ĐẠI HỌC QUỐC GIA HÀ NỘI

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

-----*-----

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

ĐỀ TÀI

RECOMMENDING ACTIONS FROM CLUSTERING ANALYSIS

Nhóm sinh viên thực hiện:

1. Nguyễn Tiến An – 20021080
2. Hồ Tú Minh – 22022674

Giảng viên hướng dẫn: TS. Trần Hồng Việt

Ths. Ngô Minh Hương

HÀ NỘI, 12/2024

MỤC LỤC

MỞ ĐẦU

Công nghệ Big Data đã vươn tới đỉnh cao trong việc thực hiện các chức năng của mình. Vào tháng 8/2015, Big Data chính thức bước ra khỏi bảng xếp hạng các công nghệ mới nổi của Cycle Hype do Gartner công bố, đánh dấu một cột mốc quan trọng và gây tiếng vang lớn trong xu hướng công nghệ toàn cầu. Big Data chứa đựng vô số thông tin giá trị, và nếu được khai thác hiệu quả, nó có thể mang lại lợi ích to lớn cho các lĩnh vực như y tế, giao thông, giáo dục, và nhiều ngành khác.

Do đó, các framework hỗ trợ xử lý Big Data cũng ngày càng được chú trọng phát triển mạnh mẽ. Một trong những công nghệ cốt lõi hiện nay là Apache Spark – một framework mạnh mẽ cho phép xử lý và phân tích dữ liệu lớn với tốc độ nhanh, dựa trên mô hình xử lý phân tán và bộ nhớ trong. Spark đã trở thành một lựa chọn hàng đầu trong việc khai thác tiềm năng của Big Data.

Vì vậy, chúng em đã chọn đề tài: "Recommending actions from clustering analysis" để làm báo kết thúc môn học của mình.

Bài báo cáo gồm 4 chương chính là:

Chương 1: Tổng quan về dữ liệu lớn

Chương 2: Phân cụm dữ liệu sử dụng thuật toán Kmeans Clustering

Chương 3: Ứng dụng thuật toán Kmeans trong bán hàng trực tuyến

Chương 4: Kết luận và hướng phát triển

CHƯƠNG 1: TỔNG QUAN VỀ DỮ LIỆU LỚN

1.1. Định nghĩa

Dữ liệu lớn là thuật ngữ dùng để mô tả một lượng dữ liệu khổng lồ và phức tạp tới mức các công cụ quản lý dữ liệu truyền thống không có khả năng thu thập, quản lý và xử lý dữ liệu. Dữ liệu lớn bao gồm việc phân tích, thu thập, giám sát dữ liệu, tìm kiếm, chia sẻ, lưu trữ, truyền nhận, trực quan, truy vấn và tính riêng tư.

Dựa trên cấu trúc dữ liệu, dữ liệu lớn có thể được phân loại thành ba loại chính:

- **Dữ liệu có cấu trúc**: Là loại dữ liệu dễ dàng nhất để quản lý và tìm kiếm. Dữ liệu có cấu trúc được lưu trữ và xử lý ở các định dạng cố định, có thể dễ dàng truy cập và xử lý bằng các công cụ như MySQL, Oracle, SQL Server. Ví dụ: Thông tin khách hàng, dữ liệu giao dịch, dữ liệu tài chính, …
- **Dữ liệu bán cấu trúc**: Dữ liệu này có một số cấu trúc nhất định nhưng không hoàn toàn tuân theo định dạng cố định. Chúng được xử lý bằng các công cụ truyền thống sau khi được xử lý sơ bộ. Ví dụ: email HTML, XML, JSON, …
- **Dữ liệu phi cấu trúc**: Dữ liệu này không có định dạng cố định và khó khăn trong việc xử lý bằng các công cụ truyền thống. Chúng chiếm phần lớn khối lượng dữ liệu Big Data. Ví dụ: email, tin nhắn, hình ảnh, video, âm thanh, dữ liệu cảm biến, nhật ký, …

1.2. Đặc trưng cơ bản

Dữ liệu lớn có ba đặc trưng chính là:

- Khối lượng (Volume): Là số lượng dữ liệu được tạo ra và lưu trữ.
- Tốc độ (Velocity): Là tốc độ các dữ liệu được tạo ra và xử lý để đáp ứng các nhu cầu thực tế.
- Đa dạng (Variety): Là chỉ sự đa dạng về các dạng và kiểu của dữ liệu. Dữ liệu được thu thập từ nhiều nguồn khác nhau và các kiểu dữ liệu cũng có rất nhiều cấu trúc khác nhau.

Ngoài ba đặc trưng chính trên, dữ liệu lớn còn có những đặc trưng khác như:

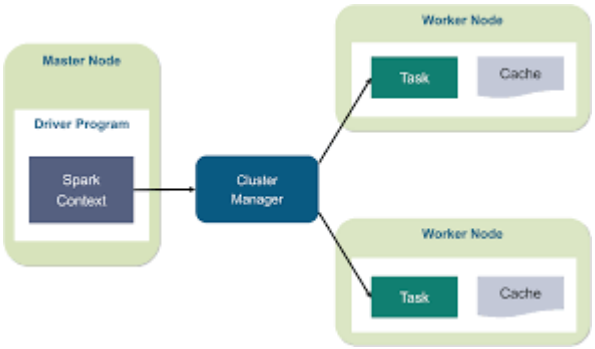
- Độ chính xác (Veracity): Là sự đảm bảo chất lượng dữ liệu cho một quy chuẩn nào đó phù hợp với bài toán thực tế. Chất lượng của dữ liệu thu được có thể khác nhau rất nhiều, ảnh hưởng đến sự phân tích chính xác.
- Tính biến đổi (Variability): Là sự biến đổi của dữ liệu về mặt bối cảnh, diễn giải hoặc có thể là cả phương pháp thu thập dữ liệu.
- Giá trị (Value): Là sự phù hợp về mặt dữ liệu với từng bài toán cụ thể.

1.3. Tổng quan về Apache Spark

Apache Spark là một nền tảng xử lý dữ liệu phân tán mạnh mẽ, được thiết kế để hỗ trợ các tác vụ xử lý dữ liệu lớn nhanh chóng và hiệu quả. Được phát triển tại AMPLab của Đại học California, Berkeley, Spark nổi bật nhờ khả năng thực hiện tính toán trong bộ nhớ (in-memory computing), giúp tăng tốc độ xử lý lên gấp nhiều lần so với các hệ thống dựa trên đĩa như Hadoop MapReduce.

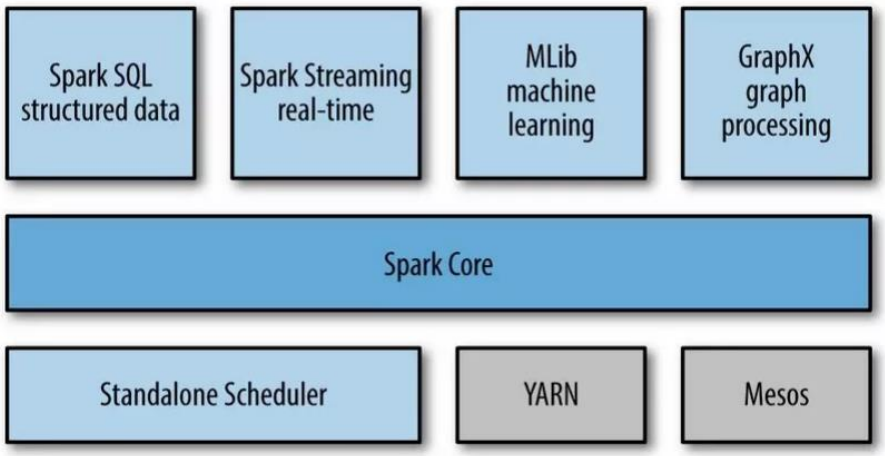
Spark cung cấp giao diện lập trình dễ sử dụng, hỗ trợ nhiều ngôn ngữ như Scala, Java, Python và R, giúp các nhà phát triển linh hoạt hơn trong việc triển khai các ứng dụng. Với kiến trúc mô-đun, Spark tích hợp nhiều thư viện mạnh mẽ như Spark SQL, MLlib (máy học), GraphX (xử lý đồ thị) và Spark Streaming (xử lý dữ liệu thời gian thực), đáp ứng đa dạng nhu cầu từ xử lý dữ liệu hàng loạt đến phân tích dữ liệu phức tạp. Đây là công cụ lý tưởng cho các doanh nghiệp muốn tối ưu hóa quy trình xử lý dữ liệu và khai thác triệt để giá trị từ dữ liệu lớn.

Kiến trúc Apache Spark:



- Driver Program: quản lý, điều khiển quá trình xử lý trên cluster và tạo Spark Context. Spark Context gồm tất cả các chức năng cơ bản. Spark Driver: chương trình chính của ứng dụng Spark, chạy trên một node trong cluster, quản lý và điều khiển quá trình xử lý trên cluster.
- Cluster Manage: quản lý và phân phối tài nguyên trên các node của cluster, phân phối và giám sát tiến trình xử lý trên các node để đảm bảo hoạt động hiệu quả.
- Executors: lên tiến trình chạy trên các node khác trong cluster, được quản lý bởi driver program để thực hiện các nhiệm vụ xử lý dữ liệu. Mỗi Executor có thể chứa nhiều task (công việc) được giao để xử lý.

Các thành phần của Apache Spark:



- Spark Core: Thành phần trung tâm xử lý dữ liệu và quản lý RDD.
- Spark SQL: Hỗ trợ truy vấn dữ liệu có cấu trúc bằng SQL.
- Spark Streaming: Xử lý dữ liệu thời gian thực.
- Mllib: Thư viện cung cấp các thuật toán máy học.
- GraphX: Công cụ phân tích và xử lý đồ thị.

- Standalone Scheduler: Quản lý tài nguyên nội bộ của Spark.
- YARN: Kết hợp Spark với hệ sinh thái Hadoop.
- Mesos: Phân bổ tài nguyên cho nhiều ứng dụng trên cụm.

1.4. Tổng quan về thuật toán Kmeans

K-Means là một thuật toán phân cụm phổ biến trong học máy, được sử dụng để nhóm các điểm dữ liệu thành k cụm dựa trên độ tương đồng. Mục tiêu chính là giảm thiểu tổng bình phương khoảng cách từ các điểm dữ liệu đến tâm cụm tương ứng (centroid). K-Means hoạt động hiệu quả với dữ liệu có cấu trúc rõ ràng và được ứng dụng rộng rãi trong nhiều lĩnh vực như phân tích khách hàng, nén hình ảnh, và phân đoạn thị trường.

Các bước thực hiện thuật toán K-means:

Bước 1: Khởi tạo: Chọn ngẫu nhiên kkk tâm cụm ban đầu (centroids).
Bước 2: Gán cụm: Gán mỗi điểm dữ liệu vào cụm có tâm gần nhất (dựa trên khoảng cách, thường là khoảng cách Euclid).
Bước 3: Cập nhật tâm cụm: Tính lại tâm cụm bằng cách lấy trung bình của tất cả các điểm dữ liệu trong cụm.
Bước 4: Lặp lại: Tiếp tục thực hiện hai bước trên (gán cụm và cập nhật tâm cụm) cho đến khi tâm cụm không còn thay đổi hoặc đạt đến số vòng lặp tối đa.
Bước 5: Kết quả: Thu được k cụm với các điểm dữ liệu thuộc về cụm tương ứng.

CHƯƠNG 2: PHÂN CỤM DỮ LIỆU SỬ DỤNG THUẬT TOÁN K-MEANS CLUSTERING

2.1. Giới thiệu thuật toán K-Means Clustering

- Giới thiệu:
    - Thuật toán K-Means Clustering được thiết kế để giải quyết bài toán học máy không giám sát (Unsupervised Learning)
    - Thuật toán cơ bản được thiết kế để làm việc với dữ liệu dạng số (numerical data), nhưng cũng có nhiều phương pháp và biến thể của thuật toán để làm việc với dữ liệu hạng mục (categorical data)
- Ý tưởng:
    - Nhóm các điểm dữ liệu có khoảng cách bé nhất lại với nhau. Khoảng cách thường được sử dụng là khoảng cách Euclidean

2.2. Triển khai thuật toán K-Means Clustering

- Input: dữ liệu **X** gồm *n* điểm dữ liệu và số lượng cụm cần tìm **K**
- Output: các điểm trung tâm **M** và vector nhãn cho từng điểm dữ liệu **Y**
- Các bước:
    1. Chọn K điểm bất kỳ làm các trung tâm cụm ban đầu
    2. Phân mỗi điểm dữ liệu vào cụm có trung tâm gần nó nhất
    3. Nếu việc gán từng điểm dữ liệu vào cụm ở bước 2 không thay đổi so với vòng lặp trước thì ta dừng thuật toán
    4. Cập nhật điểm trung tâm cho từng cụm bằng cách lấy trung bình cộng của tất cả các điểm dữ liệu đã được gán vào cụm đó sau bước 2
    5. Quay lại bước 2

2.3. Ví dụ minh họa thuật toán K-Means Clustering

- Dữ liệu: bộ dữ liệu này là các thông tin về khoản vay của các khách hàng tại 1 ngân hàng
    - *Gender*: giới tính, nhận 1 trong 2 giá trị ‘M’, ‘F’
    - *Applicant_Income*: thu nhập của khách hàng, nhận giá trị số
    - *Loan_Amount*: khoản vay, nhận giá trị số

ID	Gender	Applicant_Income	Loan_Amount
1	M	2000	240
2	F	2500	230
3	M	3000	220
4	M	4000	210
5	F	6000	200
6	F	8000	190
7	F	10000	180
8	M	12000	170
9	M	14000	160
10	F	2000	150
11	M	2500	140
12	F	3000	130
13	M	4000	120
14	F	6000	110
15	F	8000	100
16	F	10000	90
17	M	12000	80
18	M	14000	70
19	M	2000	60
20	M	2500	50
21	M	3000	40
22	M	4000	30

23	F	6000	20
24	F	8000	10

- Nhóm em chọn 2 cột *Applicant_Income* và *Loan_Amount* để phân loại các khoản vay thành các cụm khác nhau
- Các bước:
    1. Chọn K điểm ngẫu nhiên làm các trung tâm cụm ban đầu (với bài toán này nhóm em sẽ lấy K = 4 để minh hoạ)
        - *Cụm 1 (C1): điểm có ID 1(Applicant_Income = 2000, Loan_Amount = 240)*
        - *Cụm 2 (C2): điểm có ID 7(Applicant_Income = 10000, Loan_Amount = 180)*
        - *Cụm 3 (C3): điểm có ID 13(Applicant_Income = 4000, Loan_Amount = 120)*
        - *Cụm 4 (C4): điểm có ID 19(Applicant_Income = 2000, Loan_Amount = 60)*
        - Biểu diễn các điểm dữ liệu trên một đồ thị *xy* với trục *y* là khoản vay và trục *x* là thu nhập của các hàng
            - C1: (2000, 240)
            - C2: (10000, 180)
            - C3: (4000, 120)
            - C4: (2000, 60)
    2. Phân mỗi điểm dữ liệu vào cụm có trung tâm gần nó nhất
        - Tính khoảng cách từ mỗi điểm dữ liệu đến các trung tâm cụm bằng công thức khoảng cách Euclid dưới đây:

$$d = \sqrt{(ApplicantIncome_i - ApplicantIncome_c)^2 + (LoanAmount_i - LoanAmount_c)^2}$$

ID	Applicant_Income	Loan_Amount	d(C1)	d(C2)	d(C3)	d(C4)	phân vào cụm
1	2000	240	0	8000.224997	2003.596766	12001.20411	1
2	2500	230	500.09999	7500.166665	1504.027925	11501.11299	1
3	3000	220	1000.19998	7000.114285	1004.987562	11001.02268	1
4	4000	210	2000.224987	6000.075	90	10000.97995	3
5	6000	200	4000.199995	4000.05	2001.599361	8001.05618	3

6	8000	190	6000.20833	2000.025	4000.612453	6001.19988	2
7	10000	180	8000.224997	0	6000.299993	4001.512214	2
8	12000	170	10000.245	2000.025	8000.156248	2002.498439	2
9	14000	160	12000.26666	4000.05	10000.08	90	4
10	2000	150	90	8000.05625	2000.224987	12000.26666	1
11	2500	140	509.9019514	7500.106666	1500.133327	11500.21304	1
12	3000	130	1006.031809	7000.178569	1000.049999	11000.16364	3
13	4000	120	2003.596766	6000.299993	0	10000.125	3
14	6000	110	4002.111942	4000.612453	2000.025	8000.099999	3
15	8000	100	6001.633111	2001.599361	4000.05	6000.075	2
16	10000	90	8001.406126	90	6000.075	4000.05	2
17	12000	80	10001.27992	2002.498439	8000.099999	2000.025	4
18	14000	70	12001.20411	4001.512214	10000.125	0	4
19	2000	60	180	8000.899949	2000.899798	12000.00417	1
20	2500	50	534.8831648	7501.126582	1501.632445	11500.01739	1
21	3000	40	1019.803903	7001.39986	1003.194896	11000.04091	3
22	4000	30	2010.994779	6001.874707	90	10000.08	3
23	6000	20	4006.045432	4003.198721	2002.498439	8000.156248	3
24	8000	10	6004.406715	2007.211997	4001.512214	6000.299993	2

3. Nếu việc gán từng điểm dữ liệu vào cụm ở bước 2 không thay đổi so với vòng lặp trước thì ta dừng thuật toán
4. Cập nhật điểm trung tâm cho từng cụm bằng cách lấy trung bình cộng của tất cả các điểm dữ liệu đã được gán vào cụm đó sau bước 2

Minh họa với cụm 2 (cách tính các cụm còn lại tương tự) có các điểm với ID là 6, 7, 8, 15, 16, 24

- $ApplicantIncome_{C2} = \frac{8000 + 10000 + 12000 + 8000 + 10000 + 8000}{6} = 9333$
- $LoanAmount_{C2} = \frac{190 + 180 + 170 + 100 + 90 + 10}{6} = 123$
- Sau khi tính trung bình ta thu được điểm C2 mới với tọa độ (9333, 123)
- Tính các cụm còn lại tương tự

5. Quay lại bước 2

CHƯƠNG 3: ỨNG DỤNG THUẬT TOÁN KMEANS TRONG CÁC HỆ THỐNG GỢI Ý

3.1. Mô tả về bộ dữ liệu

Bộ dữ liệu Online Retail mô tả tất cả các giao dịch mua hàng được thực hiện cho một công ty bán lẻ trực tuyến có trụ sở tại Vương quốc Anh trong khoảng thời gian tám tháng. Đây là nguồn dữ liệu quan trọng để phân tích hành vi của khách hàng, từ đó phân tích nhu cầu mua hàng của khách hàng và đưa ra một số hành động phù hợp nhằm tăng doanh thu cho công ty.
Bộ dữ liệu chỉ bao gồm một tệp CSV, tệp CSV có 6 cột dữ liệu và chức năng như sau:

InvoiceNo	Là một số nguyên gồm 6 chữ số được gán duy nhất cho từng giao dịch. Nếu mã này bắt đầu bằng chữ cái 'C', điều đó cho thấy giao dịch đã bị hủy.
StockCode	Là một số nguyên gồm 5 chữ số được gán duy nhất cho từng sản phẩm cụ thể.
Description	Mô tả về sản phẩm
Quantity	Số lượng của từng sản phẩm (hàng hóa) trong mỗi giao dịch
InvoiceDate	Ngày và giờ lập hóa đơn. Là thời điểm mà giao dịch được tạo.
UnitPrice	Đơn giá. Giá của từng đơn vị sản phẩm, tính bằng đồng bảng Anh
CustomerID	Mã khách hàng. Là một số nguyên gồm 5 chữ số được gán duy nhất cho từng khách hàng.
Country	Tên quốc gia. Là tên quốc gia nơi khách hàng sinh sống.

Với những thuộc tính trong tệp dữ liệu trên, ta có thể trích xuất được những tệp khách hàng mua những loại sản phẩm khác nhau.

3.2. Chuẩn bị dữ liệu

Bài toán đặt ra là gợi ý các mặt hàng dựa trên nhu cầu mua hàng của người dùng. Ta sẽ tìm các mặt hàng gợi ý dựa trên mô hình RFM (Recency, Frequency, Monetary). Mô hình RFM (Recency, Frequency, Monetary) là một công cụ phân tích và phân khúc khách hàng dựa

trên hành vi tiêu dùng thông qua dữ liệu lịch sử giao dịch. Ba thành phần chính của RFM gồm:

- Recency (Gần đây): Đại diện cho khoảng thời gian kể từ lần giao dịch gần nhất của khách hàng. Khoảng thời gian càng ngắn, khả năng khách hàng tiếp tục mua sắm hoặc phản hồi tốt với các chiến dịch marketing càng cao. Ngược lại, khoảng thời gian dài có thể cảnh báo nguy cơ mất khách hàng.
- Frequency (Tần suất): Đo lường tần suất giao dịch của khách hàng với doanh nghiệp trong một khoảng thời gian cụ thể. Khách hàng giao dịch thường xuyên thể hiện mối quan hệ tốt với thương hiệu, đồng thời là đối tượng tiềm năng cho các chương trình bán chéo hoặc bán thêm.
- Monetary (Giá trị tiền tệ): Tổng số tiền mà khách hàng đã chi tiêu trong khoảng thời gian nhất định. Chỉ số này giúp đánh giá mức độ tiêu dùng của khách hàng và giá trị mà họ mang lại cho doanh nghiệp. Khách hàng chi tiêu lớn thường được coi là nhóm khách hàng trọng tâm.

Đầu tiên, ta loại bỏ những sản phẩm không có người mua, tức là những sản phẩm không có customerID:

```
[6]  df = df[df['CustomerID'].notnull()]
     df.info()
```

Tiếp theo, tính tổng giá trị sản phẩm đã mua:

```
[11] df['TotalSum'] = df['Quantity'] * df['UnitPrice']
     df.head()
```

Trích xuất các thuộc tính Recency, Frequency, Monetary. Từ các giá trị đã tính:

```
[12] rfm = df.groupby('CustomerID').agg({
         'InvoiceDay': lambda x: (pin_date - x.max()).days,
         'InvoiceNo': 'count',
         'TotalSum': 'sum'
     })
     rfm
```

CustomerID	InvoiceDay	InvoiceNo	TotalSum
12346.0	326	2	0.00
12347.0	3	182	4310.00
12348.0	76	31	1797.24
12349.0	19	73	1757.55
12350.0	311	17	334.40
...	...	...	...
18280.0	278	10	180.60
18281.0	181	7	80.82
18282.0	8	13	176.60
18283.0	4	756	2094.88
18287.0	43	70	1837.28

Đánh giá các thuộc tính trên từng sản phẩm bằng cách chuẩn hóa giá trị từ 0 đến 4, lưu file dưới định dạng csv để chuẩn bị thực hiện thuận toán phân cụm K-means trong Apache Spark:

```
[14] r_labels = range(4, 0, -1) #[4, 3, 2, 1]
     r_groups = pd.qcut(rfm['Recency'], q=4, labels=r_labels)
     f_labels = range(1, 5) # [1, 2, 3, 4]
     f_groups = pd.qcut(rfm['Frequency'], q=4, labels=f_labels)
     m_labels = range(1, 5)
     m_groups = pd.qcut(rfm['Monetary'], q=4, labels=m_labels)
```

```
[15] rfm['R'] = r_groups.values
     rfm['F'] = f_groups.values
     rfm['M'] = m_groups.values
     rfm
     rfm.to_csv('rfm.csv')
```

```
rfm
```

CustomerID	Recency	Frequency	Monetary	R	F	M
12346.0	326	2	0.00	1	1	1
12347.0	3	182	4310.00	4	4	4
12348.0	76	31	1797.24	2	2	4
12349.0	19	73	1757.55	3	3	4
12350.0	311	17	334.40	1	1	2
...	...	...	...	...	...	...

3.3. Thực hiện thuật toán K-means

Ta thực hiện thuật toán K-means với số cụm là 10, số bước thực hiện là 300, dưới đây là mã nguồn thực hiện thuật toán K-means:

```
[32] from pyspark.ml.feature import VectorAssembler, StandardScaler
     from pyspark.ml.clustering import KMeans
     from pyspark.ml.evaluation import ClusteringEvaluator

     columns_for_clustering = ["R", "F", "M"]
     assembler = VectorAssembler(inputCols=columns_for_clustering, outputCol="features")
     assembled_df = assembler.transform(df)

     scaler = StandardScaler(inputCol="features", outputCol="scaledFeatures", withStd=True, withMean=False)
     scaler_model = scaler.fit(assembled_df)
     scaled_df = scaler_model.transform(assembled_df)
```

```
[27] kmeans = KMeans(k=10, maxIter=300, seed=42, featuresCol="scaledFeatures", predictionCol="prediction")
     model = kmeans.fit(scaled_df)
```

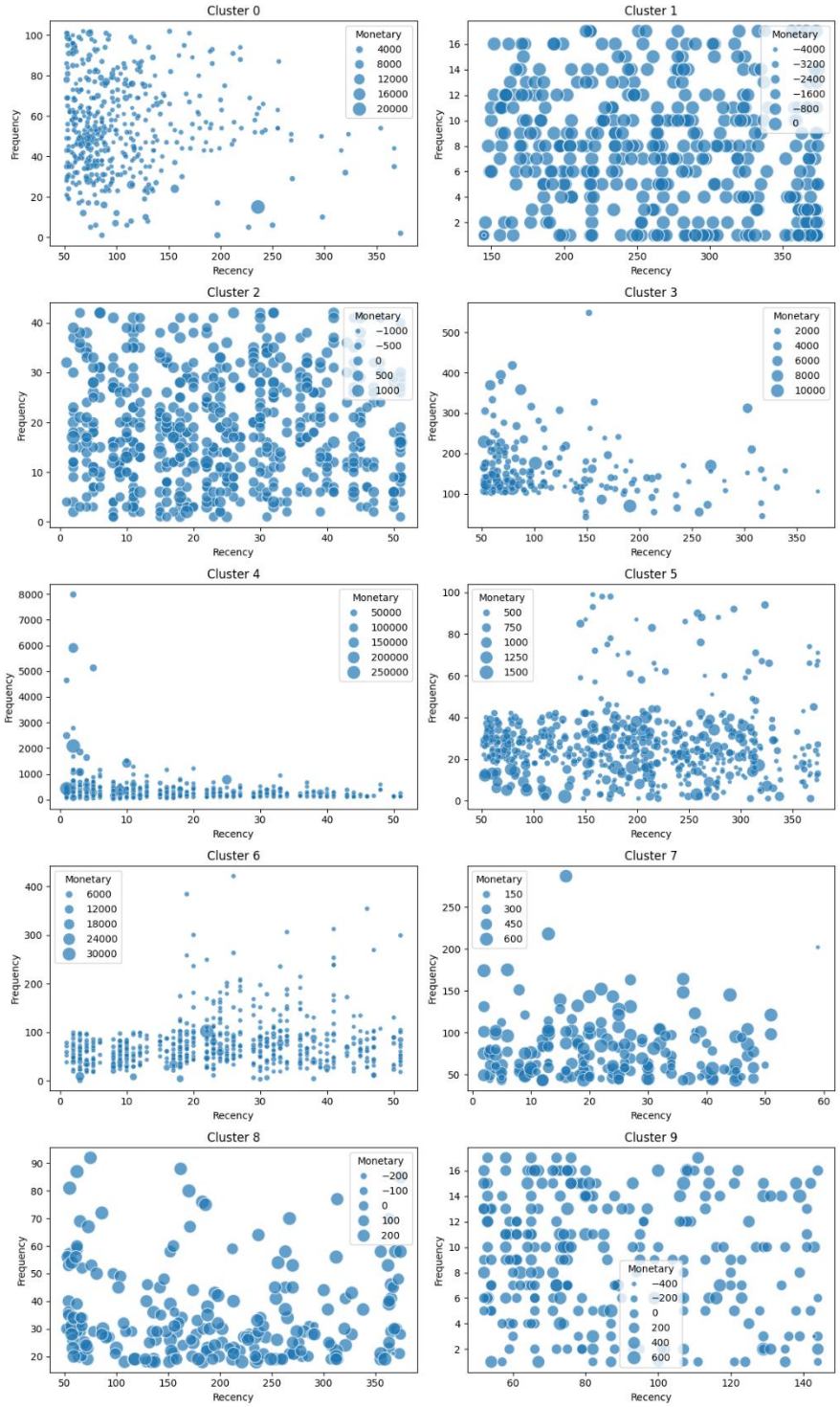
```
[41] predictions = model.transform(scaled_df)
     predictions = predictions.withColumnRenamed("prediction", "kmeans_cluster")
     predictions.show()
     result_df = predictions.select(*columns_for_clustering, "kmeans_cluster")
     result_df.show()
     centers = model.clusterCenters()
     print("Cluster Centers: ")
     for center in centers:
         print(center)
```

Sau khi đã phân loại được tệp khách hàng, ta sẽ gợi ý được những sản phẩm theo từng cụm khách hàng:

```
from pyspark.sql import functions as F

dff = spark.read.csv("/content/OnlineRetail.csv", header=True, inferSchema=True)

num_clusters = 10

cluster_recommendations = {}

for cluster_id in range(num_clusters):
    customers_in_cluster = predictions.filter(predictions.kmeans_cluster == cluster_id).select("CustomerID").distinct()

    cluster_transactions = dff.join(customers_in_cluster, on="CustomerID", how="inner")

    top_products_for_cluster = (
        cluster_transactions.groupBy("StockCode")
        .agg(F.count("InvoiceNo").alias("transaction_count"))
        .orderBy(F.desc("transaction_count"))
        .limit(10)
        .select("StockCode")
        .rdd.flatMap(lambda x: x)
        .collect()
    )

    cluster_recommendations[f"Cluster {cluster_id}"] = top_products_for_cluster

for cluster, recommended_products in cluster_recommendations.items():
    print(f"{cluster} -> Recommended Products: {recommended_products}")
```

3.4. Kết quả

Ta có đồ thị biểu diễn ba thuộc tính RFM (Recency, Frequency, Monetary) của từng cụm khách hàng:



Từ đồ thị trên, ta có được gợi ý những sản phẩm cho từng cụm khách hàng, ta sẽ gợi ý 10 sản phẩm:

```
Cluster 0 -> Recommended Products: ['22423', '85123A', '84879', '47566', 'POST', '22720', '22960', '85099B', '21212', '20725']
Cluster 1 -> Recommended Products: ['85123A', '22423', 'POST', '47566', '22457', '22427', '22969', '84879', '84946', '21232']
Cluster 2 -> Recommended Products: ['84879', 'POST', '22086', '22423', '22138', '23084', '23355', '85123A', '21034', '84946']
Cluster 3 -> Recommended Products: ['85123A', '22423', '47566', '22720', '84879', '85099B', '82482', '22960', '22961', '23298']
Cluster 4 -> Recommended Products: ['85123A', '85099B', '22423', '20725', '23203', '47566', '20727', '22197', '84879', '23209']
Cluster 5 -> Recommended Products: ['85123A', '22423', '47566', 'POST', '22720', '22960', '84879', '22457', '22470', '22178']
Cluster 6 -> Recommended Products: ['22423', '85123A', 'POST', '84879', '85099B', '23084', '22086', '22138', '20727', '47566']
Cluster 7 -> Recommended Products: ['21034', '22086', '23084', '85123A', '23321', '22197', '22469', '22578', '23301', '22910']
Cluster 8 -> Recommended Products: ['21034', '85123A', '47566', '22960', '22138', 'M', '22469', '22804', '22197', '22139']
Cluster 9 -> Recommended Products: ['84946', '84879', '85099B', '22423', '85123A', '47566', '22178', '22487', 'M', '21790']
```

CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

4.1. Kết luận

Với kiến trúc linh hoạt và hiệu suất cao, Apache Spark không chỉ đơn thuần là một công cụ xử lý dữ liệu lớn, mà còn là một nền tảng mạnh mẽ hỗ trợ các tác vụ trong lĩnh vực khoa học dữ liệu và trí tuệ nhân tạo. Điểm nổi bật của Apache Spark nằm ở khả năng xử lý dữ liệu phân tán với tốc độ cao, tích hợp các thư viện chuyên biệt như MLlib, Spark SQL, và GraphX, giúp thực hiện hiệu quả các bài toán phức tạp và trích xuất thông tin giá trị từ khối lượng dữ liệu khổng lồ. Thông qua bài tập lần này, nhóm chúng em đã có cơ hội làm việc với Apache Spark nói riêng và dữ liệu lớn nói chung, qua đó tích lũy thêm kiến thức và kinh nghiệm trong việc xử lý và phân tích dữ liệu. Dự án của nhóm em đã thành công trong việc áp dụng thuật toán K-means và phân tích cụm để đưa ra các gợi ý về sản phẩm (Recommending Actions) cho công ty để tăng doanh thu.

Tuy nhiên, dự án này của nhóm chúng em còn hạn chế:

- Bộ dữ liệu chưa đủ lớn để đảm bảo tính chính xác cao.
- Cách thực hiện thuật toán phân cụm vẫn chưa tối ưu (cách xác định số cụm, …)

4.2. Hướng phát triển

Từ kết quả thu được sau khi thực hiện đề tài, chúng em có những hướng phát triển sau:

- Mở rộng thử nghiệm với những bộ dữ liệu có khối lượng rất lớn.
- Mở rộng hệ thống, tích hợp thêm một số công nghệ như NoSQL để lưu trữ dữ liệu.

TÀI LIỆU THAM KHẢO

[1] https://www.kaggle.com/datasets/vijayuv/onlineretail/code
[2] https://blog.tomorrowmarketers.org/phan-tich-rfm-la-gi/
[3] https://machinelearningcoban.com/2017/01/01/kmeans/
[4] https://www.tutorialspoint.com/apache_spark/index.htm
[5] https://www.tutorialspoint.com/pyspark/index.htm

PHÂN CÔNG NHIỆM VỤ

Thành viên	Công việc
Nguyễn Tiến An	Tìm hiểu đề tài Lên ý tưởng về bài toán, dữ liệu Tìm hiểu về Spark, Kmeans Viết báo cáo phần 3,4 Cài đặt chương trình Làm Slides Chỉnh sửa báo cáo, slides
Hồ Tú Minh	Tìm hiểu đề tài Lên ý tưởng về bài toán, dữ liệu Tìm hiểu về Spark, Kmeans Viết báo cáo phần 1,2 Chạy chương trình demo và đánh giá Làm Slides Chỉnh sửa báo cáo, slides Thuyết trình