Báo cáo môn phân tích dữ liệu lớn

Xây dựng ứng dụng học máy phân cụm K-Means dùng thư viện MLlib trong Spark để xử lý dữ liệu trên hệ thống dữ liệu phân tán HDFS

Nhóm thực hiện : Nhóm 4

Thai Nguyen University of Information Technology and Communication

Ngày 8 tháng 5 năm 2024

Nội dung

Tổng quan về dữ liệu lớn và công cụ Apache Spark

Khái niệm về dữ liệu lớn

- Dữ liệu lớn (Big Data) đề cập đến các tập dữ liệu cực lớn và phức tạp không thể được xử lý hoặc phân tích hiệu quả bằng các phương pháp xử lý dữ liệu truyền thống.
- Chúng có các đặc trưng bởi khối lượng, tốc độ và sự đa dạng của dữ liệu bao gồm cả dữ liệu có cấu trúc và bán cấu trúc, không cấu trúc.

Đặc điểm của dữ liệu lớn

Dữ liệu lớn có 5 đặc điểm, hay còn được gọi là 5V.

Hình: 1. Đặc điểm của dữ liệu lớn

Đặc điểm của dữ liệu lớn

- Velocity (vận tốc): Tốc độ dữ liệu được tạo và thu thập.
- Variety (đa dạng): Có nhiều loại và nguồn dữ liệu khác nhau.
- Veracity (xác thực): Tính chính xác và độ tin cậy của dữ liệu.
- Volume (khối lượng): Quy mô dữ liệu được tạo và thu thập.
- Values (Giá trị): Thông tin giá trị mà dữ liệu mang lại.

Phân loại dữ liệu lớn

Có ba loại dữ liệu lớn chính, được đặc trưng bởi loại dữ liệu và nguồn dữ liệu đó được tạo ra:

- Dữ liệu có cấu trúc: Dữ liệu có tính tổ chức cao và có thể dễ dàng lưu trữ phân tích trong cơ sở dữ liệu.
- Dữ liệu phi cấu trúc: Dữ liệu không có cấu trúc hoặc định dạng được xác định trước.
- Dữ liệu bán cấu trúc: Sự kết hợp của dữ liệu có cấu trúc và phi cấu trúc.

Ngoài ra dữ liệu lớn cũng có thể được phân loại theo nguồn mà chúng được tạo ra: Dữ liệu do máy tạo ra và dữ liệu do con người tạo ra.

Ứng dụng của dữ liệu lớn

Dữ liệu lớn có nhiều ứng dụng trong nhiều lĩnh vực khác nhau bao gồm chăm sóc sức khỏe, tài chính, tiếp thị và khoa học:

- Sử dụng phân tích dữ liệu bệnh nhân cải thiện kết quả chăm sóc người bệnh.
- Phát hiện gian lận trong giao dịch tài chính.
- Phân tích dữ liệu trong khoa học để thực hiện các khám phá mới.

Các công cụ của dữ liệu lớn

Một số công cụ xử lý dữ liệu lớn bao gồm:

- Apache Hadoop: Là khung phần mềm mã nguồn mở được sử dụng rộng rãi để lưu trữ và xử lý phân tán các bộ dữ liệu lớn. Có khả năng mở rộng và chịu lỗi. Một số công cụ xử lý và phân tích dữ liệu như HDFS, Mapreduce.
- Apache Cassandra: Là hệ thống quản lý dữ liệu trên nhiều máy chủ. Có khả năng mở rộng và chịu lỗi cao có tính sẵn sằng và thông lượng ghi cao.
- Cở sở dữ liệu NoSQL: Là một loại cơ sở dữ liệu được thiết kế để xử lý dữ liệu phi và bán cấu trúc. Một số cơ sở dữ liệu phổ biến như MongoDB, Apache CouchDB.
- Thư viện máy học: Thư viện máy học được sử dụng để phát triển và triển khai các mô hình máy học: phân tích dự đoán, xử lý ngôn ngữ tự nhiên, thị giác máy tính.

Ưu điểm của dữ liệu lớn

Dữ liệu lớn có một số ưu điểm như:

- Cải thiện việc ra quyết định: Cung cấp cho các tổ chức quyền truy cập vào lượng dữ liệu khổng lồ, cho phép họ đưa ra quyết định sáng suốt trên dữ liệu.
- Tăng hiệu quả và năng suất: Cho phép các tổ chức xử lý và phân tích dữ liệu nhanh chóng và chính xác hơn. Giúp tối ưu hóa hoạt động, giảm lãng phí, tăng năng suất.
- Hiểu biết sâu sắc hơn về khách hàng: Giúp các tổ chức hiểu biết rõ hơn về hành vi, sở thích và nhu cầu của khách hàng. Cải thiện chiến lược tiếp thị và thu hút khách hàng.

Ưu điểm của dữ liệu lớn

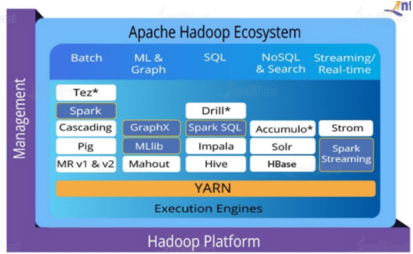
- Tăng cường đổi mới sản phẩm và dịch vụ: Cung cấp cho các tổ chức những hiểu biết sâu sắc về các xu hướng mới nổi, sở thích của người tiêu dùng và cơ hội thị trường, thúc đẩy và đổi mới sản phẩm và dịch vụ.
- Tiết kiệm chi phí: Bằng cách cải thiện hiệu quả và năng suất. Dữ liệu lớn có thể giúp các tổ chức giảm chi phí và tăng lợi nhuận. Ví dự như sử dụng để tối ưu hóa hoạt động của chuỗi cung ứng, giảm chi phí tồn kho và cải thiện việc phân bổ nguồn lực.

Nhược điểm của dữ liệu lớn

+ Bên cạnh các ưu điểm thì dữ liệu lớn cũng mang lại nhiều nhược điểm và thách thức đáng kể bao gồm nhu cầu về chuyên môn, công cụ và cơ sở hạ tầng chuyên dụng để quản lý và phân tích các bộ dữ liệu lớn.
+ Các tổ chức mong muốn làm việc với dữ liệu lớn phải đầu tư vào cơ sở hạ tầng chuyên môn cần thiết để phân tích và rút ra những hiểu biết sâu sắc từ chúng một cách hiệu quả.

Khái niệm về Apache Spark

Apache Spark là một khung tính toán cụm có tốc độ cao được thiết kế để xử lý thời gian thực. Spark khắc phục những hạn chế của Hadoop Mapreduce và mở rộng Mapreduce để xử lý dữ liệu hiệu quả.



Hình: 2. Spark tích hợp trong Apache Hadoop

Khái niệm về Apache Spark

Spark dẫn đầu thị trường về xử lý dữ liệu lớn. Chúng được sử dụng rộng rãi trong các tổ chức theo nhiều cách và vượt qua Hadoop khi chạy nhanh hơn gấp 100 lần trong bộ nhớ và 10 lần trên đĩa.

Ứng dụng của Spark

Spark đã và đang được ứng dụng và được sử dụng thành công trong nhiều lĩnh vực, nghành nghề khác nhau.

Hình: 3. Ứng dụng của Spark

- Ngân hàng: Phát hiện gian lận tài chính, đánh giá rủi ro tín dụng, phân khúc khách hàng và quảng cáo.

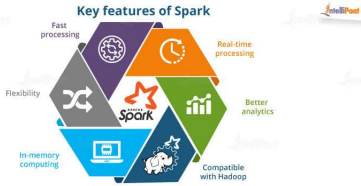
Ứng dụng của Spark

- Thương mại điện tử: Sử dụng để phân cụm dữ liệu theo thời gian thực, đưa ra khuyến nghị tốt hơn cho khách hàng để hiện thị các xu hướng mới.
- Chăm sóc sức khỏe: Sử dụng đê theo dõi hồ sơ sức khỏe của bệnh nhân, thu thập thông tin chi tiết như phản hồi của bệnh nhân và dịch vụ của bệnh viện cũng như theo dõi dữ liệu y tế.
- Phương tiện truyền thông: Sử dụng để tiếp thị có mục tiêu như các trang tin tức dựa trên độc giả, hay đề xuất các bộ phim thông qua sở thích của người xem.
- Du lịch: Nhiều công ty như TripAdvisor là một trong những công ty sử dụng spark để so sánh các gói du lịch khác nhau từ các nhà cung cấp khác nhau.

Tính năng của Apache Spark

Mốt số tính năng nổi bật của Spark có thể kể đến như:

- Xử lý nhanh: Spark có tốc độ xử lý dữ liệu nhanh nhờ chứa bộ dữ liệu RDD, giúp tiết kiệm thời gian thực hiện các thao tác đọc ghi.
- Tính linh hoạt: Hỗ trợ nhiều ngôn ngữ và cho phép các nhà phát triển viết ứng dụng bằng Java, Scala, Python, R

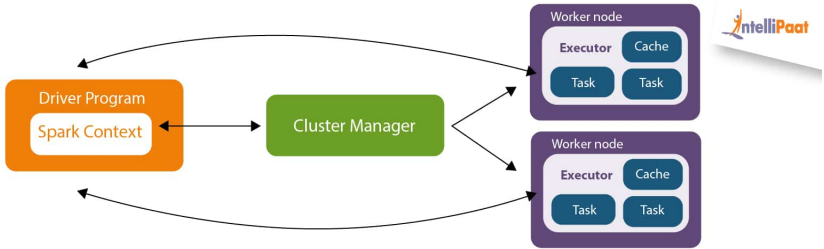


Hình: 4. Tính năng của Spark

Tính năng của Apache Spark

- Điện toán trong bộ nhớ: Spark lưu trữ dữ liệu trong RAM của máy chủ, cho phép truy cập dữ liệu nhanh chóng và từ đó tăng tốc độ phân tích.
- Xử lý thời gian thực: Có thể xử lý dữ liệu theo thời gian thực, không giống như Mapreduce xử lý dữ liệu được lưu trữ, Spark có thể xử lý dữ liệu theo thời gian thực và do đó có thể tạo ra kết quả tức thì.
- Phân tích tốt hơn: Ngược lại với Mapreduce bao gồm các chức năng Map và Reduce, Spark có nhiều tính năng hơn. Apache Spark gồm một tập hợp phong phú các truy vấn SQL, thuật toán Machine Learning, phân tích phức tạp,v.v. Nhờ đó mà phân tích dữ liệu lớn có thể được thực hiện theo cách tốt hơn.
- Khả năng tương thích với Hadoop: Spark không chỉ có khả năng hoạt động độc lập mà còn có thể hoạt động trên Hadoop. Không chỉ vậy, Spark còn chắc chắn tương thích với cả hai phiên bản của hệ sinh thái Hadoop.

Kiến trúc của Apache Spark



Hình: 5. Kiến trúc căn bản của Spark

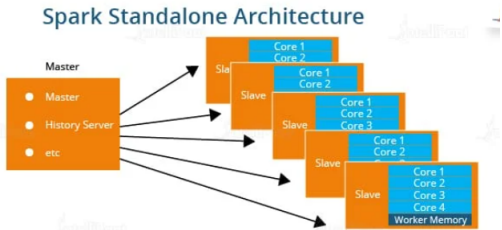
Kiến trúc của Apache Spark

- Diver Program trong kiến trúc Apache Spark gọi chương trình chính của ứng dụng và tạo SparkContext. SparkContext bao gồm tất cả các chức năng cơ bản. SparkDriver chứa nhiều thành phần khác như DAG Scheduler, Task Scheduler, Backend Scheduler, và Block Maneger, chịu trách nhiệm dịch mã do người dùng viết thành các công việc thực sự được thực thi trên cụm.
- SparkDriver và SparkContext cùng nhau giám sát việc thực hiện công việc trong cụm. SparkDriver hoạt động với Cluster Manager để quản lý nhiều công việc khác. Cluster Manager thực hiện công việc phân bổ tài nguyên, sau đó công việc được chia thành nhiều nhiệm vụ nhỏ hơn và phân bổ tiếp cho các worker node.

Kiến trúc của Apache Spark

SparkContext có thể hoạt động với nhiều Cluster Manager khác nhau:

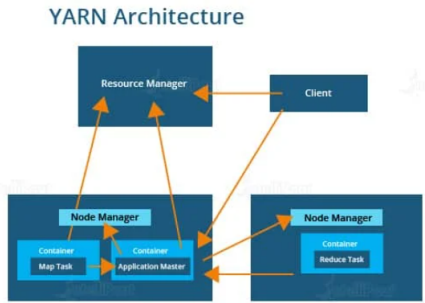
- Standardlone: Cụm độc lập bao gồm một master độc lập có chức năng như trình quản lý tài nguyên. Worker độc lập đóng vai trò là worker node.



Hình: 6. Cụm độc lập

Kiến trúc của Apache Spark

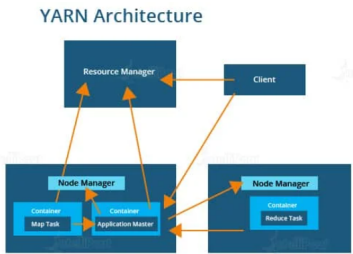
- Hadoop YARN: Đảm nhiệm việc quản lý tài nguyên cho Hadoop chúng có hai thành phần là Resource Manager và Node Manager.



Hình: 7. Hadoop YARN

Kiến trúc của Apache Spark

- Apache Mesos: xử lý khối lượng công việc từ nhiều nguồn bằng cách sử dụng tính năng chia sẻ và cách lý tài nguyên động. Giúp triển khai và quản lý các ứng dụng trong môi trường cụm quy mô lớn.



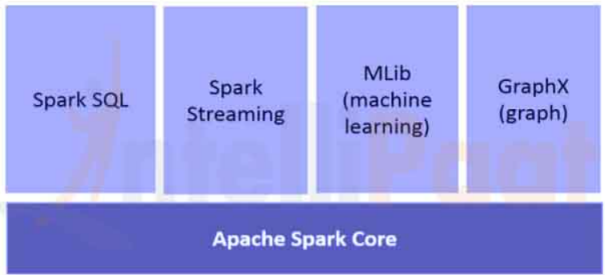
Hình: 8. Apache Mesos

Chế độ thực thi ứng dụng

Có ba chế độ thực thi được lựa chọn là:

- Chế độ cụm: Là chế độ phổ biến nhất để chạy các ứng dụng Spark trong đó tập lệnh Python, Java hoặc R được biên dịch trước và được người dùng gửi tới Cluster Manager. Sau đó, quy trình SparkDriver được Cluster Manager khởi chạy trên worker node bên trong cụm, bên cạnh các quy trình thực thi.
- Chế độ máy khách: Gần giống như chế độ cụm, ngoại trừ SparkDriver vẫn còn trên máy khách đã được gửi ứng dụng. Điều này có nghĩa là máy khách duy trì quy trình SparkDriver và Cluster Manager duy trì quy trình thực thi. Những máy này thường được gọi là máy cổng hoặc nút biên.
- Chế độ cục bộ: Ở chế độ này, toàn bộ ứng dụng Spark được chạy trên một máy. Chúng quan sát sự song song thông qua các luồng trên máy đơn lẻ đó. Đây là một cách phổ biến để thử nghiệm các ứng dụng hoặc thử nghiệm sự phát triển của máy đơn. Không được khuyến khích để chạy các ứng dụng trong sản xuất.

Thành phần của Apache Spark



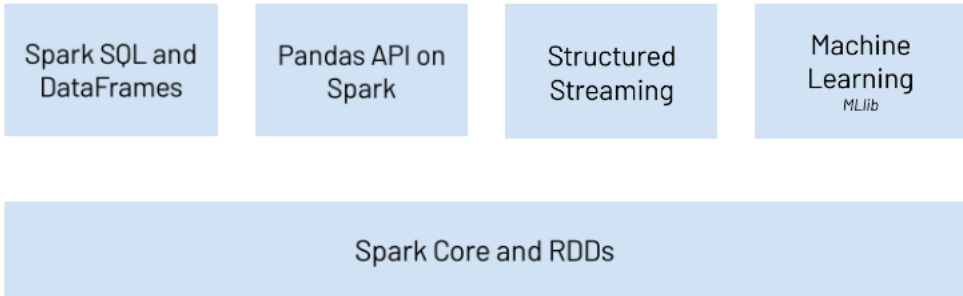
Hình: 9. Thành phần của Spark

Pyspark và thuật toán K-Means Clustering

Khải niệm về Pyspark

- Pyspark là API Python cho Apache Spark. Cho phép thực hiện xử lý dữ liệu quy mô lớn, th gian thực trong môi trường phân tán bằng Python. Chúng cung cấp các công cụ để phân tích tương tác dữ liệu.
- Pyspark kết hợp khả năng học hỏi và tính dễ sử dụng của Python với sức mạnh của Apache Spark để cho phép xử lý và phân tích dữ liệu ở mọi quy mô cho người quen thuộc với Python.

Thành phần của Pyspark



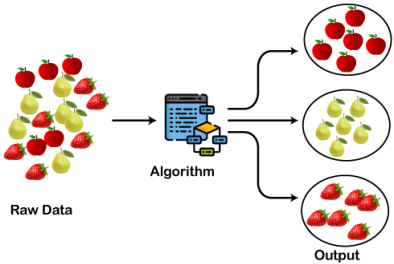
Hình: 10. Thành phần của Pyspark

Tính năng của Pyspark

- Tính toán thời gian thực: Khung Pyspark có tính năng xử lý trong bộ nhớ giúp giảm độ trễ.
- Tính linh hoạt: Hỗ trợ nhiều ngôn ngữ khác nhau bao gồm Scala, Python, Java, Python và R, khiến chúng trở thành một trong những khung ưa thích để xử lý các tập dữ liệu khổng lồ.
- Bộ đệm và tính bền bỉ của ổ đĩa: Khung này cung cấp bộ nhớ đệm mạnh mẽ với khả năng duy trì ổ đĩa vượt trội.
- Tốc độ xử lý: Khung Pyspark cung cấp tốc độ xử lý dữ liệu lớn nhanh hơn nhiều so với các ứng dụng khác.
- Hoạt động hiệu quả với RDD: Làm việc hiệu quả và tuyệt vời với RDD.

Khái niệm về phân cụm

Phân cụm hay phân tích cụm là một kỹ thuật học máy, nhằm nhóm các tập dữ liệu khổng lồ không được gán nhãn. Một cách cụ thể hơn phân cụm là "Một cách nhóm các điểm dữ liệu thành các cụm khác nhau, bao gồm các điểm dữ liệu tương tự nhau. Các đối tượng có những điểm tương đồng có thể vẫn nằm trong một nhóm có ít hoặc không có điểm tương đồng với nhóm khác".



Ứng dụng của kỹ thuật phân cụm

Kỹ thuật phân cụm có thể được sử dụng rộng rãi trong nhiều nhiệm vụ khác nhau. Một số lĩnh vực sử dụng phổ biến nhất kỹ thuật này là:

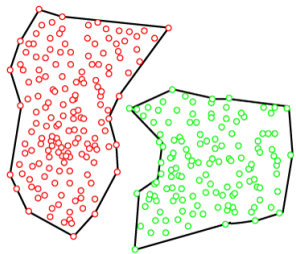
- Phân khúc thị trường
- Phân tích dữ liệu thống kê
- Phân tích mạng xã hội
- Phân đoạn hình ảnh
- Phát hiện bất thường

Các loại phương pháp phân cụm

Các phương pháp phân cụm được chia thành Hard Clustering (phân cụm cứng) điểm dữ liệu chỉ thuộc một nhóm và Soft Clustering (phân cụm mềm) điểm dữ liệu cũng có thể thuộc về một nhóm khác.

- Phân cụm phân vùng: Đây là một kiểu phân cụm chia dữ liệu thành các nhóm không phân cấp. Đây còn được gọi là phương pháp dựa trên centroid (tâm).
- Phân cụm dựa trên mật độ: Phương pháp phân cụm dựa trên mật độ là kết nối các khu vực có mật độ cao thành các cụm và các phân bố có hình dạng tùy ý được hình thành miễn là khu vực dày đặc và có thể được kết nối.

Các loại phương pháp phân cụm

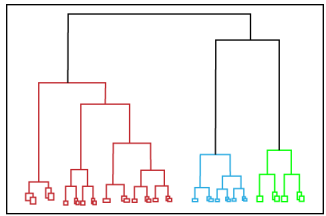


Hình: 12. Phân cụm dựa trên mật độ

- Phân cụm dựa trên mô hình phân phối: Trong phương pháp phân cụm dựa trên mô hình phân phối, dữ liệu được phân chia dựa trên xác suất về cách tập dữ liệu thuộc về một phân phối cụ thể.

Các loại phương pháp phân cụm

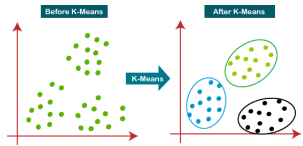
- Phân cụm theo cấp bậc: Phân cụm theo cấp bậc có thể được sử dụng thay thế cho phân cụm được phân vùng vì không có yêu cầu chỉ định trước số lượng của cụm sẽ được tạo. Trong kỹ thuật này, tập dữ liệu được chia thành các cụm để tạo ra cấu trúc dạng cây, còn được gọi là Dendrogram.



Hình: 13. Phân cụm theo cấp bậc

Khái niệm về thuật toán K-Means

Phân cụm K-means là một thuật toán học không giám sát, nhóm tập dữ liệu không được gán nhãn thành các cụm khác nhau. Ở đây k xác định số lượng cụm được xác định trước. Một cách trừu tượng hơn đây là một thuật toán lặp chia tập dữ liệu không được gán nhãn thành k cụm khác nhau sao cho mỗi tập dữ liệu chỉ thuộc một nhóm có các thuộc tính tương tự. Đây là một thuật toán dựa trên centroid (tâm), trong đó mỗi cụm được liên kết với một centroid.



Hình: 14. Thuật toán K-means Clustering

Cách thức hoạt động của K-Measn

Hoạt động của thuật toán K-means có thể được giải thích theo các bước sau:

1. Chọn số k để quyết định số cụm
2. Chọn k điểm hoặc tâm ngẫu nhiên
3. Gán từng điểm dữ liệu cho trọng tâm gần nhất của chúng, tâm này sẽ tạo thành các cụm k được xác định trước: ($c_i = \arg\min_j \|\mathbf{x}_i - \mu_j\|_2^2$).
4. Tính toán khoảng cách và đặt trong tâm mới của cụm: ($\mu_j := \frac{\sum_{i=1}^{n} \mathbf{1}(c_i=j)\mathbf{x}_i}{\sum_{i=1}^{n} \mathbf{1}(c_i=j)}$).
5. Lặp lại bước thứ 3
6. Nếu số lượng điểm dữ liệu trong một cụm có sự thay đổi thì lặp lại bước 4, nếu không thuật toán sẽ dừng lại.

Ưu điểm của thuật toán K-Means

Thuật toán K-Means có một số ưu điểm như:

- Đơn giản và dễ dàng cài đặt và sử dụng
- Độ phức tạp tính toán tương đối nhỏ, phù hợp cho tập dữ liệu nhỏ.

Nhược điểm của thuật toán K-Means

Bênh cạnh những ưu điểm thì K-Means cũng có một số hạn chế:

- Cần phải xác định trước số cụm cho thuật toán
- Vị trí tâm của cụm sẽ bị phụ thuộc vào điểm khởi tạo ban đầu của chúng
- Đối với những bộ dữ liệu có hình dạng phức tạp hoặc mất cân bằng thì thuật toán không hội tụ về qui luật phân chia tổng quát.
- Thuật toán rất nhạy cảm với outliers: Khi xuất hiện outliers thì thường khiến cho tâm cụm bị chệch và do đó dự báo cụm không còn chuẩn xác.
- Thuật toán k-Means yêu cầu phải tính khoảng cách từ một điểm tới toàn bộ các tâm cụm để tìm ra tâm cụm gần nhất. Như vậy chúng ta cần phải load toàn bộ dữ liệu lên RAM, đối với những bộ dữ liệu kích thước lớn thì sẽ vượt quá khả năng lưu trữ của RAM -> tốn bộ nhớ.

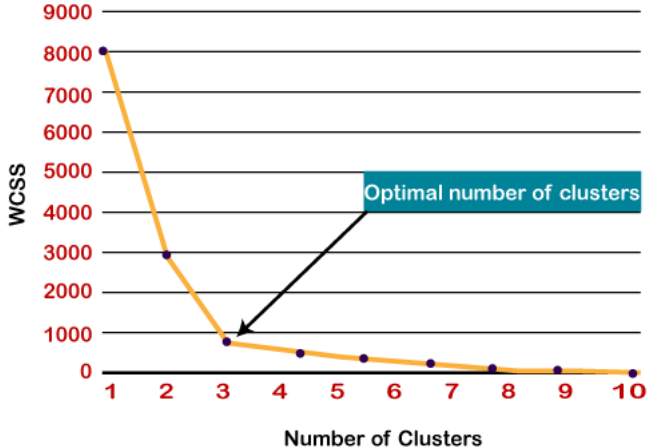
Kỹ thuật Elbow

Elbow là một cách giúp chúng ta lựa chọn được số lượng các cụm phù hợp dựa vào đồ thị trực quan hóa bằng cách nhìn vào sự suy giảm của hàm biến dạng và lựa chọn ra điểm khuỷu tay (elbow point).
Phương pháp này sử dụng khái niệm giá trị WCSS (Within Cluster Sum of Squares) để xác định tổng số biến thể trong cụm. Công thức tính giá trị WCSS cho *n* cụm như sau:

$$WCSS = \sum_{p_i \in \mu_1} distance(P_i C_1)^2 + \sum_{p_i \in \mu_2} distance(P_i C_2)^2 + \sum_{p_i \in \mu_n} distance(P_i C_n)^2$$

Trong đó $\sum_{p_i \in \mu_1} distance(P_i C_1)^2$ là tổng bình phương khoảng cách giữa mỗi điểm dữ liệu và trọng tâm của chúng trong cụm 1 và tương tự cho các số hạng còn lại.

Kỹ thuật Elbow



Kỹ thuật Elbow

Để tìm giá trị tối ưu của cụm, phương pháp khuỷu tay thực hiện theo các bước sau:

- Thực thi phân cụm K-Means trên một tập dữ liệu nhất định cho các giá trị k khác nhau ví dụ từ 1 đến 10.
- Với mỗi giá trị của k, tính WCSS.
- Vẽ đường cong giữa các giá trị WCSS được tính toán và số cụm k.
- Điểm uốn cong hoặc một điểm của đồ thị trông giống như một cánh tay thì điểm đó được coi là giá trị tốt nhất của k.

Silhouette score

Với mỗi node *i* đặt:

- a_i là khoảng cách trung bình từ *i* tới tất cả các node trong cùng cụm với *i*.
- b_i là khoảng cách trung bình ngắn nhất từ *i* tới bất kỳ cụm nào không chứa *i*. Cụm tương ứng với b_i này được gọi là cụm hàng xóm của *i*.

Khi đó:

$$s(i) = \frac{b_i - a_i}{max[a_i, b_i]}$$

s_i nằm trong đoạn [-1, 1], s_i càng gần 1 thì node *i* càng phù hợp với cụm mà nó được phân vào. $s_i = 0$ thì không thể xác định được *i* nên thuộc về cụm nào giữa cụm hiện tại và cụm hàng xóm của chúng. s_i càng gần -1 thì chứng tỏ *i* bị phân sai cụm, chúng nên thuộc về cụm hàng xóm chứ không phải là cụm hiện tại.

Cài đặt và triển khai thuật toán trên Spark

Ngôn ngữ và công cụ

- Ngôn ngữ: Sử dụng ngôn ngữ Python trong thư viện Pyspark.
- Công cụ: Sử dụng Neovim làm trình biên dịch

Tập dữ liệu

Tập dữ liệu về khách hàng trong mua bán sản phẩm điện tử.

InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
536365	85123A	WHITE HANGING HE	6	12/1/2010 8:26	2.55	17850	United Kingdom
536365	71053	WHITE METAL LAN	6	12/1/2010 8:26	3.39	17850	United Kingdom
536365	84406B	CREAM CUPID HEA	8	12/1/2010 8:26	2.75	17850	United Kingdom
536365	84029G	KNITTED UNION FL	6	12/1/2010 8:26	3.39	17850	United Kingdom
536365	84029E	RED WOOLLY HOTT	6	12/1/2010 8:26	3.39	17850	United Kingdom
536365	22752	SET 7 BABUSHKA N	2	12/1/2010 8:26	7.65	17850	United Kingdom
536365	21730	GLASS STAR FROS	6	12/1/2010 8:26	4.25	17850	United Kingdom
536366	22633	HAND WARMER UNI	6	12/1/2010 8:28	1.85	17850	United Kingdom
536366	22632	HAND WARMER RED	6	12/1/2010 8:28	1.85	17850	United Kingdom
536367	84879	ASSORTED COLOU	32	12/1/2010 8:34	1.69	13047	United Kingdom

Hình: 16. Tập dữ liệu Customer

Tiền xử lý dữ liệu

- Xử lý dữ liệu bị thiếu.

```
---------------Kiểm tra các giá trị bị thiếu-----------------
+---------+---------+-----------+--------+-----------+---------+----------+-------+
|InvoiceNo|StockCode|Description|Quantity|InvoiceDate|UnitPrice|CustomerID|Country|
+---------+---------+-----------+--------+-----------+---------+----------+-------+
|        0|        0|       1454|       0|          0|        0|    135080|      0|
+---------+---------+-----------+--------+-----------+---------+----------+-------+
```

Hình: 17. Xử lý giá trị bị thiếu

Tiền xử lý dữ liệu

Chuẩn hóa dữ liệu bằng kĩ thuật RFM. Đây là kĩ thuật dùng để tính toán 3 chỉ số Recency (Lần gần nhất mua hàng), Frequency (Tần suất mua hàng) và Monetary (Tổng số tiền đã mua hàng).

```
-----------Hiện thị  tập dữ liệu sau khi tính toán chỉ  số RFM------
+-------+---------+---------+
|Recency|Frequency|Monetary|
+-------+---------+---------+
|    326|       10|     0.0|
|      2|      910| 21550.0|
|     75|      155|  8986.2|
|     19|      365| 8787.75|
|    310|       85|  1672.0|
+-------+---------+---------+
only showing top 5 rows
```

Hình: 18. Dữ liệu sau khi được tính toán chỉ số RFM

Tiền xử lý dữ liệu

VectorAssembler là một lớp trong thư viện PySpark, được sử dụng để tổng hợp các cột dữ liệu thành một cột vector duy nhất.

```
-------------Dữ liệu sau khi được chuyển thành vector-----------------
+-------+---------+--------+--------------------+
|Recency|Frequency|Monetary|          num_vector|
+-------+---------+--------+--------------------+
|    326|       10|     0.0|    [326.0,10.0,0.0]|
|      2|      910| 21550.0| [2.0,910.0,21550.0]|
|     75|      155|  8986.2| [75.0,155.0,8986.2]|
|     19|      365| 8787.75|[19.0,365.0,8787.75]|
|    310|       85|  1672.0| [310.0,85.0,1672.0]|
+-------+---------+--------+--------------------+
```

Hình: 19. Kỹ thuật VectorAssembler

Tiền xử lý dữ liệu

Co giãn đặc trưng bằng standardscaler:

```
-------------Dữ liệu sau khi được co giãn đặc trưng-------------------
+-------+---------+--------+--------------------+--------------------+
|Recency|Frequency|Monetary|          num_vector|       scaler_number|
+-------+---------+--------+--------------------+--------------------+
|    326|       10|     0.0|    [326.0,10.0,0.0]|[2.32175727464055...|
|      2|      910| 21550.0| [2.0,910.0,21550.0]|[-0.8936310147844...|
|     75|      155|  8986.2| [75.0,155.0,8986.2]|[-0.1691762458707...|
|     19|      365| 8787.75|[19.0,365.0,8787.75]|[-0.7249223699689...|
|    310|       85|  1672.0| [310.0,85.0,1672.0]|[2.16297266775537...|
+-------+---------+--------+--------------------+--------------------+
```

Hình: 20. Chuẩn hóa dữ liệu

Huấn luyện mô hình

Áp dụng kĩ thuật silhouette để tính điểm cho các cụm được khởi tạo ngẫu nhiên từ 2 đến 10. Thực hiện huấn luyện trên tập dữ liệu và thu được kết quả:

```
Sil score for k =  2 is 0.9915020384733066
Sil score for k =  3 is 0.7424681785900861
Sil score for k =  4 is 0.46959765706212575
Sil score for k =  5 is 0.7888786239464297
Sil score for k =  6 is 0.5702297934970758
Sil score for k =  7 is 0.5699367036943114
Sil score for k =  8 is 0.7587368655106982
Sil score for k =  9 is 0.5076012030012443
```

Hình: 21. Tính điểm silhouette cho các cụm ngẫu nhiên

Huấn luyện mô hình

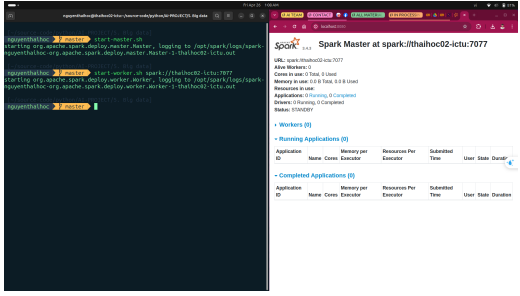
Áp dụng mô hình được huấn luyện để dự đoán cụm cho điểm dữ liệu: dụng mô hình để dự đoán cụm cho từng điểm dữ liệu:

Recency	Frequency	Monetary	Cluster
326	10	0.0	1
2	910	21550.0	0
75	155	8986.2	0
19	365	8787.75	0
310	85	1672.0	1

Hình: 22. Mô hình dự đoán cụm cho điểm dữ liệu

Triển khai ứng dụng lên Spark

- Khởi tạo một master và worker trên spark trong một máy cục bộ:



Hình: 23. Khởi tạo master và worker

Triển khai ứng dụng lên Spark

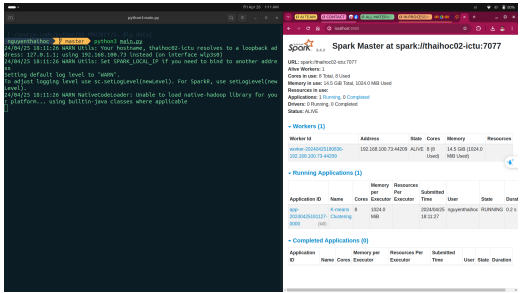
- Khởi tạo phiên ứng dụng Spark:

```
# import thư viện cần thiết

from pyspark.sql import SparkSession
from pyspark.sql.functions import col, isnan, when, count
import pandas as pd
import matplotlib.pyplot as pyplot
from pyspark.ml.feature import VectorAssembler
from pyspark.ml.feature import StandardScaler
from pyspark.ml.clustering import KMeans
from pyspark.ml.evaluation import ClusteringEvaluator

spark = SparkSession.builder.appName("K-means Clustering")
                            .master("spark://thaihoc02-ictu:7077")
                            .getOrCreate()
```

Hình: 24. Khởi tạo phiên ứng dụng Spark

Triển khai ứng dụng lên Spark

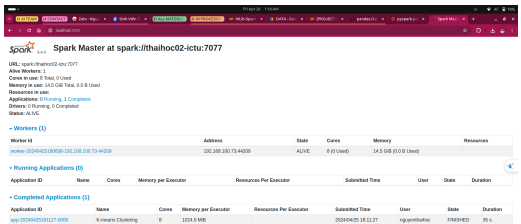
- Khởi chạy ứng dụng:



Hình: 25. Khởi chạy ứng dụng

Triển khai ứng dụng lên Spark

- Sau khi ứng dụng chạy xong tại giao diện spark phần Completed Application sẽ hiển thị ra các thông tin như thời gian khởi chạy, số thời gian chạy xong ứng dụng, tên người dùng và bộ nhớ:



Hình: 26. Kết thúc ứng dụng