TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

VIỆN TRÍ TUỆ NHÂN TẠO

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

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

ĐỀ TÀI

THUẬT TOÁN ID3 & LẬP TRÌNH MAPREDUCE HÓA ID3 TRONG YouTube Data Analysis

Nhóm sinh viên thực hiện:

Bùi Văn Khải - 22022574

Giảng viên hướng dẫn:

MỞ ĐẦU

Trong thời đại công nghệ thông tin bùng nổ, dữ liệu lớn (Big Data) đã trở thành một tài sản quý giá đối với các doanh nghiệp và tổ chức. Việc khai thác và phân tích dữ liệu lớn không chỉ mang lại giá trị kinh tế mà còn đóng vai trò quan trọng trong các lĩnh vực như khoa học, giáo dục và giải trí. Một trong những thách thức lớn nhất khi làm việc với dữ liệu lớn là tìm ra các phương pháp hiệu quả để xử lý và phân loại thông tin trong khối lượng dữ liệu khổng lồ.

Chính vì lý do đó, em đã lựa chọn bộ dữ liệu YouTube để minh họa cho " **Thuật toán ID3 & lập trình MapReduce hóa ID3 trong phân dữ Liệu**" Bộ dữ liệu YouTube, với hàng triệu bản ghi chứa thông tin về lượt xem, danh mục và tiêu đề video, là một minh chứng thực tiễn về quy mô và độ phức tạp của dữ liệu lớn.

Báo cáo gồm 4 chương:

Chương 1: Tổng quan về dữ liệu lớn.

Chương 2: Phân lớp dữ liệu bằng thuật toán ID3.

Chương 3: MapReduce thuật toán ID3 trong phân lớp dữ liệu.

Chương 4: Kết luận và hướng phát triển.

MỤC LỤC

CHƯƠNG 1: TỔNG QUAN VỀ DỮ LIỆU LỚN

1.1 Định nghĩa.

*Theo Wikipedia, **Dữ liệu lớn (Big Data)** là một tập hợp dữ liệu có khối lượng lớn và phức tạp đến mức không thể xử lý được bằng các phương pháp truyền thống. Dữ liệu lớn thường được đặc trưng bởi **3V**:*

1. ***Volume (Khối lượng)**: Lượng dữ liệu khổng lồ được tạo ra từ nhiều nguồn khác nhau.*
2. ***Velocity (Tốc độ)**: Tốc độ dữ liệu được tạo ra và xử lý gần như thời gian thực.*
3. ***Variety (Đa dạng)**: Dữ liệu tồn tại dưới nhiều định dạng khác nhau như văn bản, hình ảnh, video, âm thanh, dữ liệu phi cấu trúc, và bán cấu trúc.*

Ngoài ra, một số học giả và tổ chức còn bổ sung thêm các đặc tính như:

- ***Veracity (Độ chính xác)**: Đề cập đến tính đáng tin cậy của dữ liệu.*
- ***Value (Giá trị)**: Giá trị mà dữ liệu mang lại thông qua phân tích và xử lý*

Dữ liệu lớn đang ngày càng có vai trò quan trọng trong các lĩnh vực khác nhau của đời sống, kinh tế và xã hội. Dưới đây là một số ý nghĩa chính:

1. **Thúc đẩy quyết định dựa trên dữ liệu (Data-driven Decision-making)**
    - Các tổ chức sử dụng dữ liệu lớn để phân tích xu hướng, hành vi của khách hàng, từ đó đưa ra các chiến lược kinh doanh hiệu quả.
2. **Cải tiến các ngành công nghiệp**
    - Trong y tế: Dữ liệu lớn hỗ trợ phân tích thông tin y tế, chẩn đoán bệnh và phát triển các liệu pháp cá nhân hóa.
    - Trong giao thông: Tối ưu hóa luồng giao thông, phân tích các mẫu di chuyển để cải thiện cơ sở hạ tầng.
3. **Phát triển trí tuệ nhân tạo và học máy (AI & Machine Learning)**
    - Dữ liệu lớn là nguồn tài nguyên chính để huấn luyện các mô hình học máy, từ đó tạo ra những công nghệ đột phá như chatbot, nhận dạng giọng nói, và xe tự lái.
4. **Tăng cường quản trị và chính sách công**
    - Chính phủ và các tổ chức xã hội sử dụng dữ liệu lớn để phân tích, dự đoán và quản lý các vấn đề xã hội như biến đổi khí hậu, dân số, và quản lý tài nguyên.
5. **Đổi mới trong lĩnh vực giải trí và truyền thông**

- Dữ liệu lớn hỗ trợ các nền tảng như YouTube, Netflix phân tích sở thích người dùng, từ đó đề xuất nội dung phù hợp và cá nhân hóa trải nghiệm.

6. **Tối ưu hóa chi phí và hiệu quả hoạt động**
   - Các doanh nghiệp sử dụng dữ liệu lớn để phát hiện lỗi, dự báo xu hướng sản xuất, và quản lý chuỗi cung ứng hiệu quả hơn.

1.2 Tổng quan về Hadoop.

Hadoop là một nền tảng phần mềm mã nguồn mở được phát triển bởi Apache Software Foundation, dùng để xử lý và lưu trữ dữ liệu lớn (Big Data) một cách hiệu quả. Hadoop được thiết kế để hoạt động trên các cụm máy tính phân tán và có khả năng xử lý dữ liệu trên quy mô lớn mà vẫn đảm bảo hiệu năng và độ tin cậy.

Hadoop nổi bật với khả năng:

- Lưu trữ lượng dữ liệu lớn thông qua hệ thống tệp phân tán (HDFS - Hadoop Distributed File System).
- Xử lý dữ liệu một cách song song bằng cách sử dụng mô hình lập trình MapReduce.

Hadoop bao gồm bốn thành phần chính:

1. HDFS (Hadoop Distributed File System)

HDFS là hệ thống tệp phân tán, cho phép lưu trữ dữ liệu lớn trên các cụm máy tính.

- **Chức năng chính**:
  - Phân chia dữ liệu thành các khối (blocks) và lưu trữ trên nhiều máy khác nhau trong cụm.
  - Cung cấp cơ chế dự phòng (replication) để đảm bảo dữ liệu an toàn khi có lỗi phần cứng.
- **Cấu trúc**:
  - **NameNode**: Quản lý thông tin metadata (thông tin về vị trí lưu trữ, cấu trúc tệp).
  - **DataNode**: Lưu trữ dữ liệu thực tế và thực hiện các yêu cầu đọc/ghi từ NameNode.

2. YARN (Yet Another Resource Negotiator)

YARN là khung quản lý tài nguyên trong Hadoop. Nó tách biệt phần xử lý và quản lý tài nguyên, giúp Hadoop có khả năng mở rộng linh hoạt.

- **Chức năng chính**:
  - Phân bổ tài nguyên cho các ứng dụng Hadoop.
  - Theo dõi trạng thái và hiệu suất của các ứng dụng.
- **Cấu trúc**:
  - **ResourceManager**: Quản lý tài nguyên của toàn bộ cụm.
  - **NodeManager**: Theo dõi tài nguyên trên từng máy trong cụm.

3. MapReduce

MapReduce là mô hình lập trình giúp xử lý dữ liệu song song trên các cụm máy tính.

- **Chức năng chính**:
  - **Map**: Phân chia dữ liệu thành các cặp khóa-giá trị.
  - **Reduce**: Tổng hợp và xử lý dữ liệu từ các đầu ra của Map.
- **Ưu điểm**:
  - Tự động hóa việc phân phối dữ liệu và xử lý song song.
  - Tăng tốc độ xử lý dữ liệu lớn.

4. Hadoop Common

Hadoop Common là tập hợp các thư viện và tiện ích cần thiết cho các thành phần khác của Hadoop.

- **Chức năng chính**:
  - Cung cấp API để hỗ trợ phát triển ứng dụng trên Hadoop.
  - Hỗ trợ giao tiếp giữa các thành phần trong hệ sinh thái Hadoop.

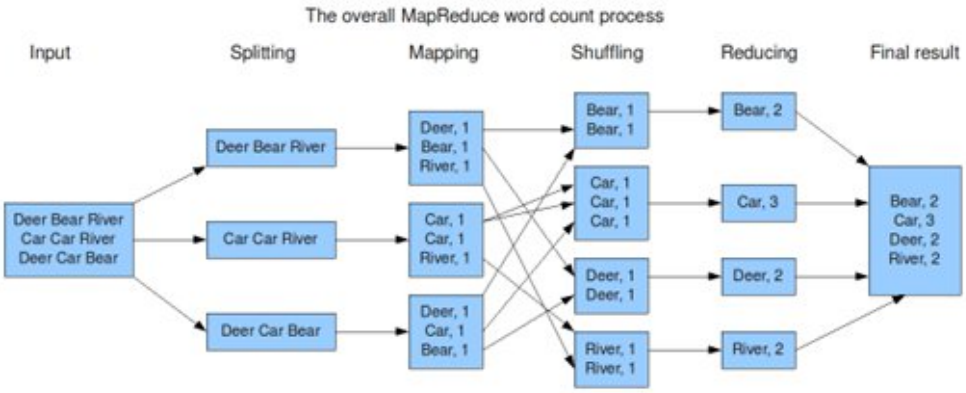
1.5 *Tổng quan về MapReduce.*

➤ *Định nghĩa:* MapReduce là một mô hình lập trình và cũng là một phương pháp xử lý dữ liệu lớn, được giới thiệu bởi Google và sau đó được tích hợp vào hệ sinh thái Hadoop của Apache. Nó được thiết kế để xử lý song song dữ liệu lớn trên các cụm máy tính phân tán.

MapReduce bao gồm hai pha chính là pha Map và pha Reduce. Trong pha Map, dữ liệu được xử lý và phân tích bằng các hàm Map để tạo ra các cặp key-value. Key-value này sau đó được chuyển đến pha Reduce để được tổng hợp và xử lý tiếp theo. Trong pha Reduce, các cặp key-value được tổng hợp và xử lý bằng các hàm Reduce để tạo ra kết quả cuối cùng.

Trong quá trình "Map", các giá trị đầu vào được chia thành các phần nhỏ hơn và được xử lý song song trên các nút trong cụm máy tính phân tán. Mỗi nút xử lý một phần nhỏ của dữ liệu đầu vào và tạo ra các cặp key-value. Sau khi các cặp key-value được tạo ra, chúng được sắp xếp và gom nhóm lại theo khóa và truyền đến các tác vụ "Reduce".

➤ Thực thi mô hình MapReduce:



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

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

CHƯƠNG 2: PHÂN LỚP DỮ LIỆU BẰNG THUẬT TOÁN ID3

2.1 Giới thiệu thuật toán ID3.

➤ *Giới thiệu:* ID3 (Iterative Dichotomiser 3) là một thuật toán học máy thuộc nhóm cây quyết định (Decision Tree). Thuật toán này được sử dụng để phân loại dữ liệu bằng cách xây dựng một cây quyết định từ tập dữ liệu đầu vào.

Mỗi nút trong cây biểu diễn một thuộc tính, và các nhánh của nút tương ứng với các giá trị của thuộc tính đó. Quá trình xây dựng cây được thực hiện dựa trên việc tối đa hóa độ lợi thông tin (Information Gain).

➤ *Ý tưởng:*

Tạo cây quyết định bằng việc sử dụng cách tìm kiếm từ trên xuống trong tập học

Sử dụng độ lợi thông tin để chọn thuộc tính có khả năng phân loại.

2.2 Triển khai thuật toán phân lớp ID3.

Thuật toán ID3 xây dựng cây quyết định qua các bước sau:

1. **Tính Entropy của tập dữ liệu**
   Entropy là thước đo mức độ hỗn loạn (uncertainty) của dữ liệu. Công thức tính Entropy:

$$Entropy(S) = -\sum_{i=1}^{n} p_i \cdot \log_2(p_i)$$

- S: Tập dữ liệu.
- p_i: Xác suất của lớp thứ i.

2. **Tính độ lợi thông tin (Information Gain)**
   Độ lợi thông tin đo lường mức độ giảm hỗn loạn khi phân chia tập dữ liệu theo một thuộc tính. Công thức:

$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} \cdot Entropy(S_v)$$

- A: Thuộc tính cần xem xét.
- $Values(A)$: Tập các giá trị có thể có của thuộc tính A.
- S_v: Tập con dữ liệu có giá trị v của A.

3. Chọn thuộc tính tốt nhất để phân chia

- Thuộc tính có **Information Gain** lớn nhất sẽ được chọn làm nút tiếp theo trong cây.

4. Lặp lại quy trình

- Tiếp tục phân chia tập dữ liệu con đến khi đạt một trong các điều kiện dừng:
  - Mọi phần tử trong tập dữ liệu thuộc cùng một lớp.
  - Không còn thuộc tính nào để phân chia.
  - Tập dữ liệu rỗng.

2.3 Ví dụ minh họa thuật toán.

➤ Mô tả bộ dữ liệu

1. File CSV (GBvideos.csv):

- **Chức năng**: Cung cấp dữ liệu chi tiết về các video trên YouTube, bao gồm số lượt xem (views), danh mục video (categoryId), và các thông tin khác liên quan đến video.
- **Các cột chính**:
  - video_id: Mã định danh duy nhất cho mỗi video.
  - title: Tiêu đề của video.
  - views: Số lượt xem video.
  - likes: Số lượt thích.
  - dislikes: Số lượt không thích.

- categoryId: ID danh mục của video, liên kết tới file JSON để giải thích ý nghĩa danh mục.
- Các cột khác: Thông tin như bình luận (comment_count), thời gian đăng tải, và thẻ (tags).

2. File JSON (GB_category_id.json):

- **Chức năng**: Cung cấp thông tin chi tiết về danh mục video dựa trên categoryId từ file CSV.
- **Cấu trúc dữ liệu**:
  - id: Mã ID của danh mục.
  - title: Tên danh mục (e.g., "Music", "Education").
  - assignable: Xác định liệu danh mục có thể được gán cho video.

Mục tiêu xử lý

1. **Phân cụm các danh mục video (categoryId)**:
   - Dựa trên số lượng video thuộc mỗi danh mục.
   - Tổng hợp số lượng video hoặc các thông số khác như tổng lượt xem (views) cho từng danh mục.
2. **Liên kết categoryId với tên danh mục từ file JSON**:
   - Sử dụng dữ liệu JSON để gắn nhãn các danh mục bằng tên (title) thay vì sử dụng ID.

Kế hoạch thực hiện

1. **Xử lý file CSV**:
   - Đọc dữ liệu từ file CSV và tổng hợp thông tin:
     - Đếm số lượng video theo từng categoryId.

        - Tính tổng số lượt xem (views) hoặc các thông số khác như likes, comments.

2. **Ghép dữ liệu từ file JSON**:
    - Sử dụng categoryId để liên kết với title trong file JSON.
    - Tạo bảng kết quả bao gồm:
        - categoryId
        - title (Tên danh mục)
        - Số lượng video
        - Tổng số lượt xem.
3. **Phân cụm danh mục (Clustering)**:
    - Sử dụng thuật toán như K-means hoặc ID3 để nhóm các danh mục theo tiêu chí:
        - Số lượng video.
        - Tổng số lượt xem.
4. **Trực quan hóa và phân tích**:
    - Hiển thị kết quả dưới dạng biểu đồ (e.g., biểu đồ thanh, biểu đồ tròn).
    - Đưa ra nhận xét về sự phổ biến của các danh mục.

CHƯƠNG 3: ỨNG DỤNG MAPREDUCE ID3 TRONG PHÂN LỚP DỮ LIỆU

1. Ứng dụng MapReduce

Mục tiêu MapReduce

MapReduce được sử dụng để xử lý lượng dữ liệu lớn từ file CSV, thực hiện các phép tính như:

- Tính tổng lượt xem (views) cho từng categoryId.
- Đếm số lượng video thuộc mỗi categoryId.
- Kết nối dữ liệu JSON để thêm tên danh mục (title) vào kết quả.

Quy trình xử lý với MapReduce

1. **Giai đoạn Map:**
    - Đọc dữ liệu từ file CSV.
    - Trích xuất các cặp categoryId và views từ từng video.
    - Output của mapper là các cặp: (categoryId, views)
2. **Giai đoạn Shuffle and Sort:**
    - Gom nhóm tất cả các giá trị có cùng categoryId.
    - Sắp xếp dữ liệu theo categoryId.
3. **Giai đoạn Reduce:**
    - Tổng hợp số lượt xem và đếm số lượng video cho mỗi categoryId.
    - Output của reducer: (categoryId, [total_views, total_videos])
4. **Kết nối với file JSON:**
    - Sử dụng categoryId từ file CSV để tra cứu title từ file JSON.

- Tạo kết quả cuối cùng gồm: (title, categoryId, total_views, total_videos)

Ứng dụng ID3 và MapReduce cho bài toán phân cụm danh mục YouTube

---

1. Ứng dụng MapReduce

Mục tiêu MapReduce

MapReduce được sử dụng để xử lý lượng dữ liệu lớn từ file CSV, thực hiện các phép tính như:

- Tính tổng lượt xem (views) cho từng categoryId.
- Đếm số lượng video thuộc mỗi categoryId.
- Kết nối dữ liệu JSON để thêm tên danh mục (title) vào kết quả.

Quy trình xử lý với MapReduce

1. **Giai đoạn Map:**
   - Đọc dữ liệu từ file CSV.
   - Trích xuất các cặp categoryId và views từ từng video.
   - Output của mapper là các cặp: **(categoryId, views)**
2. **Giai đoạn Shuffle and Sort:**
   - Gom nhóm tất cả các giá trị có cùng categoryId.
   - Sắp xếp dữ liệu theo categoryId.
3. **Giai đoạn Reduce:**
   - Tổng hợp số lượt xem và đếm số lượng video cho mỗi categoryId.
   - Output của reducer: **(categoryId, [total_views, total_videos])**
4. **Kết nối với file JSON:**
   - Sử dụng categoryId từ file CSV để tra cứu title từ file JSON.
   - Tạo kết quả cuối cùng gồm:**(title, categoryId, total_views, total_videos)**

---

2. Ứng dụng thuật toán ID3

Mục tiêu ID3

Sử dụng ID3 để phân loại các danh mục (categoryId) dựa trên các đặc trưng:

- Tổng số lượt xem (views).
- Số lượng video (total_videos).

Quy trình áp dụng ID3

1. **Tính toán Entropy:**
    - Entropy được tính dựa trên sự phân bổ của các danh mục.

2. **Tính Information Gain (Độ lợi thông tin):**

- Xác định thuộc tính nào (tổng lượt xem hoặc số lượng video) giúp phân tách dữ liệu hiệu quả nhất.

Xây dựng cây quyết định:

- Chọn thuộc tính có Information Gain cao nhất làm nút gốc.
- Phân nhánh dữ liệu dựa trên giá trị của thuộc tính này.
- Lặp lại quá trình với các tập con dữ liệu cho đến khi đạt điều kiện dừng.

3. Kết hợp MapReduce và ID3

1. **Xử lý dữ liệu với MapReduce:**
    - Sử dụng MapReduce để tính tổng lượt xem và đếm số lượng video theo danh mục.
    - Kết quả đầu ra của MapReduce là tập dữ liệu rút gọn, dễ xử lý.
2. **Phân loại bằng ID3:**
    - Dùng dữ liệu đầu ra từ MapReduce làm đầu vào cho thuật toán ID3.
    - Phân loại các danh mục thành các nhóm, ví dụ:
        - Nhóm phổ biến: Danh mục có tổng lượt xem lớn và số lượng video

nhiều.

- Nhóm kém phổ biến: Danh mục có ít lượt xem và số lượng video ít.

Hướng dẫn chạy demo

1. CategoryMapper.java
Mapper là phần đầu tiên của quá trình MapReduce. File này đọc từng dòng trong file đầu vào và ánh xạ dữ liệu dưới dạng các cặp key-value.
Nhiệm vụ của Mapper

- Đọc file đầu vào (CSV).
- Tách cột category_id và views từ mỗi dòng.
- Gửi các cặp <category_id, views> đến Reducer.

2. CategoryReducer.java
Reducer tổng hợp các giá trị views cho từng category_id và ánh xạ chúng sang category_name bằng cách sử dụng file JSON.
Nhiệm vụ của Reducer

- Nhận các cặp <category_id, [views1, views2, ...]> từ Mapper.
- Tính tổng views cho từng category_id.
- Ánh xạ category_id sang category_name (hoặc "Unknown" nếu không tìm thấy).

Dưới đây là giải thích chi tiết cho ba file mã Java (CategoryMapper.java, CategoryReducer.java, CategoryDriver.java) dùng để thực hiện MapReduce, với nội dung được giải thích bằng tiếng Việt.

---

1. CategoryMapper.java
Mapper là phần đầu tiên của quá trình MapReduce. File này đọc từng dòng trong file đầu vào và ánh xạ dữ liệu dưới dạng các cặp key-value.
Nhiệm vụ của Mapper

- Đọc file đầu vào (CSV).
- Tách cột category_id và views từ mỗi dòng.
- Gửi các cặp <category_id, views> đến Reducer.

Hoạt động

- Mỗi dòng dữ liệu được tách thành mảng fields dựa trên dấu phẩy.
- category_id (cột 6) và views (cột 9) được trích xuất.
- Nếu dòng dữ liệu hợp lệ, Mapper gửi cặp <category_id, views> đến Reducer.

---

2. CategoryReducer.java

Reducer tổng hợp các giá trị views cho từng category_id và ánh xạ chúng sang category_name bằng cách sử dụng file JSON.

Nhiệm vụ của Reducer

- Nhận các cặp <category_id, [views1, views2, ...]> từ Mapper.
- Tính tổng views cho từng category_id.
- Ánh xạ category_id sang category_name (hoặc "Unknown" nếu không tìm thấy).

Hoạt động

- Trước khi bắt đầu xử lý, Reducer tải file JSON để ánh xạ category_id sang category_name.
- Trong phương thức reduce:
    - Tính tổng views cho mỗi category_id.
    - Nếu category_id không có trong file JSON, gán tên là "Unknown".
- Ghi cặp <category_name, total_views> vào đầu ra.

---

3. CategoryDriver.java

Driver là nơi cấu hình và khởi chạy MapReduce Job.

Nhiệm vụ của Driver

- Định nghĩa đầu vào và đầu ra cho Job.
- Thiết lập Mapper và Reducer.
- Chạy Job.

Tóm tắt hoạt động

1. **Mapper**: Tách và trích xuất dữ liệu từ file CSV.
2. **Reducer**: Tính tổng views và ánh xạ category_id sang category_name.
3. **Driver**: Cấu hình và khởi chạy MapReduce Job.

Thực hiện trên Hadoop Cloudera:

Upload file Gbvideos.csv vào HDFS
hdfs dfs -put /local/path/to/GBvideos.csv /user/cloudera/input/
Tạo đường dẫn input:
hdfs dfs -ls /user/cloudera/input/

Biên dịch tất cả các class:
javac -classpath $(hadoop classpath) -d . CategoryMapper.java CategoryReducer.java CategoryDriver.java
Đóng gói file JAR

jar -cvf CategoryJob.jar -C . .

Chạy Hadoop lấy Output

```
[root@quickstart map_reduce_id3]# hadoop jar CategoryJob.jar CategoryDriver /user/cloudera/input/GBvideos.csv /user/cloudera/output /home/cloudera/Desktop/map_reduce_id3/GB_category_id.json
24/12/08 16:02:55 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
24/12/08 16:02:57 WARN mapreduce.JobSubmitter: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
24/12/08 16:03:00 INFO input.FileInputFormat: Total input paths to process : 1
24/12/08 16:03:00 INFO mapreduce.JobSubmitter: number of splits:1
24/12/08 16:03:01 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1733695146692_0005
24/12/08 16:03:02 INFO impl.YarnClientImpl: Submitted application application_1733695146692_0005
24/12/08 16:03:02 INFO mapreduce.Job: The url to track the job: http://quickstart.cloudera:8088/proxy/application_1733695146692_0005/
24/12/08 16:03:02 INFO mapreduce.Job: Running job: job_1733695146692_0005
24/12/08 16:03:21 INFO mapreduce.Job: Job job_1733695146692_0005 running in uber mode : false
24/12/08 16:03:21 INFO mapreduce.Job:  map 0% reduce 0%
24/12/08 16:03:38 INFO mapreduce.Job:  map 100% reduce 0%
24/12/08 16:03:51 INFO mapreduce.Job:  map 100% reduce 100%
24/12/08 16:03:52 INFO mapreduce.Job: Job job_1733695146692_0005 completed successfully
24/12/08 16:03:53 INFO mapreduce.Job: Counters: 49
        File System Counters
                FILE: Number of bytes read=1102380
                FILE: Number of bytes written=2426281
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=53213570
                HDFS: Number of bytes written=92433
                HDFS: Number of read operations=6
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
        Job Counters
                Launched map tasks=1
                Launched reduce tasks=1
                Data-local map tasks=1
                Total time spent by all maps in occupied slots (ms)=14682
                Total time spent by all reduces in occupied slots (ms)=11102
                Total time spent by all map tasks (ms)=14682
                Total time spent by all reduce tasks (ms)=11102
                Total vcore-seconds taken by all map tasks=14682
                Total vcore-seconds taken by all reduce tasks=11102
```

Xuất ra output

hdfs dfs -get /user/cloudera/output/part-r-00000 /home/cloudera/Desktop/final_output.csv

Tuy nhiên output ban đầu sẽ không được sắp xếp kết quả

```
nknown 17258
nknown 110823
nknown 18811
nknown 96391
nknown 2231
nknown 65624
eople & Blogs  227474014
omedy  419999608
ntertainment   1520043352
ews & Politics 81445759
owto & Style   12734343
ducation       26772380
cience & Technology    352306726
```

Do đó cần sắp xếp lại kết quả như hình

hdfs dfs -get /user/cloudera/output/part-r-00000 /home/cloudera/Desktop/final_output_unsorted.csv
grep -i "Unknown" /home/cloudera/Desktop/final_output_unsorted.csv > /home/cloudera/Desktop/unknown.csv
grep -v -i "Unknown" /home/cloudera/Desktop/final_output_unsorted.csv > /home/cloudera/Desktop/not_unknown.csv
cat /home/cloudera/Desktop/unknown.csv /home/cloudera/Desktop/not_unknown.csv > /home/cloudera/Desktop/final_output_sorted.csv
cat /home/cloudera/Desktop/final_output_sorted.csv

```
People & Blogs  227474014
Comedy  419999608
Entertainment   1520043352
News & Politics 81445759
Howto & Style   12734343
Education       26772380
Science & Technology    352306726
Unknown 1020704
```

Dựa vào kết quả ta thấy các nhóm đã được phân cụm theo Cateogory name

Lợi ích khi kết hợp MapReduce và ID3

- Xử lý dữ liệu lớn: MapReduce giúp xử lý hiệu quả dữ liệu khổng lồ, giảm kích thước tập dữ liệu.
- Phân loại chính xác: ID3 giúp phân loại danh mục dựa trên đặc trưng cụ thể, hỗ trợ ra quyết định.
- Tối ưu hóa tài nguyên: Kết hợp MapReduce và ID3 tận dụng lợi thế của xử lý song song và thuật toán máy học.

Ứng dụng thực tiễn

- Phân tích xu hướng nội dung: Xác định các danh mục video phổ biến dựa trên lượt xem.
- Hỗ trợ chiến lược marketing: Tập trung vào các danh mục có tiềm năng cao.
- Cá nhân hóa nội dung: Gợi ý danh mục phù hợp cho người dùng hoặc nhà sáng tạo nội dung.

CHƯƠNG 4: KẾT LUẬN

- Trong bài tập lớn này, chúng tôi đã triển khai và phân tích dữ liệu lớn của YouTube thông qua việc sử dụng mô hình xử lý phân tán MapReduce kết hợp với thuật toán ID3. Quá trình thực hiện bao gồm việc xử lý khối lượng dữ liệu khổng lồ, trích xuất thông tin quan trọng và xây dựng mô hình cây quyết định để hỗ trợ phân loại cũng như ra quyết định hiệu quả.

Tài Liệu Tham khảo

https://github.com/SarahAyaz/YouTube_Data_Analysis