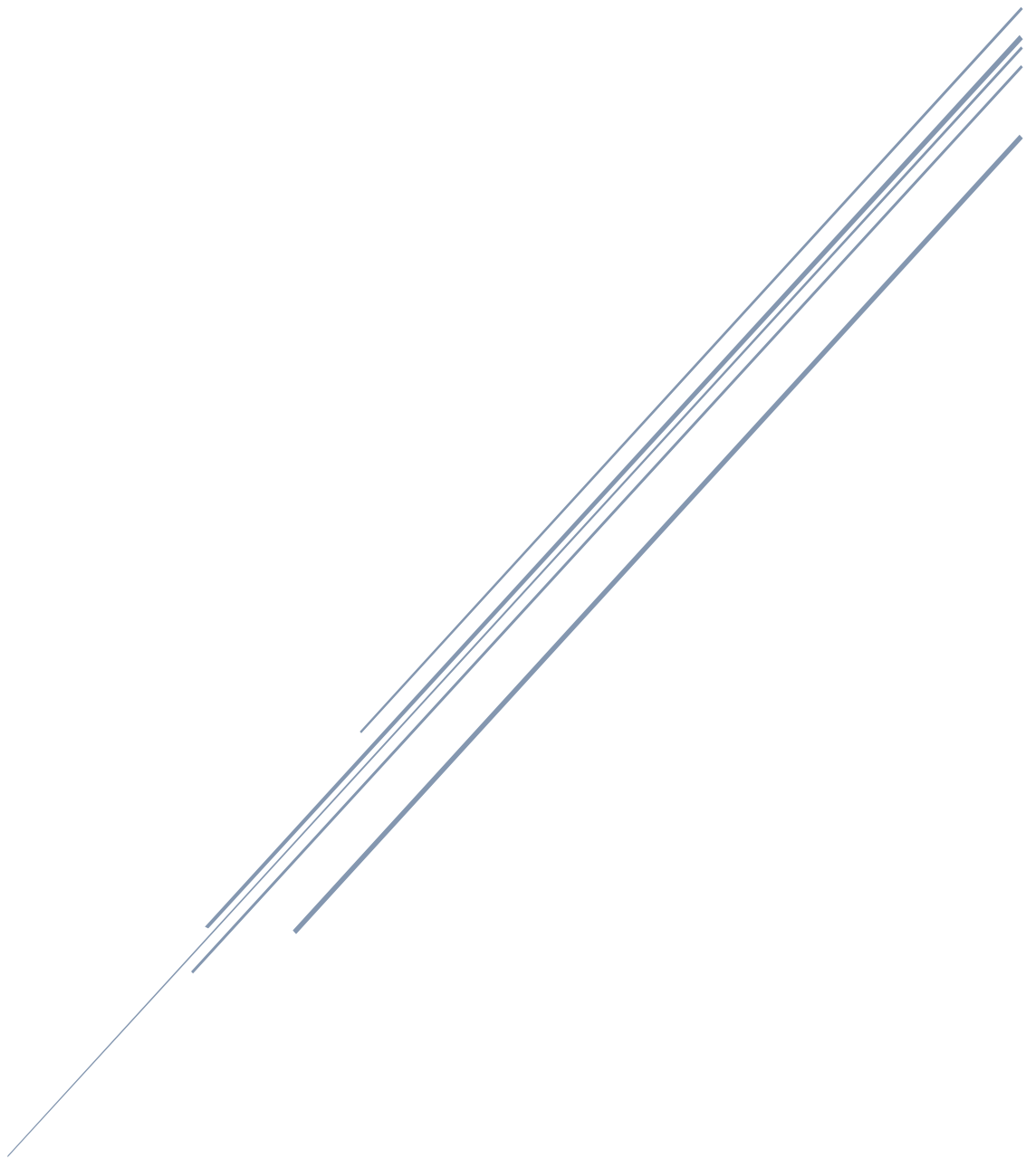


PHÂN TÍCH RFM BẰNG PHÂN CỤM K-MEANS VỚI CÔNG CỤ PYSPARK



PHẦN 1: NỘI DUNG BÁO CÁO

1. Giới thiệu đề tài

Hiện nay, mục tiêu hiện nay của các doanh nghiệp đó là quản lý, phục vụ và chăm sóc khách hàng một cách tốt nhất. Chính vì vậy việc thấu hiểu khách hàng, cố gắng đem những sản phẩm, dịch vụ, trải nghiệm tốt nhất, phù hợp nhất luôn luôn là ưu tiên hàng đầu của các doanh nghiệp. Để làm được điều đó, doanh nghiệp cần phân chia khách hàng thành các nhóm khác nhau được gọi là phân khúc khách hàng, nhằm tập trung hóa và chăm sóc khách hàng tốt hơn dựa vào các đặc điểm riêng của từng phân khúc khách hàng.

Ngày nay, cùng với cuộc cách mạng khoa học công nghệ 4.0 là sự phát triển mạnh mẽ của khoa học dữ liệu, việc thu thập và lưu trữ dữ liệu của khách hàng là nguồn tài nguyên mang lại nhiều giá trị tiềm năng cho doanh nghiệp. Nhờ vào việc phân tích dựa vào dữ liệu, các quyết định của người quản lý có tính khách quan và sự đa chiều hơn, giảm bớt được sự cảm tính khó đo lường được. Chính vì vậy, việc áp dụng phân tích dữ liệu vào việc chia phân khúc khách hàng sẽ góp phần vào sự thành công của trong chiến lược hay các chính sách chăm sóc khách hàng nói riêng cũng như sự phát triển của doanh nghiệp nói chung trong giai đoạn thị trường cạnh tranh ngày càng khốc liệt.

Để giải quyết được vấn đề trên, trong báo cáo này sẽ tập trung vào bài toán phân chia phân khúc khách hàng với mô hình dựa trên sự kết hợp giữa nền tảng kinh doanh và công nghệ thông tin. Mà cụ thể nền tảng kinh doanh ở đây là vận dụng lý thuyết phân tích RFM và công nghệ thông tin ở đây là áp dụng phương pháp học máy không giám sát phân cụm K-Means.

Nội dung tiếp theo của bài báo cáo là phần 2 và phần 3 gồm các nghiên cứu liên quan và các lý thuyết, công cụ, độ đo đánh giá nhằm xác định các mô hình, thuật toán phù hợp với mục tiêu đặt ra. Các vấn đề liên quan và quá trình thực hiện được mô tả trong phần 4 - phương pháp nghiên cứu. Sau đó là quá trình thực nghiệm sẽ được trình bày ở phần 5 và thảo luận kết quả. Cuối cùng là kết luận và hướng phát triển của nghiên cứu.

2. Các nghiên cứu liên quan

Với sự phát triển mạnh mẽ của công nghệ khoa học dữ liệu ngày nay, có rất nhiều bài nghiên cứu phân tích hành vi khách hàng trong lĩnh vực bán lẻ với mô hình RFM. Trong đó phải kể đến tác giả Phan Châu Minh Trường đã sử dụng phương pháp phân cụm K-means và mô hình RFM để phân tích rõ hành vi tập trung vào việc phân khúc khách hàng và được thực hiện bằng cách dùng các kỹ thuật học máy không giám sát ứng dụng vào

mô hình RFM. Từ kết quả nghiên cứu, tác giả đã phân tích hành vi khách hàng, hiểu từng phân khúc khách hàng [1]

Mai Thị Kim Ngân đã sử dụng kỹ thuật phân cụm K-Means để doanh nghiệp có thể phân nhóm khách hàng để chăm sóc tốt hơn và phân tích RFM để phân cụm khách hàng. Ngoài ra, còn sử dụng luật kết hợp là Apriori và FPGrowth, được dùng trong bài toán phân tích sản phẩm khách hàng hay mua [2]

3. Các lý thuyết, công cụ, độ đo đánh giá

3.1. Cơ sở lý thuyết

3.1.1. Giới thiệu về Apache Spark

Apache Spark là một framework mã nguồn mở tính toán cụm, ban đầu được phát triển vào năm 2009 bởi AMPLab. Sau đó, Spark được trao cho Apache Software Foundation vào năm 2013 và phát triển cho đến nay.[4]

Tốc độ xử lý của Spark đạt được là do việc tính toán được thực hiện cùng lúc trên nhiều máy khác nhau. Đồng thời việc tính toán được thực hiện trong bộ nhớ trong hoặc được thực hiện hoàn toàn trong RAM.[4]

Spark cho phép xử lý dữ liệu theo thời gian thực, vừa nhận dữ liệu từ các nguồn khác nhau và ngay lập tức thực hiện xử lý trên dữ liệu nhận được.[4]

3.1.2. Giới thiệu về RFM

Phương pháp RFM thường được sử dụng trong phân khúc nhóm khách hàng và tìm hiểu đặc điểm của từng phân khúc khách hàng. Phương pháp RFM cho phép các marketer nhắm mục tiêu các nhóm khách hàng cụ thể bằng các thông tin liên lạc phù hợp hơn nhiều với các hành vi cụ thể của họ – và do đó tạo ra tỷ lệ phản hồi cao hơn nhiều, cộng với sự gia tăng lòng trung thành và giá trị lâu dài của khách hàng. RFM là viết tắt của ba yếu tố bao gồm:

- Recency: là lần cuối cùng khách hàng đã mua hàng. Những khách hàng đã mua hàng của bạn gần đây có nhiều khả năng sẽ mua lại hàng của bạn hơn những khách hàng từ quá khứ xa xưa. Đây là một yếu tố xếp hạng quan trọng – đó là lý do tại sao nó đứng đầu danh sách.
- Frequency: là tần suất mua của khách hàng hoặc khách hàng đã mua hàng bao nhiêu lần. Một khách hàng đến mỗi ngày có nhiều khả năng mua lại hơn những người chỉ đến một lần mỗi năm.

- Monetary: là tổng số tiền mà khách hàng có chi tiêu cho tất cả các lần mua hàng. Một khách hàng thực hiện một giao dịch mua lớn có nhiều khả năng sẽ mua lại hơn một khách hàng chi tiêu ít hơn rất nhiều.

3.1.3. Giới thiệu về K-means

Phân cụm K-means là một thuật toán học không giám sát được sử dụng để giải quyết các vấn đề về phân cụm. Trong thuật toán phân cụm K-means, chúng tôi không biết nhãn của từng điểm dữ liệu [5]. Mục tiêu là làm thế nào để chia dữ liệu thành các cụm khác nhau để dữ liệu trong cùng một cụm có các thuộc tính giống nhau.[5]

3.2. Công cụ, độ đo đánh giá

Ban đầu việc thiết lập phương pháp RFM khá dễ dàng nhưng về sau công nghệ và nhu cầu của con người phát triển, khiến các yếu tố ảnh hưởng đến RFM cũng thay đổi khác nhau. Trong các nghiên cứu gần đây, các data analyst đã ứng dụng và cải tiến phương pháp RFM bằng việc sử dụng các thuật toán học không giám sát. Cụ thể là sử dụng phương pháp phân cụm K-means để phân chia nhóm khách hàng dựa trên ba yếu tố trong phương pháp RFM. Mỗi phân khúc khách hàng hiện được coi là một cụm trong K-means. Áp dụng học máy không giám sát có thể dễ dàng phân tích phân khúc khách hàng và tìm ra những giá trị đích thực có khả năng tác động, ảnh hưởng đến hành vi và quyết định mua hàng của khách hàng.

4. Phương pháp nghiên cứu

4.1. Mô hình đề xuất

Hình trình bày mô hình đề xuất với 2 giai đoạn chính như sau. Giai đoạn 1 thu thập từ dữ liệu đầu vào là tập dữ liệu, sau đó khám phá và tiền xử lý dữ liệu. Sau đó từ các thuộc tính cần thiết từ tập dữ liệu để tính toán ba giá trị quan trọng là Recency, Frequency, Monetary và cuối cùng là hoàn chỉnh dữ liệu theo mô hình RFM. Giai đoạn 2 là giai đoạn cuối cùng cũng là giai đoạn có độ phức tạp nhất. Ở giai đoạn này, sẽ lựa chọn các phương pháp, mô hình phù hợp với dữ liệu nhằm giải quyết việc chuẩn hóa dữ liệu đầu vào và thực hiện phương pháp gom cụm K-Means để phân khúc khách hàng. Từ đó, đưa ra các kết quả phân tích, các quyết định nhóm khách hàng dựa trên kết quả phân cụm K-Means đã được trực quan hóa qua biểu đồ.

4.2. Thu thập dữ liệu

Phương pháp được thực hiện trên tập dữ liệu của một cửa hàng bán lẻ trực tuyến quà tặng và phụ kiện. Trong đó đa số khách hàng của cửa hàng là nhà bán lẻ. Các giao dịch này được thực hiện từ năm 2010 đến năm 2011. Bộ dữ liệu được minh họa ở hình

InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
536365	85123A	WHITE HANGING HEA...	6	12/1/2010 8:26	2.55	17850	United Kingdom
536365	71053	WHITE METAL LANTERN	6	12/1/2010 8:26	3.39	17850	United Kingdom
536365	84406B	CREAM CUPID HEART...	8	12/1/2010 8:26	2.75	17850	United Kingdom
536365	84029G	KNITTED UNION FLA...	6	12/1/2010 8:26	3.39	17850	United Kingdom
536365	84029E	RED WOOLLY HOTTIE...	6	12/1/2010 8:26	3.39	17850	United Kingdom
536365	22752	SET 7 BABUSHKA NE...	2	12/1/2010 8:26	7.65	17850	United Kingdom
536365	21730	GLASS STAR FROSTE...	6	12/1/2010 8:26	4.25	17850	United Kingdom
536366	22633	HAND WARMER UNION...	6	12/1/2010 8:28	1.85	17850	United Kingdom
536366	22632	HAND WARMER RED P...	6	12/1/2010 8:28	1.85	17850	United Kingdom
536367	84879	ASSORTED COLOUR B...	32	12/1/2010 8:34	1.69	13047	United Kingdom
536367	22745	POPPY'S PLAYHOUSE...	6	12/1/2010 8:34	2.1	13047	United Kingdom
536367	22748	POPPY'S PLAYHOUSE...	6	12/1/2010 8:34	2.1	13047	United Kingdom
536367	22749	FELTCRAFT PRINCES...	8	12/1/2010 8:34	3.75	13047	United Kingdom
536367	22310	IVORY KNITTED MUG...	6	12/1/2010 8:34	1.65	13047	United Kingdom
536367	84969	BOX OF 6 ASSORTED...	6	12/1/2010 8:34	4.25	13047	United Kingdom
536367	22623	BOX OF VINTAGE JI...	3	12/1/2010 8:34	4.95	13047	United Kingdom
536367	22622	BOX OF VINTAGE AL...	2	12/1/2010 8:34	9.95	13047	United Kingdom
536367	21754	HOME BUILDING BLO...	3	12/1/2010 8:34	5.95	13047	United Kingdom
536367	21755	LOVE BUILDING BLO...	3	12/1/2010 8:34	5.95	13047	United Kingdom
536367	21777	RECIPE BOX WITH M...	4	12/1/2010 8:34	7.95	13047	United Kingdom

Dữ liệu gồm có các thuộc tính CustomerID mỗi một đơn hàng sẽ ứng với một khách hàng; InvoiceNo mỗi đơn hàng sẽ có mã số hóa đơn riêng, mã số hóa đơn này sẽ giúp phân biệt với các hóa đơn khác. Thuộc tính này sẽ giúp xác định được giá trị Frequency. Với thuộc tính Quantity đã mua mỗi hóa đơn, UnitPrice; ta có công thức $Quantity \times Price$ có thể xác định được tổng số tiền trên mỗi món hàng và từ đó xác định được thành tiền của mỗi đơn hàng. Thuộc tính này dùng để tính giá trị Monetary. Thuộc tính InvoiceDate dùng để tính giá trị Recency.

4.3. Tiền xử lý, làm sạch, chuẩn hóa dữ liệu mô hình RFM

Sau quá trình khảo sát và tiền xử lý cũng như loại bỏ các giá trị Null hay bị lỗi và giữ lại các giá trị phù hợp với việc thực hiện. Ta có mô hình dữ liệu RFM được xây dựng và kết quả được trình bày ở Hình

summary	CustomerID	Recency	Frequency	Monetary
count	4339	4339	4339	4339
mean	15299.936851809172	92.0414842129523	4.271952062687255	2053.7930168241505
stddev	1721.8897579594227	100.00775734416375	7.70549277131483	8988.248381460095
min	12346	0	1	0.0
max	18287	373	210	280206.02

Thông qua mô hình dữ liệu RFM có thể thấy các giá trị Recency, Frequency và Monetary, ta có thể nhận thấy có sự khác biệt về đơn vị và độ chênh lệch phạm vi giá trị rất lớn giữa ba giá trị Recency, Frequency và Monetary.

Giá trị Recency trải dài từ 0 đến 373 (ngày mua hàng gần nhất), Frequency trải dài từ 1 đến 210 (lần mua hàng), Monetary là giá trị có miền giá trị lớn nhất từ 0 đến 280206.02 (đơn vị tiền tệ). Nhìn vào Hình có thể thấy Monetary có giá trị lớn hơn rất nhiều so với

hai yếu tố còn lại. Chính vì vậy ta cần chuẩn hóa dữ liệu (scaling data) để cho ra kết quả khi huấn luyện mô hình một cách tốt nhất. Hình dưới đây mô tả việc chuẩn hóa dữ liệu từ mô hình RFM.

CustomerID	rfm_features	rfm_standardized
15555	[12.0,16.0,4805.17]	[0.11999069190906...
15574	[177.0,4.0,702.25]	[1.76986270565869...
15634	[17.0,1.0,243.55]	[0.16998681353784...
13610	[12.0,7.0,1131.88]	[0.11999069190906...
13192	[95.0,2.0,911.94]	[0.94992631094675...

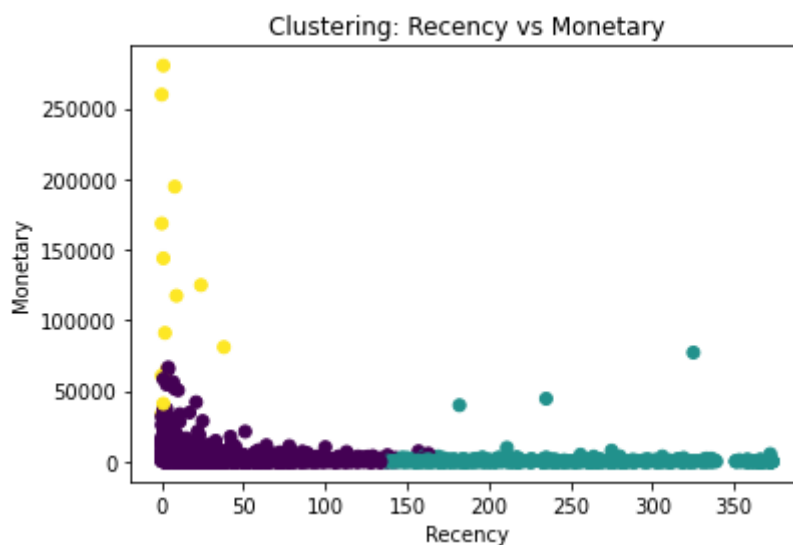
4.4. Phân khúc khách hàng bằng gom cụm K-Means

Sau khi đã chuẩn hóa dữ liệu, tiếp theo sẽ huấn luyện mô hình RFM với thuật toán học máy gom cụm K-Means. Ta chia với 3 nhóm khách hàng ứng với cluster k=3. Kết quả dự đoán (prediction) với mỗi CustomerID sẽ ứng với một nhóm khách hàng là 0, 1, 2 với đặc trưng với mỗi cụm được minh họa ở Hình

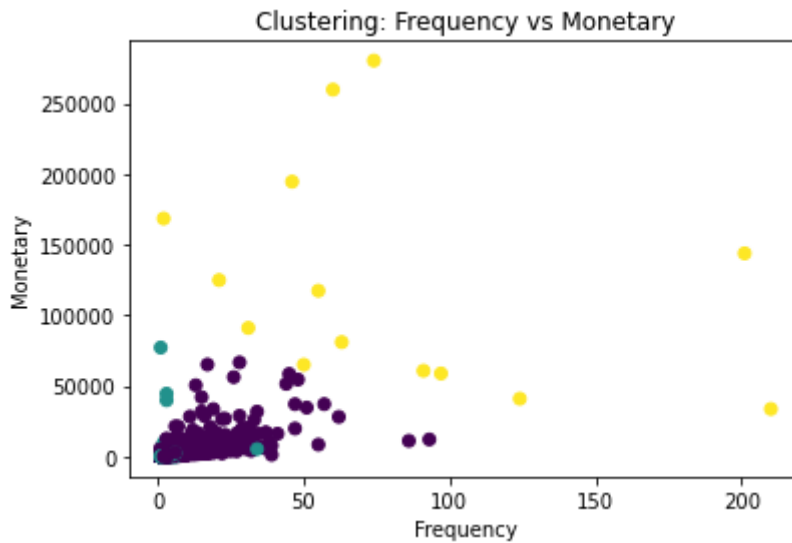
CustomerID	Recency	Frequency	Monetary	prediction
13192	95	2	911.94	0
13610	12	7	1131.88	0
14157	19	2	432.88	0
15555	12	16	4805.17	0
15574	177	4	702.25	1

5. Kết quả thực nghiệm và phân tích

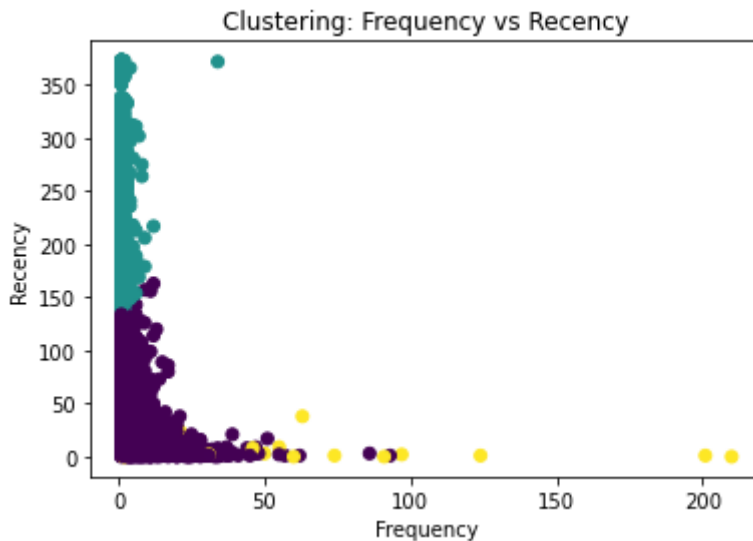
Trực quan hóa dữ liệu bằng biểu đồ Recency và Monetary ở Hình: từ biểu đồ chúng ta có thể nói rằng nhóm màu vàng là những người có chi tiêu nhiều hơn và họ là những khách hàng gần đây.



Với biểu đồ Frequency và Monetary: nhìn vào có thể thấy ở biểu đồ này, nhóm khách hàng màu vàng luôn là nhóm mua nhiều hơn và thường xuyên hơn, nhưng phân cụm này tương đối ít và rải rác, nên có thể đây là những dữ liệu khó xác định được. Trong khi nhóm màu tím có nhu cầu mua sắm gần đây và mức chi tiêu vừa phải và ổn định.



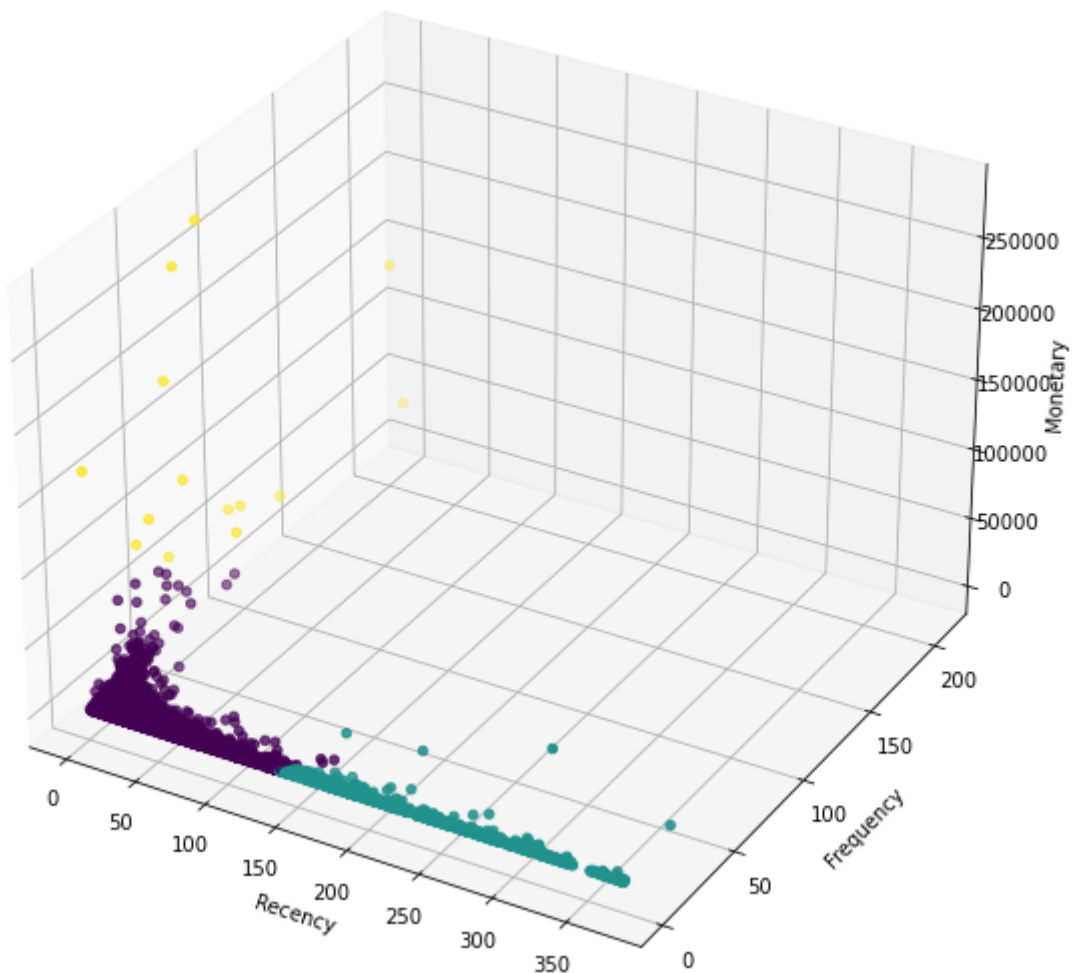
Biểu đồ Frequency và Recency: ở đây, nhóm khách hàng màu vàng vẫn thường xuyên mua sản phẩm và thuộc nhóm gần đây nhất. Nhóm màu tím đã cố gắng mua gần đây nhất nhưng họ không phải là người mua thường xuyên nhất, nên suy ra có thể xác định rằng đây có thể là nhóm khách hàng khách hàng trung thành của doanh nghiệp.



Hình dưới đây là biểu đồ 3D của tất cả các khách hàng được phân khúc. Có thể thấy mật độ các điểm của cụm màu tím và màu xanh lá tương đối ổn định. Còn với cụm màu vàng thì có số lượng phần tử ít hơn và rải rác hơn, nên có thể đây là các dữ liệu ngoại lai. Chính vì vậy rất khó xác định được nhóm khách hàng phân cụm màu vàng là nhóm khách hàng gì.

Nhìn vào biểu đồ 3D có thể dự đoán, nhóm cụm màu tím có thể là nhóm khách hàng trung thành của doanh nghiệp. Ngày mua hàng gần nhất của nhóm khách hàng này nằm trong nhóm tốt nhất. Nhóm khách hàng này có thể sẵn sàng chi nhiều tiền hơn cho hoạt động mua sắm. Với các đặc điểm về Recency, Frequency và Monetary, đây không chỉ là nhóm khách hàng trung thành mà đây còn là nhóm khách hàng tiềm năng mang lại lợi ích to lớn cho doanh nghiệp.

Với cụm màu xanh lá, có thể nói đây là nhóm khách hàng có số lượng khá đông đảo, đây là nhóm khách hàng phổ thông của doanh nghiệp. Trong đó mức chi tiêu không quá cao và thấp hơn nhóm khách hàng trung thành cụm tím nhưng lại chiếm một tỉ lệ khá cao. Về giá trị Recency và Frequency thì suy trì ở mức độ ổn định hơn. Với nhóm khách hàng này, doanh nghiệp có thể tiếp tục cải thiện các chính sách bán hàng để có thể giữ chân nhóm khách hàng này ở lại với doanh nghiệp.



6. Kết luận và hướng phát triển

Kết quả từ phân cụm RFM K-Means đã giúp nhóm các phân khúc khách hàng để đưa ra các giải pháp kinh doanh cần thiết. Mô hình này được thực nghiệm đầy đủ các bước với bộ dữ liệu với ba yếu tố Recency, Frequency và Monetary được quan tâm. Để khai

thác mô hình RFM một cách hiệu quả, phương pháp phân cụm K-Means đã được kết hợp với phân tích RFM bằng Apache Spark mà cụ thể ở đây là Pyspark để phân khúc khách hàng. Từ đó, chúng ta có thể thấy công ty đã làm tốt trong việc duy trì được lượng khách hàng trung thành và tìm kiếm thêm những khách hàng mới. Thực hiện phân khúc khách hàng giúp công ty hiểu được khách hàng của mình để đưa ra các kế hoạch về chiến lược liên quan đến khách hàng, bên cạnh việc có thể giữ chân được những khách hàng trung thành, thu hút được lòng tin nhiệm của khách hàng mới, mà còn có thể tác động đến phân khúc chuẩn bị rời bỏ, giảm được lượng khách hàng chỉ đến một lần và kéo khách hàng đã mất trở lại.

Nhóm em cũng đã sử dụng Chuỗi thời gian để dự đoán trong tương lai, công ty sẽ nhận về được doanh thu là bao nhiêu. Nếu công ty nỗ lực và duy trì áp dụng phương pháp phân khúc khách hàng, doanh thu và lợi nhuận trong tương lai sẽ tăng lên đáng kể.

Do bộ dữ liệu mà nhóm sử dụng không lớn bằng cấp độ big data nên nhóm chúng em khó có thể lấy được nhiều thông tin từ bộ dữ liệu bằng cách sử dụng các quy tắc tương quan, v.v. Chúng em cũng đã không triển khai RFM một cách triệt để và chính xác do thiếu kiến thức và kỹ năng thực hành về công nghệ liên quan. Cuối cùng, do thiếu kinh nghiệm phân tích kinh doanh nên các phân tích của chúng em về dữ liệu có thể chưa thực sự được bao quát và khai thác hết một cách chuyên sâu các khía cạnh của ngành Bán lẻ Trực tuyến.

Qua Phân tích RFM bằng phân cụm K-means sử dụng công cụ Pyspark, nhóm hướng đến mục tiêu học hỏi về BigData và thực hành trên các công cụ phát triển. Từ đó, phân tích RFM sâu hơn với những phương pháp và thuật toán khác. Ngoài ra, nhóm cũng hướng đến phân tích những dữ liệu thực tế khác trong tương lai. Tuy nhiên với kết quả phân cụm dựa trên Apache Spark mà cụ thể là Pyspark, doanh nghiệp hoặc người quản lý có kiến thức nền tảng về kinh tế cần xác thực lại tính đúng đắn của kết quả dưới góc nhìn kinh doanh sao cho phù hợp. Cần kết hợp nhiều phương pháp khác nhau, đa dạng mô hình phân tích để bám sát thực tế giúp đưa ra quyết định tối ưu nhất, xây dựng những chiến lược kinh doanh phù hợp. Từ dữ liệu về phân khúc khách hàng và kết hợp với các nghiên cứu khác trong tương lai có thể xây dựng các chiến lược phân khúc cụ thể hơn, bám sát thực tế doanh nghiệp hơn, đưa ra các chiến lược chăm sóc khách hàng riêng cho từng nhóm cũng như nguồn dữ liệu cho doanh nghiệp.

Tài liệu tham khảo

- [1] Phan Châu Minh Trường (2022), Phân tích hành vi khách hàng trong lĩnh vực bán lẻ với mô hình RFM tiếp cận bằng kỹ thuật học máy không giám sát
- [2] Mai Thị Kim Ngân (2020). Ứng dụng khai thác dữ liệu vào hệ thống quản trị quan hệ khách hàng.
- [3]. <https://aws.amazon.com/free/analytics/>
- [4]. Phuc Ngoc Nghia (2020), Tìm hiểu về Apache Spark, từ <https://viblo.asia/p/tim-hieu-ve-apache-spark-ByEZkQQW5Q0>
- [5]. Bài 4: K-means Clustering, từ <https://machinelearningcoban.com/2017/01/01/kmeans/>


PHẦN 2: PHỤ LỤC

```
#Set up spark context and SparkSession
import pyspark
from pyspark.sql import SparkSession

spark = SparkSession.builder.master("local").appName("demo1").getOrCreate()

23/01/15 15:09:06 WARN Utils: Your hostname, n1 resolves to a loopback address: 127.0.1.1; using 10.0.2.15 instead (on interface en
23/01/15 15:09:06 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
23/01/15 15:09:10 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where
```

spark

 **SparkSession - in-memory**

SparkContext

[Spark UI](#)

Version
v3.3.1

Master
local

AppName
demo1

```
#Load dataset
df_raw = spark.read.csv("dataRFM_new.csv", header=True)
```

```
#Check dataset
df_raw.show()
```

InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
536365	85123A	WHITE HANGING HEA...	6	12/1/2010 8:26	2.55	17850	United Kingdom
536365	71053	WHITE METAL LANTERN	6	12/1/2010 8:26	3.39	17850	United Kingdom
536365	84406B	CREAM CUPID HEART...	8	12/1/2010 8:26	2.75	17850	United Kingdom
536365	84029G	KNITTED UNION FLA...	6	12/1/2010 8:26	3.39	17850	United Kingdom
536365	84029E	RED WOOLLY HOTTIE...	6	12/1/2010 8:26	3.39	17850	United Kingdom
536365	22752	SET 7 BABUSHKA NE...	2	12/1/2010 8:26	7.65	17850	United Kingdom
536365	21730	GLASS STAR FROSTE...	6	12/1/2010 8:26	4.25	17850	United Kingdom
536366	22633	HAND WARMER UNION...	6	12/1/2010 8:28	1.85	17850	United Kingdom
536366	22632	HAND WARMER RED P...	6	12/1/2010 8:28	1.85	17850	United Kingdom
536367	84879	ASSORTED COLOUR B...	32	12/1/2010 8:34	1.69	13047	United Kingdom
536367	22745	POPPY'S PLAYHOUSE...	6	12/1/2010 8:34	2.1	13047	United Kingdom
536367	22748	POPPY'S PLAYHOUSE...	6	12/1/2010 8:34	2.1	13047	United Kingdom
536367	22749	FELTCRAFT PRINCES...	8	12/1/2010 8:34	3.75	13047	United Kingdom
536367	22310	IVORY KNITTED MUG...	6	12/1/2010 8:34	1.65	13047	United Kingdom
536367	84969	BOX OF 6 ASSORTED...	6	12/1/2010 8:34	4.25	13047	United Kingdom
536367	22623	BOX OF VINTAGE JI...	3	12/1/2010 8:34	4.95	13047	United Kingdom
536367	22622	BOX OF VINTAGE AL...	2	12/1/2010 8:34	9.95	13047	United Kingdom
536367	21754	HOME BUILDING BLO...	3	12/1/2010 8:34	5.95	13047	United Kingdom
536367	21755	LOVE BUILDING BLO...	3	12/1/2010 8:34	5.95	13047	United Kingdom
536367	21777	RECIPE BOX WITH M...	4	12/1/2010 8:34	7.95	13047	United Kingdom

only showing top 20 rows

```
#Check dataset xem cac cot cua bang
df_raw.printSchema()
```

```
root
|-- InvoiceNo: string (nullable = true)
|-- StockCode: string (nullable = true)
|-- Description: string (nullable = true)
|-- Quantity: string (nullable = true)
|-- InvoiceDate: string (nullable = true)
|-- UnitPrice: string (nullable = true)
|-- CustomerID: string (nullable = true)
|-- Country: string (nullable = true)
```

```
#Data clean and data manipulation
#Check and remove the null values
```

```
from pyspark.sql.functions import count
```

```
def my_count(df_in):
    df_in.agg( *[ count(c).alias(c) for c in df_in.columns ] ).show()
```

```
my_count(df_raw)
```

```
+-----+-----+-----+-----+-----+-----+-----+
|InvoiceNo|StockCode|Description|Quantity|InvoiceDate|UnitPrice|CustomerID|Country|
+-----+-----+-----+-----+-----+-----+-----+
| 397924| 397924| 397924| 397924| 397924| 397924| 397924| 397924|
+-----+-----+-----+-----+-----+-----+-----+
```

```
df = df_raw.dropna(how='any')
```

```
my_count(df)
```

```
+-----+-----+-----+-----+-----+-----+-----+
|InvoiceNo|StockCode|Description|Quantity|InvoiceDate|UnitPrice|CustomerID|Country|
+-----+-----+-----+-----+-----+-----+-----+
| 397924| 397924| 397924| 397924| 397924| 397924| 397924| 397924|
+-----+-----+-----+-----+-----+-----+-----+
```

```
# Dealwith the InvoiceDate
```

```
spark.sql("set spark.sql.legacy.timeParserPolicy=LEGACY")
```

```
from pyspark.sql.functions import to_utc_timestamp, unix_timestamp, lit, datediff, col
```

```
timeFmt = "MM/dd/yy HH:mm"
```

```
df = df.withColumn('NewInvoiceDate'
                  , to_utc_timestamp(unix_timestamp(col('InvoiceDate'),timeFmt).cast('timestamp')
                  , 'UTC'))
```

```
df.show(5)
```

```
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|InvoiceNo|StockCode|Description|Quantity|InvoiceDate|UnitPrice|CustomerID|Country|NewInvoiceDate|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 536365| 85123A|WHITE HANGING HEA...|6|12/1/2010 8:26|2.55|17850|United Kingdom|2010-12-01 08:26:00|
| 536365| 71053| WHITE METAL LANTERN|6|12/1/2010 8:26|3.39|17850|United Kingdom|2010-12-01 08:26:00|
| 536365| 84406B|CREAM CUPID HEART...|8|12/1/2010 8:26|2.75|17850|United Kingdom|2010-12-01 08:26:00|
| 536365| 84029G|KNITTED UNION FLA...|6|12/1/2010 8:26|3.39|17850|United Kingdom|2010-12-01 08:26:00|
| 536365| 84029E|RED WOOLLY HOTTIE...|6|12/1/2010 8:26|3.39|17850|United Kingdom|2010-12-01 08:26:00|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
```

only showing top 5 rows

```
#Calculate Total Price
```

```
from pyspark.sql.functions import round
```

```
df = df.withColumn('TotalPrice', round( df.Quantity * df.UnitPrice, 2 ) )
```

```
#Calculate time difference
```

```
from pyspark.sql.functions import mean, min, max, sum, datediff, to_date
```

```
spark.sql("set spark.sql.legacy.timeParserPolicy=LEGACY")
```

```
date_max = df.select(max('NewInvoiceDate')).toPandas()
```

```
current = to_utc_timestamp( unix_timestamp(lit(str(date_max.iloc[0][0])), \
        'yy-MM-dd HH:mm').cast('timestamp'), 'UTC' )
```

```
# Calculatre Duration
```

```
df = df.withColumn('Duration', datediff(lit(current), 'NewInvoiceDate'))
```

```
/usr/local/lib/python3.10/dist-packages/pyspark/sql/pandas/conversion.py:248: FutureWarning: Passing unit-less datetime64 dtype to
series = series.astype(t, copy=False)
```

```
recency = df.groupBy('CustomerID').agg(min('Duration').alias('Recency'))
```

```
frequency = df.groupBy('CustomerID', 'InvoiceNo').count()\
```

```
    .groupBy('CustomerID')\
```

```
    .agg(count("*").alias("Frequency"))
```

```
monetary = df.groupBy('CustomerID').agg(round(sum('TotalPrice'), 2).alias('Monetary'))
```

```
rfm = recency.join(frequency, 'CustomerID', how = 'inner')\
```

```
    .join(monetary, 'CustomerID', how = 'inner')
```

```
rfm.show(5)
```

```
+-----+-----+-----+-----+
|CustomerID|Recency|Frequency|Monetary|
+-----+-----+-----+-----+
| 15555| 12| 16| 4805.17|
| 15574| 177| 4| 702.25|
| 15634| 17| 1| 243.55|
| 13610| 12| 7| 1131.88|
| 13192| 95| 2| 911.94|
+-----+-----+-----+-----+
only showing top 5 rows
```

```
rfm.describe().show()
```

```
+-----+-----+-----+-----+
|summary| CustomerID| Recency| Frequency| Monetary|
+-----+-----+-----+-----+
| count| 4339| 4339| 4339| 4339|
| mean| 15299.936851809172| 92.0414842129523| 4.271952062687255| 2053.7930168241505|
| stddev| 1721.8897579594227| 100.00775734416375| 7.70549277131483| 8988.248381460095|
| min| 12346| 0| 1| 0.0|
| max| 18287| 373| 210| 280206.02|
+-----+-----+-----+-----+
```

```
from pyspark.ml.clustering import KMeans
from pyspark.ml.feature import VectorAssembler
from pyspark.ml.feature import StandardScaler
from pyspark.ml.evaluation import ClusteringEvaluator
```

```
features = rfm.columns[1:]
```

```
# vectorize all features
assembler = VectorAssembler(inputCols=features, outputCol="rfm_features")
assembled_data = assembler.transform(rfm)
assembled_data = assembled_data.select('CustomerID', 'rfm_features')
assembled_data.show(5)
```

```
+-----+-----+
|CustomerID| rfm_features|
+-----+-----+
| 15555| [12.0,16.0,4805.17]|
| 15574| [177.0,4.0,702.25]|
| 15634| [17.0,1.0,243.55]|
| 13610| [12.0,7.0,1131.88]|
| 13192| [95.0,2.0,911.94]|
+-----+-----+
only showing top 5 rows
```

```
# Standardization
scaler = StandardScaler(inputCol='rfm_features', outputCol='rfm_standardized')
data_scale = scaler.fit(assembled_data)
scaled_data = data_scale.transform(assembled_data)
scaled_data.show(5)
```

```
+-----+-----+-----+
|CustomerID| rfm_features| rfm_standardized|
+-----+-----+-----+
| 15555| [12.0,16.0,4805.17]| [0.11999069190906...|
| 15574| [177.0,4.0,702.25]| [1.76986270565869...|
| 15634| [17.0,1.0,243.55]| [0.16998681353784...|
| 13610| [12.0,7.0,1131.88]| [0.11999069190906...|
| 13192| [95.0,2.0,911.94]| [0.94992631094675...|
+-----+-----+-----+
only showing top 5 rows
```

```
k_means = KMeans(featuresCol='rfm_standardized', k=3)
model = k_means.fit(scaled_data)
predictions = model.transform(scaled_data)

result = predictions.select('CustomerID', 'prediction')
```

```
result.show(5)
```

```
+-----+-----+
|CustomerID|prediction|
+-----+-----+
|    15555|         0|
|    15574|         1|
|    15634|         0|
|    13610|         0|
|    13192|         0|
+-----+-----+
only showing top 5 rows
```

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
```

```
centers = model.clusterCenters()
print ("cluster center ")
for center in centers:
    print(center)
```

```
cluster center
[0.40564571 0.62954553 0.22372967]
[2.45703213 0.20495575 0.07011877]
[ 0.0635665  10.4285534  13.67211963]
```

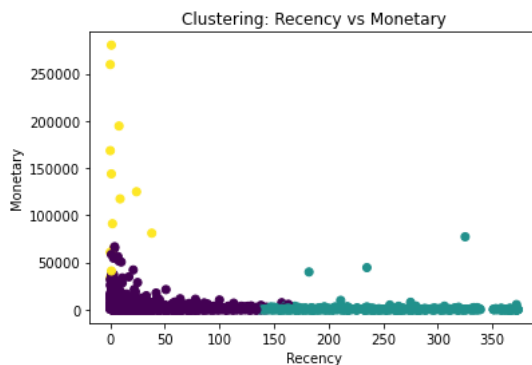
```
result_new = rfm.join(predictions.select('CustomerID','prediction'),'CustomerID',how='left')
result_new.show(5)
```

```
+-----+-----+-----+-----+-----+
|CustomerID|Recency|Frequency|Monetary|prediction|
+-----+-----+-----+-----+-----+
|    13192|     95|         2|  911.94|         0|
|    13610|     12|         7| 1131.88|         0|
|    14157|     19|         2|  432.88|         0|
|    15555|     12|        16| 4805.17|         0|
|    15574|    177|         4|  702.25|         1|
+-----+-----+-----+-----+-----+
only showing top 5 rows
```

```
rfm_visual = result_new.toPandas().set_index('CustomerID')
rfm_visual.head()
```

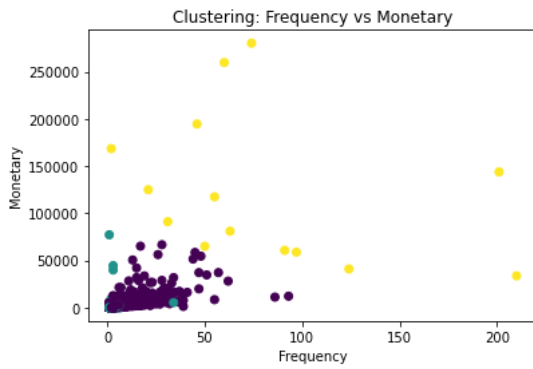
	Recency	Frequency	Monetary	prediction
CustomerID				
12346	325	1	77183.60	1
12347	2	7	4310.00	0
12348	75	4	1797.24	0
12349	18	1	1757.55	0
12350	310	1	334.40	1

```
plt.scatter(rfm_visual.Recency, rfm_visual.Monetary, c=rfm_visual.prediction)
plt.title('Clustering: Recency vs Monetary')
plt.xlabel('Recency')
plt.ylabel('Monetary')
plt.show()
```

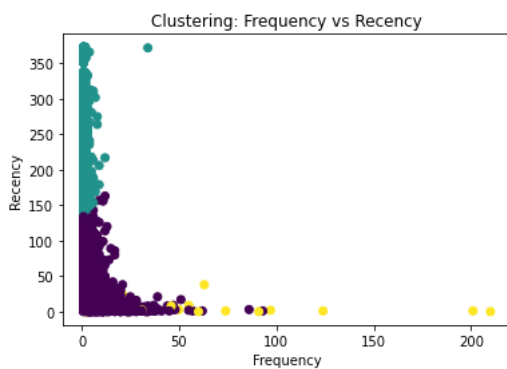


```
plt.scatter(rfm_visual.Frequency, rfm_visual.Monetary, c=rfm_visual.prediction)
plt.title('Clustering: Frequency vs Monetary')
```

```
plt.xlabel('Frequency')
plt.ylabel('Monetary')
plt.show()
```



```
plt.scatter(rfm_visual.Frequency, rfm_visual.Recency, c=rfm_visual.prediction)
plt.title('Clustering: Frequency vs Recency')
plt.xlabel('Frequency')
plt.ylabel('Recency')
plt.show()
```



```
threedee = plt.figure(figsize=(12,10)).gca(projection='3d')
threedee.scatter(rfm_visual.Recency, rfm_visual.Frequency, rfm_visual.Monetary, c=rfm_visual.prediction)
threedee.set_xlabel('Recency')
threedee.set_ylabel('Frequency')
threedee.set_zlabel('Monetary')
plt.show()
```

/tmp/ipykernel_15283/3370210349.py:1: MatplotlibDeprecationWarning: Calling gca() with
threedee = plt.figure(figsize=(12,10)).gca(projection='3d')

