502)

[2. Tạo file chứa dữ liệu 12](#_Toc187782503)

[3. Tạo các class xử lý nhiệm vụ phân cụm Kmean 13](#_Toc187782504)

[4. Tạo file Jar 16](#_Toc187782505)

[5. Thử nghiệm 17](#_Toc187782506)

[KẾT LUẬN 18](#_Toc187782507)

[TÀI LIỆU THAM KHẢO 19](#_Toc187782508)

## Mô tả bài toán

### Khái quát vấn đề:

* Ngày nay, với sự phát triển của ngành công nghiệp rượu vang, việc phân tích và đánh giá chất lượng rượu trở thành một bài toán quan trọng, đòi hỏi các công cụ hiệu quả để xử lý khối lượng dữ liệu lớn về các chỉ số hóa lý của rượu. Việc tích hợp thuật toán K-means với mô hình MapReduce mang lại giải pháp mạnh mẽ để phân cụm và phân tích dữ liệu chất lượng rượu này.
* MapReduce hỗ trợ phân phối dữ liệu lớn, bao gồm các chỉ số hóa lý như độ axit, độ pH, hàm lượng cồn và đường dư, trên nhiều nút tính toán. Điều này giúp tăng tốc độ xử lý và đảm bảo khả năng phân tích hiệu quả trên các tập dữ liệu lớn. Trong bước Reduce, thuật toán K-means được áp dụng để phân cụm rượu vang theo các nhóm chất lượng tương đồng, từ đó xác định các đặc điểm hóa lý quan trọng ảnh hưởng đến chất lượng rượu.
* Với cách tiếp cận này, việc sử dụng thuật toán K-means cùng MapReduce không chỉ rút ngắn thời gian phân tích dữ liệu mà còn cung cấp những thông tin giá trị về mối quan hệ giữa các chỉ số hóa lý và chất lượng rượu. Điều này mở ra tiềm năng ứng dụng trong việc tối ưu hóa quy trình sản xuất rượu vang và cải thiện chất lượng sản phẩm, đáp ứng tốt hơn nhu cầu của người tiêu dùng.

### Phạm bi dự án:

- **Phạm vi**: Dự án này tập trung vào việc phân cụm và phân tích chất lượng rượu vang đỏ dựa trên các chỉ số hóa lý. Dữ liệu đầu vào bao gồm các thông số như độ axit, độ pH, hàm lượng cồn, đường dư, và các yếu tố hóa học khác. Thuật toán K-means sẽ được kết hợp với mô hình MapReduce để xử lý và phân tích dữ liệu lớn, nhằm phân nhóm rượu theo chất lượng và xác định các đặc điểm chính ảnh hưởng đến chất lượng rượu.

## Mô tả bộ dữ liệu

* **Nguồn gốc:** Dữ liệu này được trích xuất từ các mẫu rượu vang đỏ sản xuất tại Bồ Đào Nha (vùng Vinho Verde).
* **Kích thước:** Dữ liệu gồm 1,599 hàng (mẫu) và 12 cột (11 thuộc tính + 1 nhãn).
* **Thuộc tính (Features)**

Tất cả các cột, trừ cột **quality**, là các thuộc tính hóa học liên tục. Dưới đây là danh sách và ý nghĩa của từng thuộc tính:

| **Thuộc tính** | **Ý nghĩa** | **Phạm vi giá trị** |
| --- | --- | --- |
| **fixed acidity** | Hàm lượng axit cố định (tartaric acid) góp phần tạo độ chua. | Khoảng: 4.6 - 15.9 |
| **volatile acidity** | Hàm lượng axit dễ bay hơi (acetic acid), ảnh hưởng đến mùi vị. | Khoảng: 0.12 - 1.58 |
| **citric acid** | Hàm lượng axit citric, mang lại độ tươi và vị chua nhẹ. | Khoảng: 0.0 - 1.0 |
| **residual sugar** | Lượng đường còn sót lại sau khi lên men, tạo vị ngọt. | Khoảng: 0.9 - 15.5 |
| **chlorides** | Hàm lượng clo, có thể ảnh hưởng đến vị mặn. | Khoảng: 0.012 - 0.611 |
| **free sulfur dioxide** | Lưu huỳnh dioxide tự do, giúp bảo quản rượu. | Khoảng: 1 - 72 |
| **total sulfur dioxide** | Tổng lượng lưu huỳnh dioxide. | Khoảng: 6 - 289 |
| **density** | Khối lượng riêng của rượu, liên quan đến hàm lượng cồn và đường. | Khoảng: 0.990 - 1.003 |
| **pH** | Độ pH, đo độ axit/bazơ của rượu. | Khoảng: 2.74 - 4.01 |
| **sulphates** | Hàm lượng sulfat, góp phần vào hương vị đắng hoặc chát. | Khoảng: 0.33 - 2.0 |
| **alcohol** | Nồng độ cồn (%). | Khoảng: 8.4 - 14.9 |

* **Mục tiêu:** Phân cụm - nhóm các loại rượu vang có đặc điểm tương tự nhau dựa trên các thuộc tính hóa học của chúng.
* **Nhận xét bộ dữ liệu:**
* Kích thước bộ dữ liệu 1599 mẫu chứa nhiều thuộc tính quan trọng đủ lớn để phân cụm hiệu quả.
* Các giá trị đều là số liệu liên tục. Tuy nhiên, chênh lệch về đơn vị giữa các thuộc tính khá lớn: giá trị **citric acid** chi dao động từ 0 - 1, mà giá trị của **total sulfur dioxide** thì khoảng từ 6 - 289. Điều này có thể ảnh hưởng đến kết quả đánh giá của thuật toán Kmeans

## Một số kiến thức cơ bản

### Tiền xử lý dữ liệu

#### 1.1 Làm sạch dữ liệu

- Là quá trình xử lý dữ liệu bị thiếu, nhận diện phần tử biên và giảm thiểu nhiễu, xử lý dữ liệu không nhất quán

* Đối với bộ dữ liệu Quality wine: loại bỏ cột nhãn **quality** cho mục đích phân cụm.

#### 1.2. 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
* Liên quan đến cấu trúc và tính không thuần nhất (heterogeneity) về ngữ nghĩa (semantics) của dữ liệu
* Hỗ trợ việc giảm và tránh dư thừa và không nhất quán về dữ liệu => cải thiện tính chính xác và tốc độ quá trình khai phá dữ liệu

#### 1.3. Biến đổi dữ liệu (chuẩn hóa dữ liệu)

* Là quá trình biến đổi hay kết hợp dữ liệu vào những dạng thích hợp cho quá trình khai phá dữ liệu
* Bao gồm:
  + Làm trơn dữ liệu
  + Kết hợp dữ liệu
  + Tổng quát hóa
  + Chuẩn hóa
  + Xây dựng thuộc tính/đặc tính
  + Thu giảm dữ liệu
* Đối với bộ dữ liệu Quality wine: chuẩn hóa các đặc trưng của dữ liệu về cùng một phân phối có giá trị trung bình (mean) bằng 0 và độ lệch chuẩn (standard deviation) bằng 1.

### Mô hình MapReduce

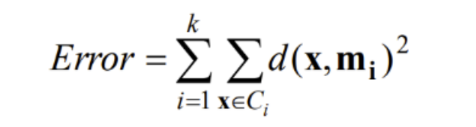
* MapReduce là một mô hình lập trình được sử dụng để xử lý các tập dữ liệu lớn phân tán trên một cụm máy tính. MapReduce được phát triển bởi Google và sau đó được Apache Hadoop sử dụng để xử lý dữ liệu trong Hadoop Distributed File System (HDFS).
* **MapReduce** nhận dữ liệu vào và ra đều là một tập hợp lớn các cặp khóa/giá trị. Mô hình này dựa trên 2 bước cơ bản:
* **Map**: nhận dòng dữ liệu thô và xử lý, tạo ra các cặp khóa/giá trị. Thư viện MapReduce sẽ tự động nhóm các giá trị có cùng khóa lại với nhau, và chuyển chúng đến các hàm Reduce.
* **Reduce**: nhận một khóa và các giá trị tương ứng với khóa đó dưới dạng list, sẽ tiếp tục xử lý các giá trị này để tạo thành kết quả cuối cùng tùy theo yêu cầu công việc. Thư viện MapReduce tiếp nhận các kết quả này và xuất ra kết quả cuối cùng của bài toán dưới các dạng dữ liệu khác nhau.
* **Ưu điểm của MapReduce:**
* Mapreduce có thể dễ dàng xử lý tất cả mọi bài toán có lượng dữ liệu khổng lồ nhờ khả năng tính toán và tác vụ phân tích phức tạp.
* Mapreduce có khả năng chạy song song trên các máy tính có sự phân tán khác nhau với khả năng hoạt động độc lập kết hợp với việc phân tán và xử lý các lỗi kỹ thuật để mang đến hiệu quả cao cho toàn bộ hệ thống.
* Mapreduce có khả năng thực hiện được trên đa dạng nhiều loại ngôn ngữ lập trình khác nhau như ngôn ngữ C/C++, Java, Perl, Python, Ruby,... cùng với những thư viện hỗ trợ tương ứng.
* Mã độc trên Internet ngày càng nhiều khiến cho việc xử lý các đoạn mã độc này trở nên phức tạp và tiêu tốn nhiều thời gian hơn. Do đó, Mapreduce đang dần hướng quan tâm nhiều hơn cho việc phát hiện các mã độc để có thể nhanh chóng xử lý các đoạn mã độc đó. Nhờ đó, hệ điều hành được đảm bảo vận hành trơn tru với tính bảo mật cao nhất.

### Thuật toán Kmeans

#### Khái niệm:

* Là một trong các thuật toán phân cụm không giám sát đơn giản và điển hình nhất
* Do MacQueen đề xuất trong lĩnh vực thống kê năm 1967
* Được ứng dụng rộng rãi cho các bài toán như phân loại khách hàng, phân tích dữ liệu văn bản, phát hiện bất thường, nén dữ liệu hình ảnh, phân tích gen...

#### Các bước chính của thuật toán Kmeans:

* Bước 1: **Khởi tạo**: Chọn số lượng nhóm (K) bạn muốn phân chia dữ liệu. Chọn ngẫu nhiên K điểm dữ liệu làm các "tâm nhóm" ban đầu.
* Bước 2**: Gán điểm dữ liệu vào nhóm**: Đối với mỗi điểm dữ liệu, tính toán khoảng cách (thường là khoảng cách Euclid) từ điểm đó đến các tâm nhóm và gán điểm vào nhóm có tâm gần nhất.
* Bước 3**: Cập nhật tâm nhóm**: Sau khi gán tất cả các điểm dữ liệu vào nhóm, tính lại tâm mới của mỗi nhóm bằng cách lấy trung bình của tất cả các điểm trong nhóm đó.
* Bước 4**: Lặp lại**: Tiếp tục bước 2 và 3 cho đến khi các tâm nhóm không thay đổi nữa hoặc thay đổi rất ít (đạt được sự hội tụ).
* **K-means Hội Tụ nếu**:
* Rất ít điểm dữ liệu được gán lại vào các cụm mới, hoặc
* Các centroid không thay đổi đáng kể, hoặc
* Tổng sau đây không thay đổi đáng kể:

Trong đó:

Ci là cụm thứ i;

mi là centroid của cụm Ci

d là khoảng cách Euclide giữa 2 đối tượng

#### Nhận xét:

* Ưu điểm của K-means:

1. **Đơn giản và dễ hiểu**: dễ triển khai và dễ hiểu, phù hợp cho người mới bắt đầu
2. **Tính toán nhanh**: Thuật toán chạy nhanh với các bộ dữ liệu lớn, đặc biệt khi số nhóm K không quá lớn.
3. **Phân nhóm tốt trong nhiều trường hợp**: Hiệu quả khi các nhóm có hình dạng gần hình cầu và phân bố đều.
4. **Chạy trên dữ liệu lớn**: Dễ dàng xử lý tập dữ liệu lớn vì chỉ cần tính khoảng cách và trung bình.

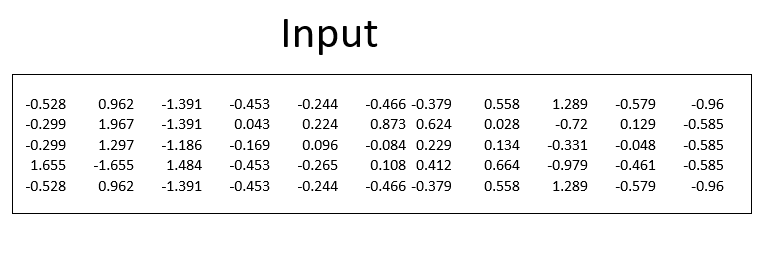
* Nhược điểm của K-means:

1. **Cần xác định trước số nhóm K**: Phải xác định số nhóm K trước, điều này có thể khó khi không có thông tin về nhóm.
2. **Dễ bị ảnh hưởng bởi giá trị ngoại lệ (outliers)**: Các giá trị ngoại lệ có thể làm lệch tâm nhóm và ảnh hưởng đến kết quả.
3. **Khó xử lý các nhóm có hình dạng phức tạp**:Không tốt với các nhóm có hình dạng không phải hình cầu
4. **Thuật toán có thể hội tụ vào tối ưu địa phương**: Khởi tạo ngẫu nhiên có thể dẫn đến tối ưu địa phương, không phải tối ưu toàn cục.
5. **Yêu cầu tính toán khoảng cách giữa tất cả các điểm**: Cần tính khoảng cách giữa các điểm và tâm nhóm, đặc biệt khi dữ liệu có chiều cao.
6. **Không xử lý được dữ liệu không gian không gian:** Khó phân nhóm khi có quá nhiều chiều dữ liệu do "curse of dimensionality".

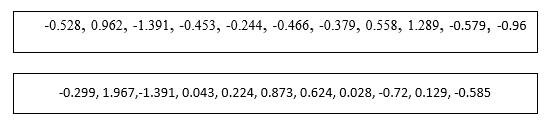
## MAPREDUCE HOÁ KMEANS

### 3.1. MapReduce hoá để huấn luyện

Bước 1: Input - Thu thập dữ liệu cần phân cụm

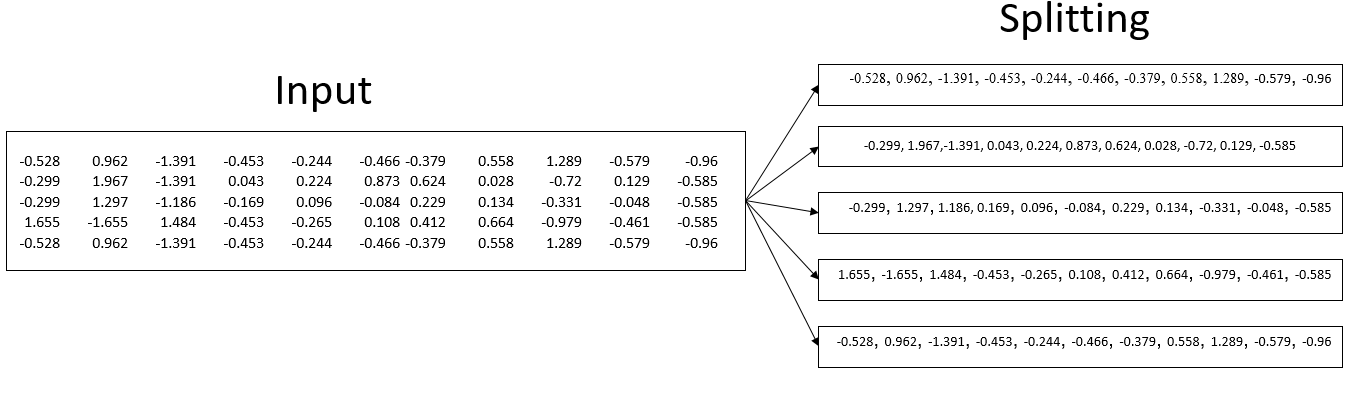


* Khởi tạo tâm cụm:



Bước 2: Splitting - Chia dữ liệu thành từng dòng để xử lý

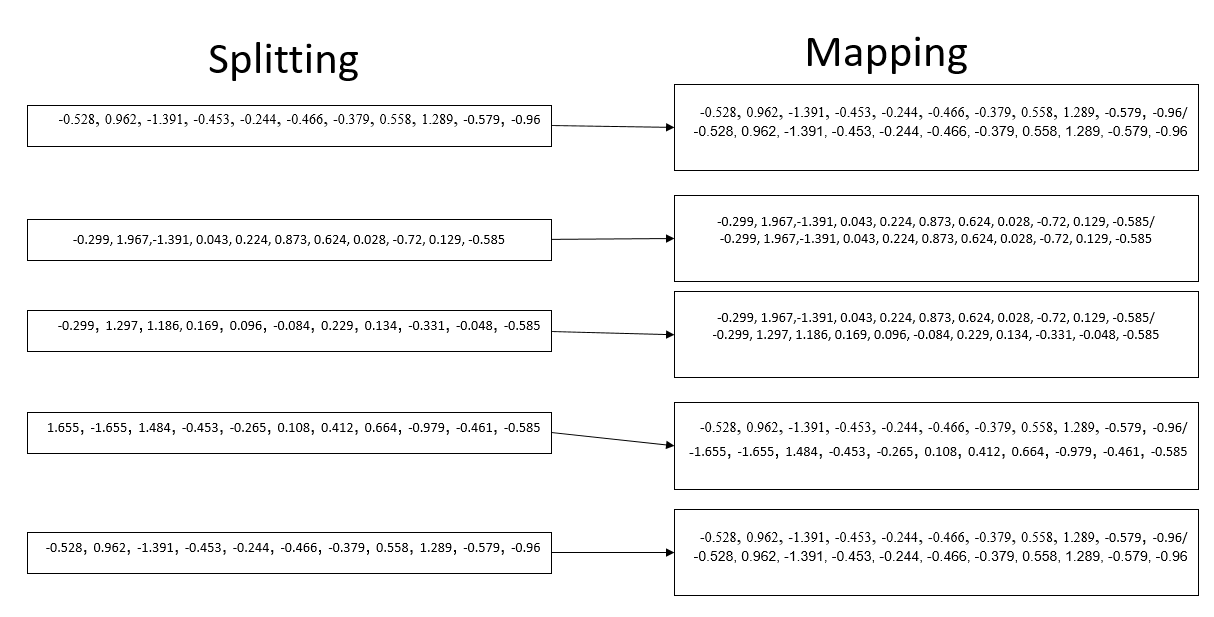
– Chia tập dữ liệu thành 1 hoặc một nhóm các điểm



Bước 3: Mapping

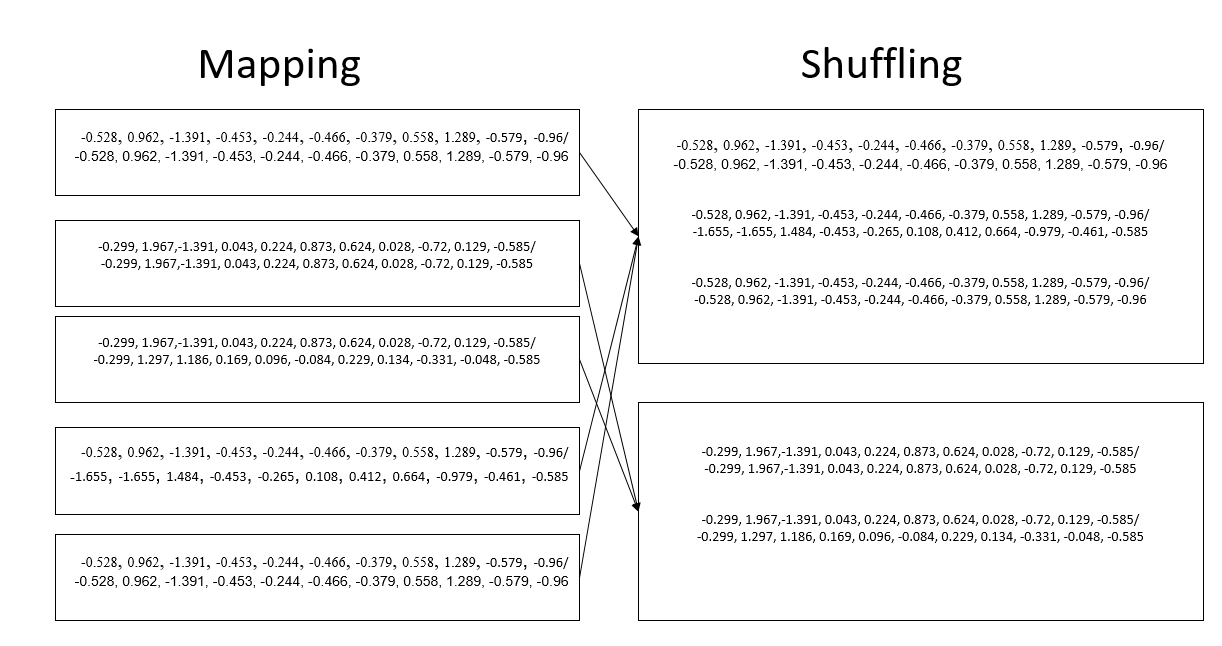
– Tính khoảng cách các điểm đến các tâm cụm

– Tạo cặp key/value, với key là trọng tâm gần nhất, value là toạ độ của điểm đang xét



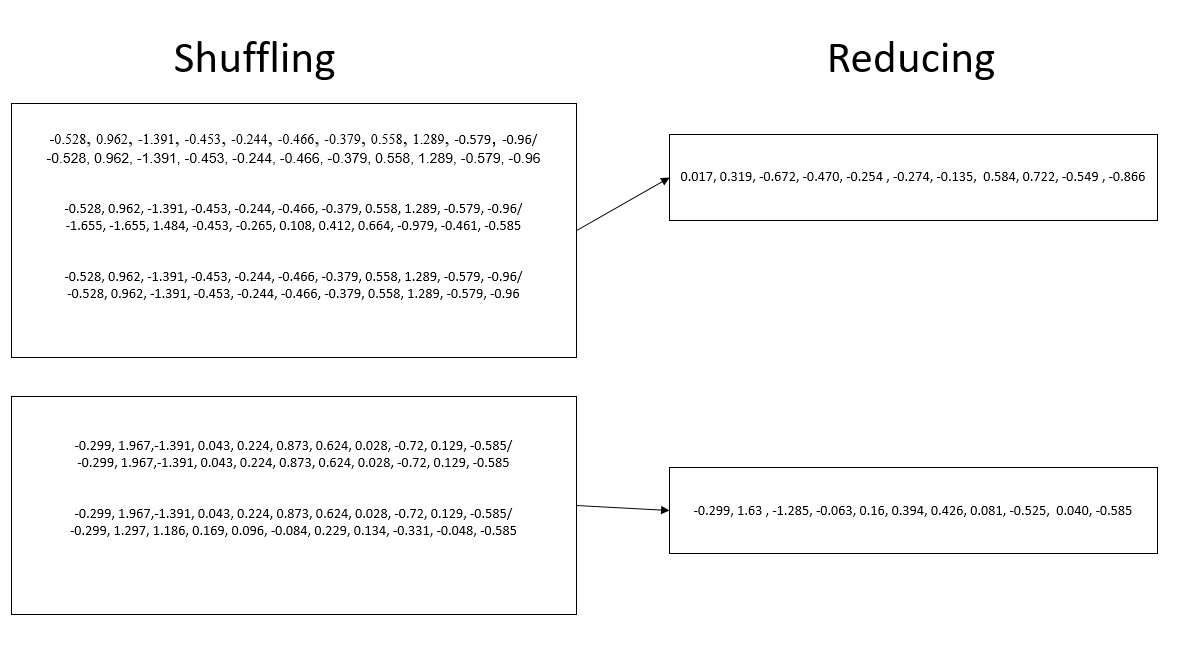
Bước 4: Shuffling

– Gom các dữ liệu có cùng key thành 1 nhóm



Bước 5: Reducing

– Tính toạ độ tâm mới bằng cách tính trung bình cộng các điểm đã được gán trọng tâm



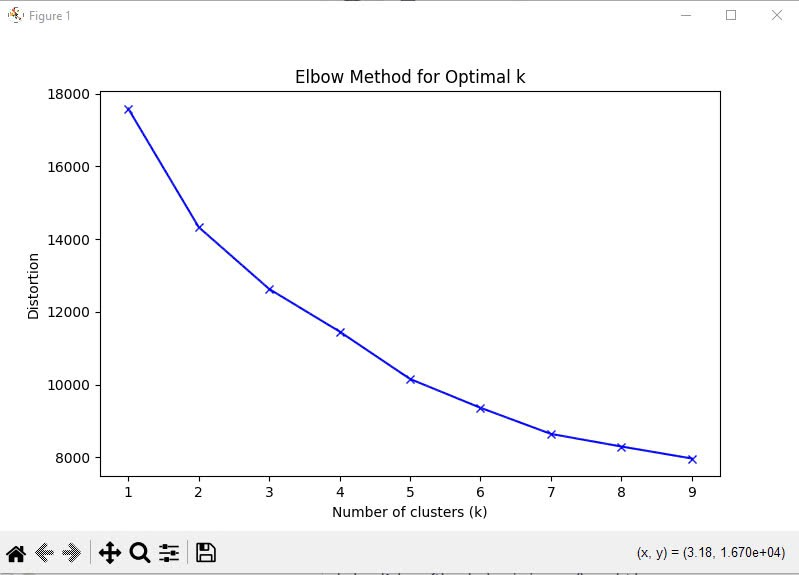
Bước 6: Lặp lại và kiểm tra điểm dừng

– Gán tâm mới vào tâm khởi tạo ban đầu và thực hiện một vòng lặp mới

– Điều kiện dừng khi các tâm cũ không đổi so với tâm mới

## Thực Nghiệm

1. **Lựa chọn K bằng phương pháp Elbow**



*Ảnh thể hiện sự biến động của K bằng phương pháp Elbow*

- Nhận xét: từ biểu đồ trên ta có thể thấy khí K = 3 độ dốc của đường cong giảm đáng kể và bắt đầu trở nên "phẳng" hơn. Đây được gọi là điểm gập khuỷu tay, độ giảm của Distortion không còn lớn như trước.

- Kết luận: Ta lựa chọn K = 3 là hợp lí với dữ liệu này.

### Tạo file chứa dữ liệu

Bước 1: Tạo thư mục đầu vào trong hdfs

hdfs dfs -mkdir/k\_input\_btl

Bước 2: Đẩy file wineQuality.csv vào folder k\_input\_btl vừa tạo

hadoop fs - put D:\bigData\wineQuality.csv /k\_input\_btl

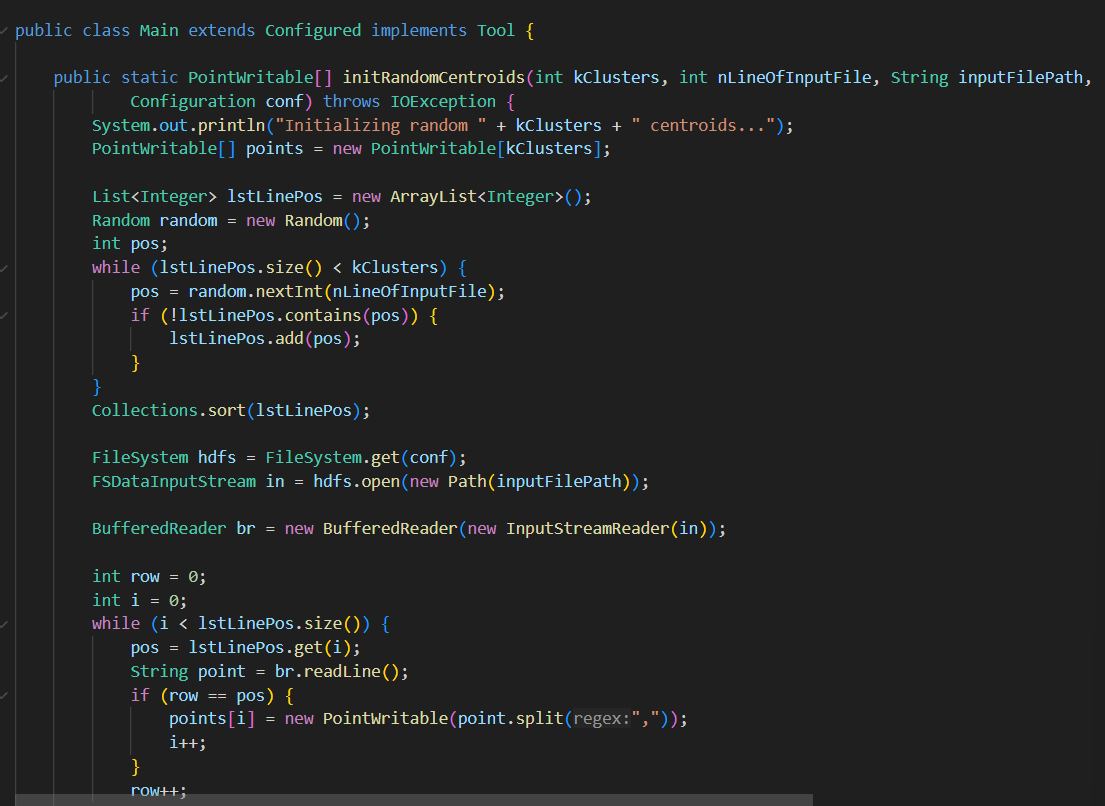
Bước 3: Tạo project đặt tên là KmeansMapReduce

Bước 4: Thêm thư viện cần thiết để chạy MapReduce

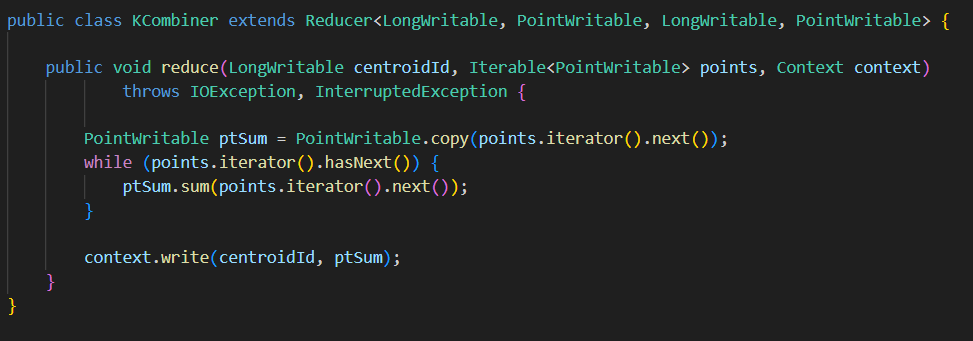
Bước 5: Tạo các class xử lý nhiệm vụ mapper, combiner, reducer mô hình Kmeans

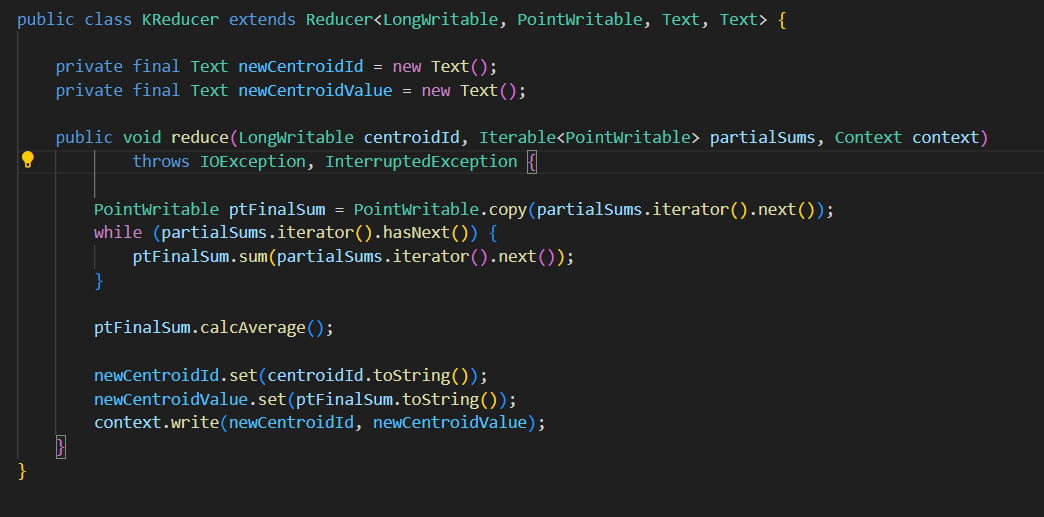
Double click vào project **KmeanMapReduce**, chuột phải vào **src** và chọn **New > Class**

### Tạo các class xử lý nhiệm vụ phân cụm Kmean

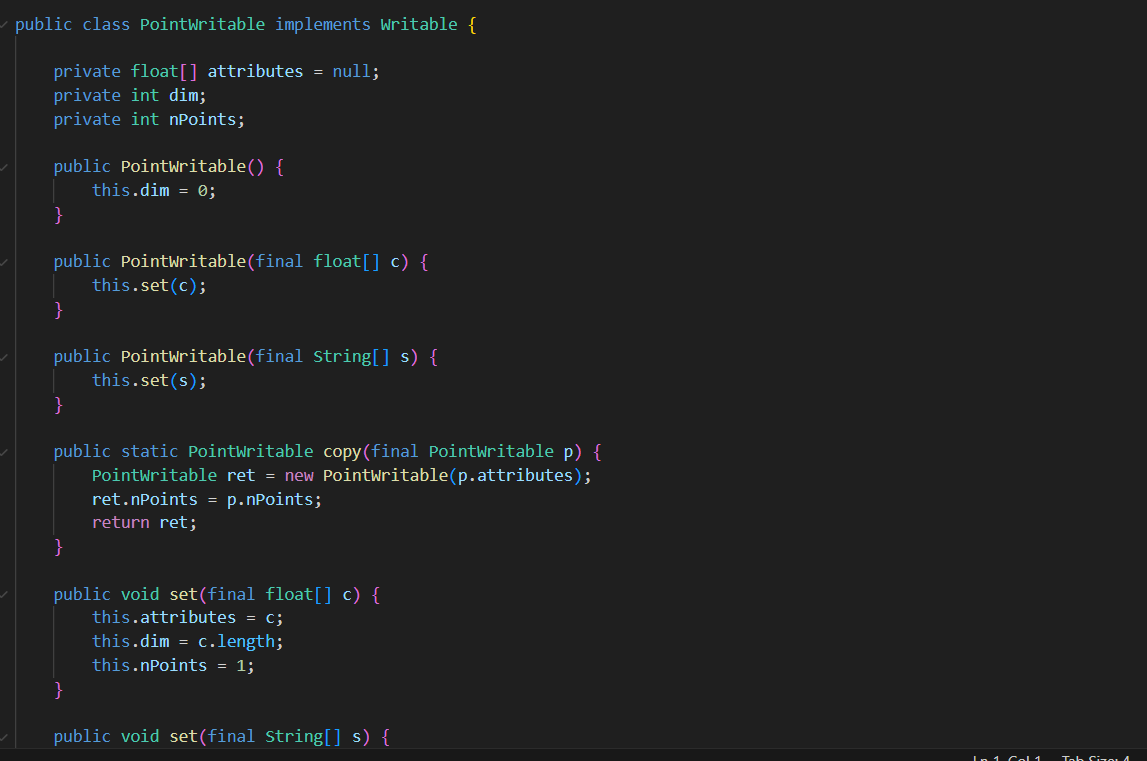
- Tạo class Main:

- Tạo class KCombiner:



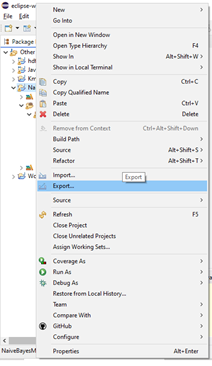
- Tạo class KReducer:   


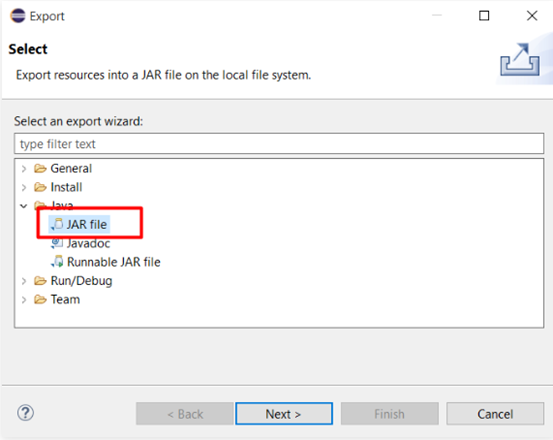
- Tạo class PointWritable:



### Tạo file Jar

* Chuột phải vào project KmeansMapReduce chọn Export

****

* Chọn **Java > JAR File** rồi bấm **Finish**

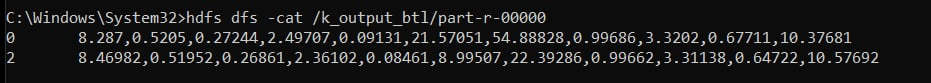
### Thử nghiệm

* Chạy chương trình KmeansMapReduce chạy lệnh :

hadoop jar C:jar\kMeans\_btl.jar -Din /k\_input\_btl/wineQuality.csv -Dlines 1600 -Dresult result.txt -Dmaxloop 50 -Dk 3 -Dthresh 0.0001 -DNumReduceTask 2 -Dout /k\_output\_btl

* Xem kết quả sử dụng lệnh

hdfs dfs -cat /k\_output\_btl/part-r-00000



## KẾT LUẬN

Qua quá trình nghiên cứu và triển khai, chúng ta đã đạt được một cái nhìn tổng quan và chi tiết về việc áp dụng MapReduce trong thuật toán K-means, từ việc tiền xử lý dữ liệu, xây dựng mô hình đến thực nghiệm trên dữ liệu cụ thể. Dưới đây là những điểm nổi bật và bài học rút ra từ các phần:

1. **Mô tả bộ dữ liệu**: Bộ dữ liệu được sử dụng đã được phân tích và làm rõ đặc điểm, tạo nền tảng cho các bước xử lý và huấn luyện tiếp theo. Việc hiểu rõ cấu trúc và tính chất của dữ liệu đóng vai trò quan trọng trong việc lựa chọn và tối ưu hóa thuật toán.
2. **Kiến thức cơ bản và tiền xử lý**: Các kỹ thuật tiền xử lý dữ liệu, chẳng hạn như làm sạch, chuẩn hóa và chuyển đổi, đã được áp dụng nhằm chuẩn bị dữ liệu phù hợp cho thuật toán K-means. Đồng thời, kiến thức về mô hình MapReduce cũng được hệ thống hóa, giúp hiểu rõ cách chia nhỏ và xử lý dữ liệu song song.
3. **MapReduce hóa K-means**: Việc triển khai K-means theo mô hình MapReduce đã chứng minh được tính hiệu quả của phương pháp này trong việc xử lý dữ liệu lớn. Quá trình huấn luyện được thực hiện thông qua việc chia nhỏ các nhiệm vụ (tasks) và thực hiện chúng đồng thời, từ đó giảm thiểu thời gian xử lý và tăng khả năng mở rộng của hệ thống.
4. **Thực nghiệm và triển khai**: Các bước thực nghiệm, từ tạo file chứa dữ liệu, xây dựng các class xử lý đến thử nghiệm trên hệ thống, đã giúp kiểm chứng hiệu quả của phương pháp đề xuất. Việc tạo file Jar và triển khai thực tế trên hệ thống đã minh chứng tính khả thi và hiệu năng của giải pháp.

Nhìn chung, dự án đã cho thấy sự kết hợp giữa thuật toán K-means và mô hình MapReduce có thể mang lại giải pháp hiệu quả cho các bài toán phân cụm dữ liệu lớn. Tuy nhiên, vẫn cần nghiên cứu thêm để tối ưu hóa các bước tiền xử lý, giảm thiểu chi phí tính toán và cải thiện độ chính xác trong các ứng dụng thực tế. Các kết quả đạt được từ thực nghiệm là cơ sở để tiếp tục phát triển các ứng dụng khai phá dữ liệu trong tương lai.

## TÀI LIỆU THAM KHẢO

1. Bài giảng và thực hành môn dữ liệu lớn của trường Đại học Thủy lợi

2. [Link lấy dữ liệu từ Kaggle](https://www.kaggle.com/code/nimapourmoradi/red-wine-quality)