

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA KHOA HỌC VÀ KỸ THUẬT THÔNG TIN

LÊ HOÀNG NGUYỄN
ĐẶNG NGUYỄN BÌNH AN

ĐỒ ÁN CUỐI KỲ
MÔN: HỌC MÁY THỐNG KÊ
LỚP: DS102.L21

PHÂN KHÚC KHÁCH HÀNG BẰNG MACHINE
LEARNING

CUSTOMER SEGMENTATION WITH MACHINE
LEARNING

SINH VIÊN KHOA HỌC VÀ KỸ THUẬT THÔNG TIN

TP. HỒ CHÍ MINH, 2021

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA KHOA HỌC VÀ KỸ THUẬT THÔNG TIN

LÊ HOÀNG NGUYỄN - 19520182

ĐẶNG NGUYỄN BÌNH AN - 19521170

ĐỒ ÁN CUỐI KỲ

MÔN: HỌC MÁY THỐNG KÊ

LỚP: DS102.L21

**PHÂN KHÚC KHÁCH HÀNG BẰNG MACHINE
LEARNING**

**CUSTOMER SEGMENTATION WITH MACHINE
LEARNING**

SINH VIÊN KHOA HỌC VÀ KỸ THUẬT THÔNG TIN

GIẢNG VIÊN HƯỚNG DẪN

TS. VÕ TẤN TRẦN MINH KHANG

ThS. VÕ DUY NGUYỄN

TP. HỒ CHÍ MINH, 2021

LỜI CẢM ƠN

Lời đầu tiên nhóm chúng em xin trân trọng cảm ơn Trường đại học Công nghệ Thông tin, Khoa Khoa học và kỹ thuật thông tin đã tạo điều kiện cho chúng em được học môn “Học máy thống kê”, môn học cơ sở ngành Khoa học dữ liệu sau các môn đại cương. Qua quá trình học, chúng em không chỉ đơn thuần là học được lý thuyết mà hiểu được cơ bản một số kiến thức trong những lĩnh vực liên quan qua những bài vô cùng thiết thực. Tiếp theo em xin chân thành cảm ơn thầy Nguyễn Tấn Trần Minh Khang, thầy Võ Duy Nguyên và anh Hồ Thái Ngọc đã truyền đạt kiến thức, dẫn dắt chúng em trong môn học này, giúp chúng em tiếp cận, nắm được kiến thức cơ bản và có cái nhìn bao quát về “Học máy”.

Trong đề tài lần này nhóm chúng em đã cố gắng hết mình nghiên cứu, tìm hiểu và hoàn thành tốt nhất có thể theo khả năng để có được một báo cáo hoàn chỉnh. Tuy nhiên do kiến thức và kỹ năng còn nông nên không tránh khỏi sai sót. Rất mong các thầy có thể góp ý, đánh giá để chúng em ngày càng tiến bộ hơn.

Nhóm chúng em trân trọng cảm ơn!

This image shows a single page of white paper designed for handwriting practice. It features ten evenly spaced, horizontal dotted lines that run across the entire width of the page. These lines are intended to guide the placement of letters, typically serving as the top line for capital letters and the middle line for lowercase letters. The background is plain white, and there are no other markings or text on the page.

MỤC LỤC

TÓM TẮT BÁO CÁO	1
Chương 1. TỔNG QUAN	2
1.1. Phân khúc khách hàng	2
1.1.1. Khái niệm	2
1.1.2. Tầm quan trọng.....	2
1.1.3. Lợi ích của việc phân khúc khách hàng	3
1.2. Phân khúc khách hàng và machine learning.....	3
1.3. Mục tiêu đề tài	4
Chương 2. CÁC CÔNG TRÌNH LIÊN QUAN	5
2.1. Maximizing Strategy Improvement In Mall Customer Segmentation Using K-Means Clustering	5
2.2. Implement Of K-Means++ Algorithm For Store Customers Segmentation Using Neo4j	5
2.3. Machine Learning Based Classification And Segmentation Techniques For Crm: A Customer Analytics	7
Chương 3. CƠ SỞ LÝ THUYẾT	9
3.1. Khoảng cách Euclide	9
3.2. K-Means Clustering.....	9
3.2.1. Định nghĩa	9
3.2.2. Phân tích toán học	10
3.2.3. Thuật toán	12
3.3. Phương pháp Elbow.....	13
3.3.1. Định nghĩa	13
3.3.2. Thực hiện	14
3.4. Phương pháp Silhouette	14

3.4.1.	Định nghĩa	14
3.4.2.	Tính hệ số Silhouette	14
3.4.3.	Thực hiện	15
3.5.	Gaussian Naïve Bayes	15
3.5.1.	Định lý xác suất bayes	15
3.5.2.	Naïve Bayes	16
3.6.	Các độ đo	17
3.6.1.	Accuracy	18
3.6.2.	Precision	18
3.6.3.	Recall	18
3.6.4.	F1-score	19
Chương 4.	BỘ DỮ LIỆU	21
4.1.	Bộ dữ liệu	21
4.2.	Tiền xử lý	21
Chương 5.	XÂY DỰNG MÔ HÌNH	24
5.1.	Xây dựng mô hình	24
5.1.1.	Mô hình 1	24
5.1.2.	Mô hình 2	38
Chương 6.	HIỆU SUẤT MÔ HÌNH	47
6.1.	Mô hình 1	47
6.2.	Mô hình 2	49
Chương 7.	KẾT LUẬN	50
TÀI LIỆU THAM KHẢO		52
PHỤ LỤC		56

DANH MỤC HÌNH

<i>Hình 0.1</i>	<i>Mô tả cách hoạt động của mô hình trong báo cáo</i>	<i>1</i>
<i>Hình 1.1</i>	<i>Phân khúc khách hàng</i>	<i>2</i>
<i>Hình 1.2</i>	<i>Khách hàng tiềm năng</i>	<i>3</i>
<i>Hình 2.1</i>	<i>Trực quan kết quả phân khúc các cụm của nghiên cứu 2.1</i>	<i>5</i>
<i>Hình 2.2</i>	<i>Trực quan kết quả các cụm của nghiên cứu 2.2</i>	<i>6</i>
<i>Hình 3.1</i>	<i>Minh họa thuật toán K-Means</i>	<i>9</i>
<i>Hình 3.2</i>	<i>Sơ đồ mô tả 5 bước của thuật toán K-Means</i>	<i>13</i>
<i>Hình 3.3</i>	<i>Minh họa phương pháp Elbow</i>	<i>13</i>
<i>Hình 4.1</i>	<i>Code load dataset</i>	<i>22</i>
<i>Hình 4.2</i>	<i>Mô tả tập dữ liệu ở giai đoạn tiền xử lý bước 1</i>	<i>22</i>
<i>Hình 4.3</i>	<i>Code tiền xử lý dữ liệu cho mô hình 1</i>	<i>23</i>
<i>Hình 4.4</i>	<i>Code tiền xử lý cho mô hình 2</i>	<i>23</i>
<i>Hình 5.1</i>	<i>Dữ liệu tập dữ liệu trước khi đưa vào phân nhóm ở mô hình 1</i>	<i>24</i>
<i>Hình 5.2</i>	<i>Code thêm các thư viện, module cho quá trình phân nhóm của mô hình 1</i>	<i>25</i>
<i>Hình 5.3</i>	<i>Code tìm K cho thuật toán K-Means trong mô hình 1</i>	<i>25</i>
<i>Hình 5.4</i>	<i>Trực quan hóa silhouette của mô hình 1</i>	<i>26</i>
<i>Hình 5.5</i>	<i>Code huấn luyện mô hình phân nhóm bằng K-Means trong mô hình 1</i>	<i>26</i>
<i>Hình 5.6</i>	<i>Code trực quan hóa 12 cụm sau khi phân nhóm của mô hình 1</i>	<i>27</i>
<i>Hình 5.7</i>	<i>Trực quan hóa 12 cụm sau phân nhóm của mô hình 1</i>	<i>27</i>
<i>Hình 5.8</i>	<i>Code trực quan hóa sự phân bố 200 điểm vào 12 cụm trong mô hình 1</i>	<i>27</i>
<i>Hình 5.9</i>	<i>Sự phân bố của 200 điểm dữ liệu vào 12 cụm theo tỉ lệ % của mô hình 1</i>	<i>28</i>
<i>Hình 5.10</i>	<i>Code thêm các thư viện, module cho quá trình phân lớp ở mô hình 1</i>	<i>33</i>
<i>Hình 5.11</i>	<i>Tập dữ liệu đã xử lý tại Bước 7 quá trình xây dựng mô hình 1</i>	<i>34</i>
<i>Hình 5.12</i>	<i>Code chia dữ liệu thành tập train, test của mô hình 1</i>	<i>34</i>
<i>Hình 5.13</i>	<i>Code huấn luyện mô hình phân lớp bằng GaussianNB trong mô hình 1</i>	<i>35</i>
<i>Hình 5.14</i>	<i>Code in ra confusion matrix trong mô hình 1</i>	<i>35</i>
<i>Hình 5.15</i>	<i>Confusion matrix của mô hình 1</i>	<i>35</i>
<i>Hình 5.16</i>	<i>Code in ra classification report cho mô hình 1</i>	<i>36</i>
<i>Hình 5.17</i>	<i>Classification report của mô hình 1</i>	<i>36</i>

Hình 5.18	Code tạo button và form cho mô hình 1	37
Hình 5.19	Giao diện dự đoán một khách hàng của mô hình 1	37
Hình 5.20	Code thêm các thư viện, module cho quá trình phân nhóm ở mô hình 2 ...	38
Hình 5.21	Code tìm K cho thuật toán K-Means trong mô hình 2.....	38
Hình 5.22	Trực quan hóa silhouette score cho mô hình 2.....	39
Hình 5.23	Code huấn luyện mô hình phân nhóm bằng K-Means cho mô hình 2.....	39
Hình 5.24	Code trực quan hóa 5 cụm sau khi phân nhóm của mô hình 2	40
Hình 5.25	Trực quan hóa dữ liệu trong 5 cụm được phân của mô hình 2	40
Hình 5.26	Code sự phân bố 200 điểm dữ liệu vào 5 cụm của mô hình 2.....	41
Hình 5.27	Sự phân bố 200 điểm dữ liệu vào 5 cụm của mô hình 2.....	41
Hình 5.28	Code thêm các thư viện, module của mô hình 2	43
Hình 5.29	Tập dữ tại bước 7 của mô hình 2.....	43
Hình 5.30	Code chia tập dữ liệu thành train, test cho mô hình 2	44
Hình 5.31	Code huấn luyện mô hình phân lớp bằng GaussianNB cho mô hình 2.....	44
Hình 5.32	Code in ra confusion matrix cho mô hình 2	44
Hình 5.33	Confusion matrix của mô hình 2.....	45
Hình 5.34	Code in ra classification report cho mô hình 2	45
Hình 5.35	Classification report của mô hình 2	45
Hình 5.36	Code tạo button và form để dự đoán khách hàng cho mô hình 2.....	46
Hình 5.37	Giao diện dự đoán một khách hàng của mô hình 2.....	46
Hình 6.1	Confusion matrix cho mô hình 1.....	47
Hình 6.2	Confusion matrix của mô hình 2.....	49

DANH MỤC BẢNG

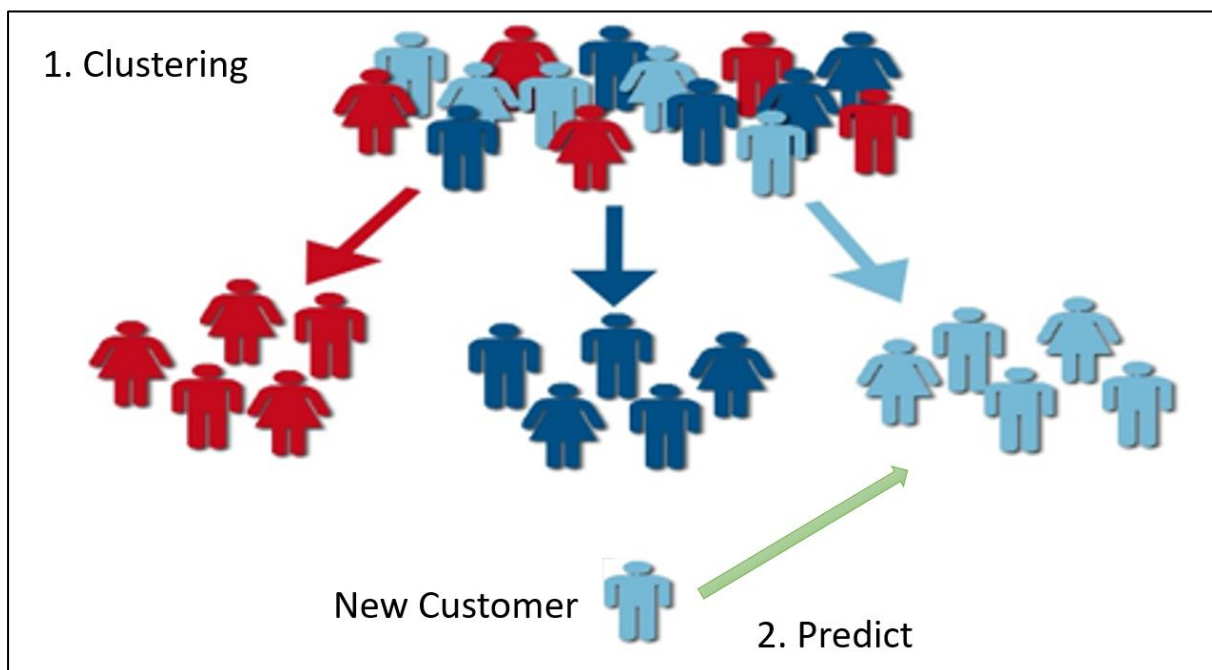
Bảng 2.1 Thống kê dữ liệu trong 5 cụm của 2 thuật toán <i>K-Means</i> và <i>K-Means++</i> của nghiên cứu 2.2	7
Bảng 2.2 Confusion matrix của thuật toán <i>MLP</i> trong nghiên cứu 2.3.....	8
Bảng 3.1 Bảng biến thiên hàm $f(m_k)$	12
Bảng 3.2 Confusion matrix của một mô hình phân lớp	18
Bảng 3.3 Confusion matrix tự tạo của một mô hình với bộ dữ liệu mất cân bằng	19
Bảng 4.1 Mô tả một số điểm dữ liệu trong tập dữ liệu	21
Bảng 5.1 Mô tả 12 cụm sau khi phân chia của mô hình 1	32
Bảng 5.2 Thông kê các điểm trong 12 cụm sau khi phân chia của mô hình 1	33
Bảng 5.3 Mô tả 5 cụm sau khi phân chia của mô hình 2	42
Bảng 5.4 Thống kê số điểm dữ liệu trong 5 cụm đã phân chia của mô hình 2.....	42
Bảng 6.1 Classification report mô hình 1	48
Bảng 6.2 Classification report mô hình 2	49

DANH MỤC TỪ VIẾT TẮT

STT	Từ viết tắt	Ý nghĩa
1	MLP	Multi-layer perceptron
2	TTTM	Trung tâm thương mại

TÓM TẮT BÁO CÁO

Ngày nay, machine learning đã không còn quá xa lạ với mọi người, nó đã xuất hiện trong rất nhiều lĩnh vực của đời sống như tài chính, thương mại điện tử, mạng xã hội,... và cái chúng tôi đề cập ở đây là trong kinh doanh, người ta đã ứng dụng machine learning vào để phân khúc khách hàng. Đây là một cách vô cùng hữu ích để giảm thiểu công sức của con người đi đáng kể. Trong báo cáo này, chúng tôi sẽ trình bày cách xây dựng 2 mô hình. Trong mô hình 1 chúng tôi xây dựng mô hình phân nhóm khách hàng dựa trên các thuộc tính nhân khẩu học như: giới tính (Gender), tuổi (Age), thu nhập năm trước đó (Annual Income (k\$)) và điểm chi tiêu (Spending score (1-100)) được trung tâm thương mại tính cho mỗi khách hàng, sau đó dự đoán một khách hàng mới sẽ thuộc nhóm nào dựa trên các thuộc tính này. Còn ở mô hình 2, chúng tôi xây dựng dựa trên 2 thuộc tính thực dụng là thu nhập năm trước đó (Annual Income (k\$)) và điểm chi tiêu (Spending score (1-100)) được trung tâm thương mại tính cho mỗi khách hàng.



Hình 0.1 Mô tả cách hoạt động của mô hình trong báo cáo

Chương 1. TỔNG QUAN

1.1. Phân khúc khách hàng

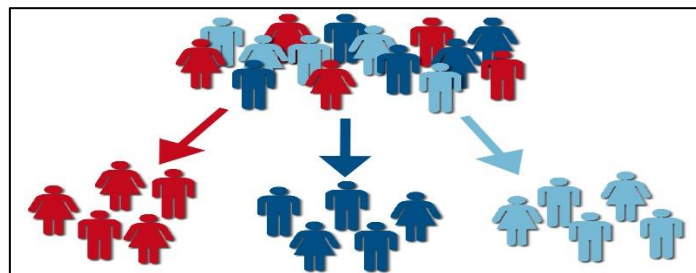
1.1.1. Khái niệm

Phân khúc khách hàng là hoạt động của doanh nghiệp nhằm phân chia khách hàng thành những nhóm có những đặc tính giống nhau để thuận tiện triển khai các chiến lược kinh doanh.

1.1.2. Tầm quan trọng

Phân khúc khách hàng là một công việc cực kỳ quan trọng trong kinh doanh. Từ những thông tin thu thập được như thói quen, sở thích mua hàng, đến khả năng chi tiêu người ta xây dựng chiến lược kinh doanh nhằm tạo ra sức mạnh cạnh tranh và lợi thế lâu dài.

Những năm gần đây, Trung tâm thương mại (TTTM) không còn quá xa lạ với mọi người, nó hầu như xuất hiện ở khắp mọi nơi. Sự phát triển kinh tế khiến các thành phố trở nên chật chội, không gian vui chơi bị bó hẹp TTTM dần trở thành địa điểm thường xuyên lui tới của nhiều gia đình. Bên cạnh đó, TTTM đang trở thành xu hướng thay thế dần cho hình thức kinh doanh truyền thống trở thành kênh bán lẻ hiện đại, bởi nó mang lại nhiều lợi ích cho cộng đồng khi nó đáp ứng hầu hết các nhu cầu từ cơ bản đến cao cấp, mang lại cuộc sống tiện nghi, thuận lợi hơn cho mọi người. Nhưng có “cầu” thì mới có “cung” – TTTM muốn hoạt động thì phải có khách hàng, vậy câu hỏi đặt ra là khách hàng đến TTTM có những mong đợi gì? Có những kiểu khách hàng như thế nào đến TTTM? Vì vậy, nếu muốn thúc đẩy người tiêu dùng đến tham quan và mua sắm thì phải phân khúc họ dựa trên một số đặc điểm để có cái nhìn sâu sắc, hữu ích hơn cho nhà quản lý TTTM.



Hình 1.1 Phân khúc khách hàng

1.1.3. Lợi ích của việc phân khúc khách hàng

- Tránh lãng phí nguồn lực theo đuổi những chính sách sai lầm.
- Quản trị khách hàng, marketing hiệu quả hơn theo từng nhóm phân khúc như địa lý, nhân khẩu học, hành vi, tâm lý. Ngày nay marketing trên website và email rất phổ biến, dựa vào từng phân khúc ta có thể có những chiến dịch, nội dung marketing phù hợp, hấp dẫn hơn.
- Phát huy “lòng trung thành” của khách hàng thân thiết để có thể giữ chân họ với những ưu đãi, chính sách.
- Cải thiện chất lượng sản phẩm, phục vụ.
- Đặc biệt là thu hút những khách hàng tiềm năng.



Hình 1.2 Khách hàng tiềm năng

1.2. Phân khúc khách hàng và machine learning

Phân khúc khách hàng trước đây là một công việc đầy thách thức và tốn thời gian để xem xét một cách thủ công tập dữ liệu được thu thập từ TTTM và truy vấn chúng với hy vọng tìm ra điểm tương đồng để nhóm các khách hàng lại với nhau. Nhưng những năm gần đây, điều đó đã trở nên dễ dàng hơn nhờ vào machine learning, các thuật toán machine learning giúp tìm ra các quy tắc thống kê dữ liệu một cách nhanh chóng và tối ưu hóa lợi nhuận. Nếu không dùng machine learning thì mỗi khi thói quen, hành vi, sở thích của khách hàng thay đổi thì ta lại phải thủ công thực hiện phân khúc khách hàng dẫn đến lãng phí nguồn lực và công sức mà không tránh khỏi sai sót. Mô hình máy học có thể xử lý dữ liệu khách hàng và khám phá ra các dữ liệu giống nhau theo một quy tắc nào đó trên tập dữ liệu. Trong một số trường hợp, các thuật toán học

máy “vô tình” tìm ra những phân khúc khách hàng mà rất khó có thể tìm ra bằng trực giác, phương pháp thử công bình thường.

1.3. Mục tiêu đề tài

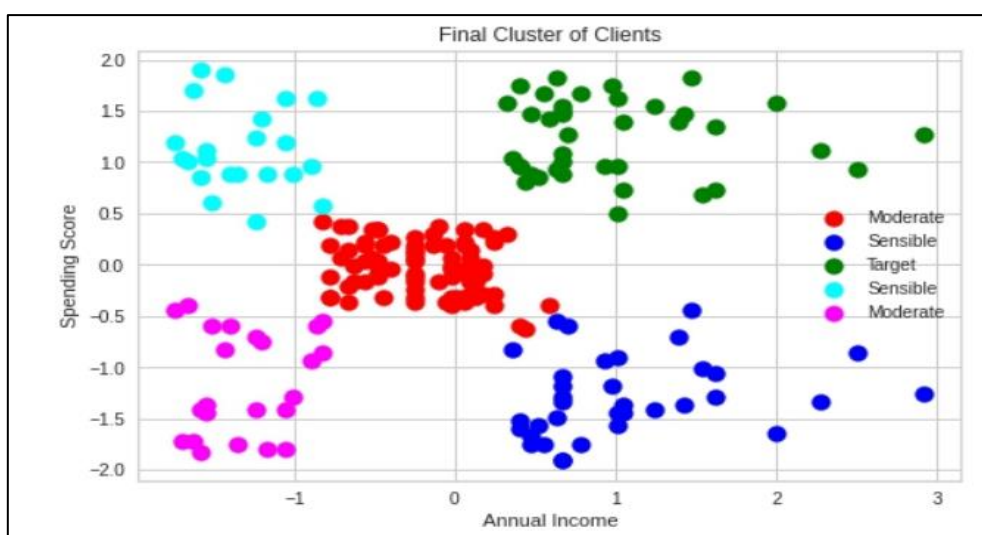
Từ tập dữ liệu khách hàng ở trung tâm thương mại với các đặc điểm của khách hàng như Giới tính (Gender), Tuổi (Age), Thu nhập hằng năm (Annual Income (k\$)) và Điểm đánh giá chi tiêu của TTTM dành cho khách hàng (Spending score (1-100)) kết hợp với các thuật toán machine learning chúng tôi xây dựng 2 mô hình:

- Mô hình 1: chúng tôi sẽ xây dựng mô hình phân nhóm khách hàng với thuật toán K-Means, sau đó dùng Naïve Bayes xây dựng mô hình dự đoán nhóm khách hàng các thuộc tính nhân khẩu học như: Gender, Age, Annual Income (k\$), Spending Score (1-100).
- Mô hình 2: chúng tôi sẽ xây dựng một mô hình phân nhóm khách hàng với thuật toán K-Means, sau đó dùng Naïve Bayes xây dựng mô hình dự đoán nhóm khách hàng dựa trên 2 thuộc tính của tập dữ liệu: Annual Income (k\$), Spending Score (1-100).

Chương 2. CÁC CÔNG TRÌNH LIÊN QUAN

2.1. Maximizing Strategy Improvement In Mall Customer Segmentation Using K-Means Clustering[14]

Ứng dụng phân khúc khách hàng thì đã rất phổ biến trên toàn thế giới trong lĩnh vực marketing, nó dùng để xác định chiến lược, biết được mong muốn của khách hàng, nếu không doanh nghiệp sẽ lãng phí nguồn lực và chạy theo những chính sách sai lầm. Vì vậy, nghiên cứu, xây dựng mô hình phân khúc khách hàng được thực hiện. Nghiên cứu được thực hiện bởi ông Musthofa Galih Pradana thuộc Khoa Khoa học thông tin tại trường Đại học Alma Ata tại Indonesia và Hoang Thi Ha thuộc Khoa Hệ thống thông tin quản lý trường Đại học Đà Nẵng. Bộ dữ liệu được sử dụng bao gồm các thuộc tính: CustomerID, Age, Gender, Annual Income và Spending Score. Trong nghiên cứu, hai người họ sử dụng thuật toán K-Means Clustering và phương pháp Elbow để chọn K (1-10) đã thành công chia khách hàng thành 5 ($K = 5$) nhóm dựa trên mối quan hệ giữa 2 thuộc tính Thu nhập hàng năm (Annual Income) và Điểm đánh giá chi tiêu khách hàng (Spending score) trong đó có nhóm khách hàng có thu nhập và điểm chi tiêu cao rất phù hợp cho triển khai chiến lược kinh doanh.

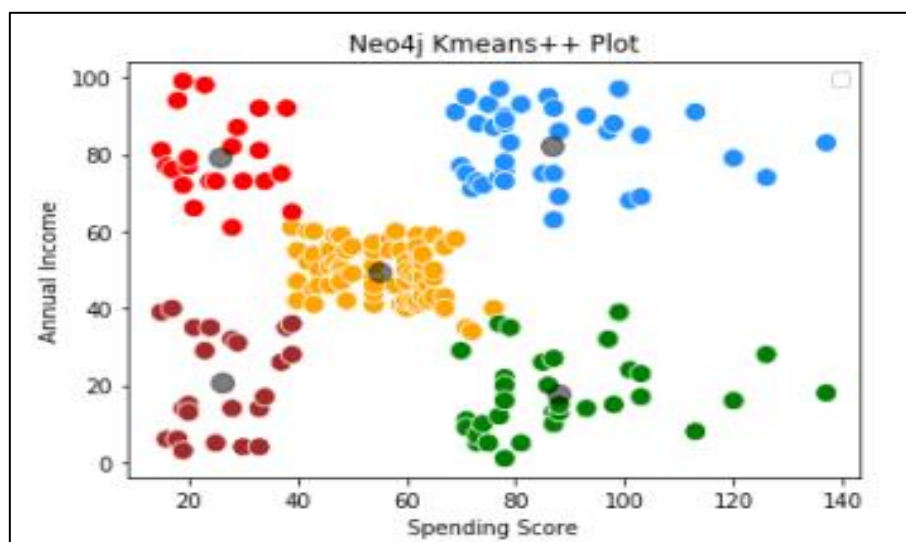


Hình 2.1 Trực quan kết quả phân khúc các cụm của nghiên cứu 2.1[14]

2.2. Implement Of K-Means++ Algorithm For Store Customers Segmentation Using Neo4j[3]

Trong kỷ nguyên dữ liệu và thông tin như hiện nay, dữ liệu đã trở thành một trong những thứ cực kỳ hữu dụng. Dữ liệu có thể trở thành thông tin hữu ích nếu được

xử lý phù hợp. Một trong các ứng dụng của xử lý dữ liệu để có được thông tin hữu ích là thực hiện phân khúc khách hàng từ tập dữ liệu có được. Phân tích kết quả của việc phân khúc có thể cung cấp thông tin giúp xác định mục tiêu thị trường hiệu quả hơn, tối ưu vốn, thúc đẩy phát triển chiến lược, ... Từ nhu cầu phân khúc khách hàng đỡ tốn công sức và chi phí, các thuật toán machine learning được áp dụng. Nghiên cứu này được thực hiện bởi Arief Chaerudin cùng các cộng sự thuộc Khoa Khoa học máy tính tại trường đại học Telkom ở Indonesia. Bộ dữ liệu được sử dụng có tên ‘Mall Customer Segmentation’ được lấy từ Kaggle¹. Trong nghiên cứu, ông cùng các cộng sự đã so sánh và chỉ ra thuật toán K-means++ được cho là phân cụm tốt hơn và nhanh hơn một chút so với K-means, với Silhouette score là 0.553931997444648 cho K-means++ và 0.553217610757543 cho K-means. K-means++ là thực chất là K-means, nhưng thay vì khởi tạo điểm trung tâm cụm (centroid) một cách ngẫu nhiên như K-means thì các centroid của K-means++ được khởi tạo qua thuật toán để cải thiện các cụm và tốc độ hội tụ của các cụm. Phương pháp Elbow cũng được sử dụng để xác định K (1-11), K được xác định trong nghiên cứu là 5. Dựa vào thuộc tính Thu nhập hằng năm (Annual Income) và Điểm đánh giá chi tiêu khách hàng (Spending score), nghiên cứu đã phân ra được 5 cụm. Từ nghiên cứu có thể suy luận ra độ ưu tiên của khách hàng và tập trung vào nhóm khách hàng đó.



Hình 2.2 *Trực quan kết quả các cụm của nghiên cứu 2.2[3]*

¹ Bộ dữ liệu Mall Customer Segmentation Data được lấy từ Kaggle:
<https://www.kaggle.com/vjchoudhary7/customer-segmentation-tutorial-in-python>

Number of nodes	K-Means	K-Means++
Blue cluster	39	39
Brown cluster	23	23
Yellow cluster	80	81
Green cluster	36	35
Red cluster	22	22

Bảng 2.1 Thống kê dữ liệu trong 5 cụm của 2 thuật toán K-Means và K-Means++ của nghiên cứu 2.2[3]

2.3. Machine Learning Based Classification And Segmentation Techniques For Crm: A Customer Analytics[17]

Machine learning và data mining giúp cho các doanh nghiệp xây dựng những công cụ thực hiện một số chức năng dựa trên thông tin khách hàng. Thông tin về khách hàng là cơ sở để duy trì mối quan hệ lâu dài cũng như để quản lý khách hàng. Nhận biết và phân khúc khách hàng được thực hiện để giữ chân khách hàng hiệu quả hơn, đồng thời tối ưu lợi nhuận và tăng hiệu suất làm việc. Trong nghiên cứu này, Narendra Singh cùng các cộng sự đã đề xuất phân khúc khách hàng dựa trên những thuộc tính nhân khẩu học như giới tính, tuổi, thu nhập và điểm đánh giá chi tiêu. Bộ dữ liệu được sử dụng trong nghiên cứu này là ‘Mall Customer’ mã nguồn mở được công khai bởi Choudhary năm 2018². Dựa vào thương của Thu nhập hằng năm (Annual Income) và Điểm đánh giá chi tiêu (Spending Score) – $Us = \text{Annual Income} / \text{Spending Score}$, ông và các cộng sự đã phân khách hàng vào các lớp: Gold ($Us > 3.5$), Silver ($1.5 < Us \leq 3.5$), Elite ($0.75 < Us \leq 1.5$), Occasional ($Us \leq 0.75$). Sau đó, Narendra Singh dùng các thuật toán phân lớp để xây dựng mô hình dự đoán lớp cho khách hàng. Bằng các thuật toán như Multi-layer perceptron (MLP), Naïve Bayes, regression và J48, ông cùng các cộng sự lần lượt xây dựng mô hình và so sánh. Cuối cùng, MLP cho kết quả khả quan nhất với độ chính xác Accuray là 98.33% và tổng thời gian chạy là 0.26 giây. Trong nghiên cứu này, Narendra Singh đã thành công trong việc phân khúc khách hàng vào trong các lớp giúp doanh nghiệp dễ dàng thực hiện chính sách ưu đãi, giảm giá cho từng nhóm khách hàng.

² Bộ dữ liệu Mall Customer được công khai bởi ông Choudhary:
<https://www.kaggle.com/vjchoudhary7/customer-segmentation-tutorial-in-python>

	Silver	Gold	Occasional	Elite
Silver	6	0	0	0
Gold	1	3	0	0
Occasional	0	0	19	0
Elite	0	0	0	31

Bảng 2.2 Confusion matrix của thuật toán MLP trong nghiên cứu 2.3[17]

Chương 3. CƠ SỞ LÝ THUYẾT

3.1. Khoảng cách Euclide

- Là khoảng cách hai điểm, vector trong không gian được định nghĩa bằng công thức:

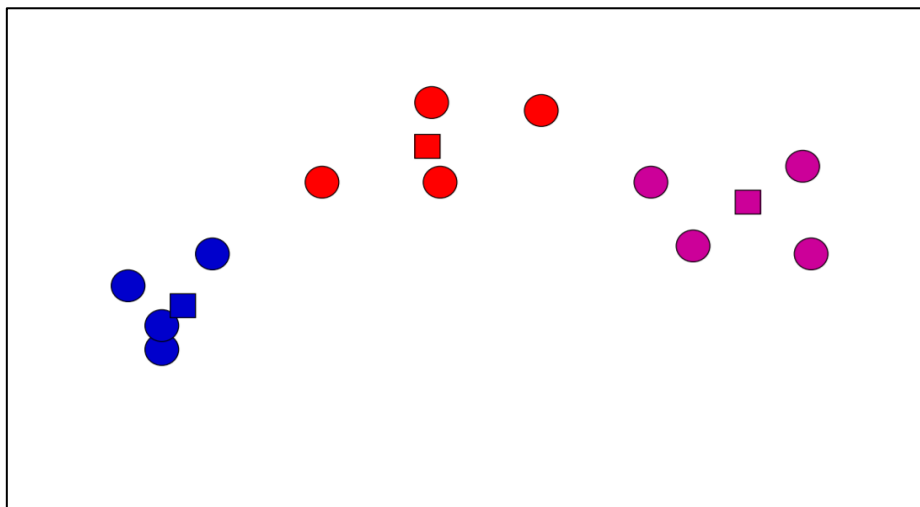
$$\text{Distance}(x, y) = \sqrt{\sum_{k=1}^N (x_k - y_k)^2}$$

- Trong đó:
 - + N là số chiều của vector (là số lượng thuộc tính trong bài toán xây dựng mô hình này).
 - + \mathbf{x}, \mathbf{y} là hai vector có N chiều (là điểm dữ liệu trong bài toán).
 - + $\mathbf{x}_k, \mathbf{y}_k$ là thành phần thuộc tính (đặc trưng) thứ k .

3.2. K-Means Clustering

3.2.1. Định nghĩa

K-Means Clustering là một thuật toán phân cụm đơn giản thuộc loại học không giám sát (unsupervised machine learning) được sử dụng nhiều trong khai thác dữ liệu và thống kê. Nó được sử dụng trong phân tích tính chất của cụm dữ liệu, công việc phân cụm xác lập dựa trên nguyên lý: phân tập dữ liệu thành k cụm, các điểm dữ liệu trong 1 cụm phải có cùng một số tính chất nhất định, cụ thể ở đây là khoảng cách.



Hình 3.1 Minh họa thuật toán K-Means

3.2.2. Phân tích toán học

3.2.2.1. Đầu vào, đầu ra:

- **Đầu vào:**

- Ma trận các điểm dữ liệu:

$$X = [x_1, x_2, \dots, x_N] \in R^{d \times N} \quad \text{với } d \text{ là số chiều của } x_i.$$

- Số cluster $K < N$.

- **Đầu ra:**

- Ma trận các centroid:

$$M = [m_1, m_2, \dots, m_N] \in R^{d \times K}$$

- Ma trận các label:

$$Y = \begin{bmatrix} y_{11}, y_{12}, \dots, y_{1K} \\ y_{21}, y_{22}, \dots, y_{2K} \\ \dots \\ y_{K1}, y_{K2}, \dots, y_{KK} \end{bmatrix}$$

- Nếu x_i thuộc cluster có m_j là centroid thì $y_{ij} = 1$, còn lại $y_{ik} = 0$ với $k = 1, 2, \dots, K$ và $k \neq j$.

3.2.2.2. Hàm mất mát

- Nếu ta xem m_k là centroid của mỗi cluster và ước lượng tất cả các điểm được phân vào cluster bởi centroid m_k , thì một điểm dữ liệu x_i được phân vào cluster

k sẽ có sai số là $\|x_i - m_k\|_2^2$.

- $\|x_i - m_k\|_2$ là khoảng cách Euclide của hai vector x_i và m_k .
- $\|x_i - m_k\|_2^2$ là bình phương của $\|x_i - m_k\|_2$ hay nói cách khác là tổng bình phương mỗi phần tử của vector $(x_i - m_k)$.

- Hàm mất mát tại \mathbf{x}_i thuộc cluster \mathbf{m}_k : Do $y_{ij} = 1$ khi và chỉ khi $\mathbf{j} = \mathbf{k}$:

$$\|x_i - m_k\|_2^2 = \sum_{j=1}^K y_{ij} \|x_i - m_j\|_2^2$$

- Từ đây ta suy ra hàm mất mát cho toàn bộ dữ liệu:

$$\sum_{i=1}^N \sum_{j=1}^K y_{ij} \|x_i - m_j\|_2^2$$

3.2.2.3. Tối ưu hóa hàm mất mát

- **Cố định M, tìm Y:** Giả sử đã tìm được các center, hãy tìm các label vector để làm mất mát đạt giá trị nhỏ nhất.
 - Trường hợp này khá đơn giản, khi ta đã có centroid cố định việc tối ưu hàm mất mát chỉ là việc xếp \mathbf{x}_i vào cluster với centroid gần nó nhất.
- **Cố định Y, tìm M:** Giả sử đã tìm được các cluster cho từng điểm, hãy tìm các center mới cho mỗi cluster để hàm mất mát đạt giá trị nhỏ nhất.
 - Trường hợp này thì phức tạp hơn, bài toán tìm centroid \mathbf{m}_k cho cluster \mathbf{k} sao cho hàm mất mát nhỏ nhất trở thành:

$$\begin{aligned} & \min \sum_{k=1}^N y_{ik} \|x_i - m_k\|_2^2 \\ &= \min \sum_{k=1}^N \|x_i - m_k\|_2^2 \\ &= \min(\|x_1 - m_k\|_2^2 + \|x_2 - m_k\|_2^2 + \dots + \|x_N - m_k\|_2^2) \\ &= \min(f(m_k)) \end{aligned}$$

- Để tìm min hàm $f(\mathbf{m}_k)$ ta thực hiện đạo hàm.

$$\begin{aligned}
 f'(\mathbf{m}_k) &= 0 \\
 \Leftrightarrow -2(x_1 + x_2 + \dots + x_N) + 2N \cdot m_k &= 0 \\
 \Leftrightarrow m_k &= \frac{x_1 + x_2 + \dots + x_N}{N} = \text{mean}(\mathbf{x})
 \end{aligned}$$

- Ta thấy:

$$f'(\mathbf{m}_k) = -2(x_1 + x_2 + \dots + x_N) + 2N \cdot m_k$$

Bảng biến thiên:

\mathbf{m}_k	$-\infty$	$\frac{x_1 + x_2 + \dots + x_N}{N}$	$+\infty$
$f'(\mathbf{m}_k)$	$-\infty$	-	+
		0	$+\infty$

Bảng 3.1 Bảng biến thiên hàm $f(m_k)$

- Do đó $\mathbf{m}_k = \text{mean}(\mathbf{x})$ là giá trị để hàm mất mát đạt giá trị nhỏ nhất.
 \Rightarrow Cái tên K-Means cũng xuất hiện từ đây.

3.2.3. Thuật toán

Bước 1: Chọn **K** điểm bất kỳ làm centroid ban đầu.

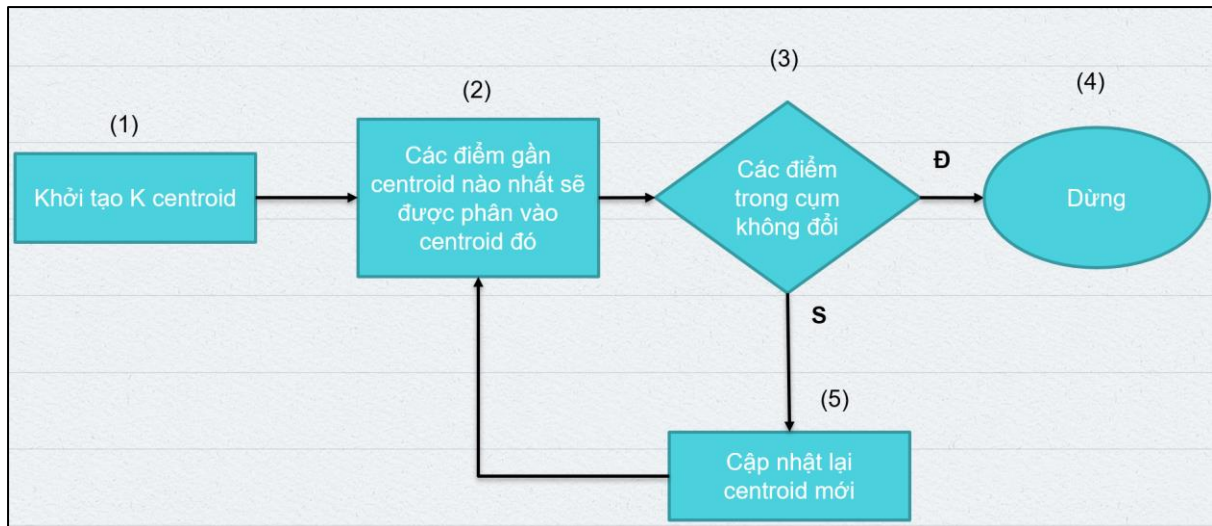
Bước 2: Phân mỗi điểm dữ liệu vào cluster có centroid gần với nó nhất.

Bước 3: Nếu việc phân các điểm dữ liệu vào các cluster không thay đổi thì ta dừng thuật toán.

Bước 4: Cập nhật centroid cho từng cluster bằng cách lấy trung bình cộng tất cả các điểm có trong cluster.

Bước 5: Quay lại bước 2.

Ta có thể đảm bảo rằng thuật toán này sẽ dừng lại ở một số hữu hạn vòng lặp do hàm mất mát bị chặn dưới (do nó luôn dương) và giảm sau mỗi lần hiệu chỉnh lại centroid và các điểm dữ liệu \Rightarrow hàm mất mát là hàm hội tụ.

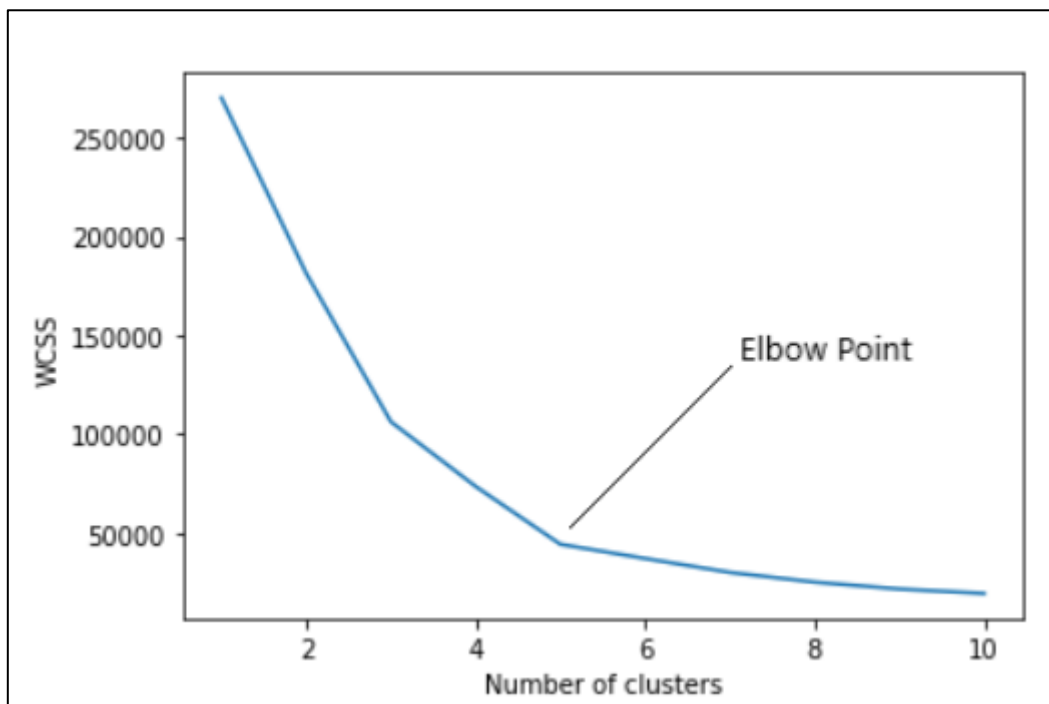


Hình 3.2 Sơ đồ mô tả 5 bước của thuật toán K-Means

3.3. Phương pháp Elbow

3.3.1. Định nghĩa

Là phương pháp thực nghiệm để lựa chọn số cụm tối ưu cho một tập dữ liệu. Phương pháp Elbow xem xét tổng bình phương các điểm trong các cụm sao cho khi thêm một cụm không làm thay đổi nhiều tổng bình phương ấy.



Hình 3.3 Minh họa phương pháp Elbow

3.3.2. Thực hiện

Bước 1: Chọn khoảng giá trị cho **k** (thường là 1-10).

Bước 2: Với mỗi **k** tính tổng bình phương các cụm.

Bước 3: Vẽ biểu đồ theo các giá trị tính được.

Bước 4: Chọn **k** tại vị trí “Elbow-khuỷu tay”.

3.4. Phương pháp Silhouette

3.4.1. Định nghĩa

Cũng là phương pháp xác định số lượng cụm tối ưu cho tập dữ liệu. Phương pháp Silhouette giải thích và xác nhận tính nhất quán của các cụm dữ liệu.

Phương pháp Elbow và Silhouette đều được sử dụng để tìm số cụm tối ưu. Nhưng phương pháp Elbow có vẻ gây ra mơ hồ trong việc chọn “giá trị K khuỷu tay”, chúng ta cần trực quan và quan sát kỹ để lựa chọn, nhưng không phải bộ dữ liệu nào cũng cho phép chúng ta làm điều này. Về phương pháp Silhouette, phương pháp này cung cấp một biểu diễn là giá trị cho biết mỗi đối tượng đã được phân nhóm tốt như thế nào. Ngoài ra, phương pháp Silhouette được cho là có lợi thế hơn để tìm ra các ngoại lệ nếu có trong một cụm.

3.4.2. Tính hệ số Silhouette

- Tính hệ số Silhouette cho 1 điểm **i**:

$$s_i = \frac{a_i - b_i}{\max(a_i, b_i)}$$

Trong đó:

- a_i là khoảng cách từ điểm **i** đến tất cả các điểm trong cụm **A**.
 - b_i là khoảng cách từ điểm **i** đến tất cả các điểm trong cụm **B** – cụm gần nhất với cụm **A**.
- Sau khi tìm được hệ số Silhouette cho tất cả các điểm, ta tiến hành lấy trung bình và thu được Silhouette score cho **k** cụm.

3.4.3. Thực hiện

Bước 1: Chọn khoảng giá trị cho **k** (thường là 1-10).

Bước 2: Với mỗi **k** tính Silhouette score.

Bước 3: Chọn **k** với Silhouette score cao nhất.

3.5. Gaussian Naïve Bayes

Mô hình này có đặc điểm là tốc độ training và test đều cực kỳ nhanh, đặc biệt phù hợp với loại dữ liệu mà các thuộc tính là các biến liên tục. Vì vậy nó phù hợp với bài toán phân lớp dữ liệu của chúng ta.

3.5.1. Định lý xác suất bayes

- **x, y** là hai sự kiện liên quan đến nhau, xác suất của sự kiện **y** khi sự kiện **x** đã xảy ra là:

$$P(y | x) = \frac{P(x | y) \cdot P(y)}{P(x)}$$

- Trong đó:
 - + **P(x|y)** xác suất xảy ra sự kiện **x** khi sự kiện **y** đã xảy ra.
 - + **P(y)** là xác suất sự kiện **y** xảy ra.
 - + **P(x)** là xác suất sự kiện **x** xảy ra.
- Ví dụ: Một sản phẩm được tạo nên bởi 2 linh kiện: A – 40% và B – 60%. Nếu một trong hai linh kiện hư thì sản phẩm này sẽ hư. Xác suất linh kiện A hư là 10% và B là 5%. Sản phẩm này bị hư, tính xác suất là do linh kiện A hư.
 - + Gọi A là sự kiện linh kiện A hư.
 - + Gọi B là sự kiện linh kiện B hư.
 - + Gọi x là xác suất sản phẩm hư:
$$p(x) = p(A) \cdot p(x|A) + p(B) \cdot p(x|B) = 0.4 \cdot 0.1 + 0.6 \cdot 0.05 = 0.07$$
 - + Xác suất sản phẩm hư là do linh kiện A hư:
$$p(A|x) = p(x|A) \cdot p(A) / p(x) = 0.4 \cdot 0.1 / 0.07 = 0.5714$$

3.5.2. Naïve Bayes

- Xét bài toán phân loại nhãn $(1, 2, \dots, C)$ cho 1 điểm dữ liệu gồm d chiều. Xác suất điểm dữ liệu này rơi vào class c thuộc một trong các class $1, 2, \dots, C$ bất kỳ là:

$$p(y = c | x)$$

- Naïve Bayes classifier quy định điểm dữ liệu x thuộc lớp c nếu xác suất điểm dữ liệu x rơi vào lớp c là lớn nhất.

3.5.2.1. Phân tích toán học

- Bài toán đặt ra: Tìm c để $p(c|x)$ lớn nhất.
- Ta có:

$$p(c | x)_{\max} = \frac{p(x | c) \cdot p(c)}{p(x)_{\max}} = [p(x | c) \cdot p(c)]_{\max}$$

- $p(x)$ là xác suất để tìm thấy x trong tập dữ liệu. nhưng ta thấy được $p(x)$ không phụ thuộc vào c nên ta sẽ không quan tâm $p(x)$ trong bài toán này.
- $p(c)$ là xác suất để một điểm dữ liệu rơi vào tập c được tính bằng cách lấy số điểm dữ liệu thuộc class c chia cho tổng số điểm dữ liệu thuộc tập train.
- $p(x|c)$ là xác suất gặp điểm dữ liệu thuộc nhãn c tương tự với x . Do điểm dữ liệu gồm rất nhiều chiều nên muốn bắt gặp điểm dữ liệu như vậy thì cần một tập dữ liệu rất lớn. Vì thế trong thực tế, người ta xem từng thuộc tính của điểm dữ liệu là biến độc lập và $p(x|c)$ được tính như sau:

$$p(x | c) = \prod_{i=1}^d p(x_i | c) \quad (1)$$

- Bài toán cuối cùng trở thành:

$$\left[p(c) \cdot \prod_{i=1}^d p(x_i | c) \right]_{\max} \quad (2)$$

- Ở bước train, $\mathbf{p}(\mathbf{c})$ và các $\mathbf{p}(\mathbf{x}_i|\mathbf{c})$ sẽ được tính thông qua tập train.
- Ở bước test, ta tìm class thỏa biểu thức (2).

3.5.2.2. Gaussian Naïve Bayes

- Với mỗi chiều dữ liệu \mathbf{x}_i thay vì được tính theo biểu thức (1) thì bây giờ sẽ tính theo biểu thức sau:

$$p(x_i | c) = p(x_i | \mu_{ci}, \sigma_{ci}^2) = \frac{1}{\sqrt{2\pi\sigma_{ci}^2}} e^{-\frac{(x_i - \mu_{ci})^2}{2\sigma_{ci}^2}}$$

- Trong đó:
 - + μ_{ci} là kỳ vọng trung bình của thuộc tính \mathbf{x}_i trong tập \mathbf{c} .
 - + σ_{ci}^2 là phương sai trung bình của thuộc tính \mathbf{x}_i trong tập \mathbf{c} .

(Hai số này được xác định bằng Maximum Likelihood, tham khảo thêm cách xác định tại Viblo[21])

3.6. Các độ đo

Trong quá trình xây dựng một mô hình phân lớp bằng machine learning, một phần không thể thiếu là đánh giá mô hình, để biết được chất lượng mô hình tốt thế nào, ta sử dụng các độ đo để đánh giá. Nhưng không phải có một độ đo và không phải độ đo nào cũng có thể phản ánh đúng chất lượng của một mô hình. Vì vậy ta sẽ khảo sát qua một số độ đo như Accuracy, Precision, Recall và F1-score.

Trong đánh giá một mô hình phân lớp, ta sẽ có định nghĩa sau:

- True Positive – TP: giá trị actual và predicted đều là positive.
- False Positive – FP: giá trị actual là negative nhưng predicted là positive.
- True Negative – TN: giá trị actual và predicted đều là negative.
- False Negative – FN: giá trị actual là positive nhưng predicted là negative.

		Predict	
		Positive	Negative
Actual	Positive	True Positive	False Negative
	Negative	False Positive	True Negative

Bảng 3.2 Confusion matrix của một mô hình phân lớp

3.6.1. Accuracy

- Độ đo Accuracy khái quát tỷ lệ các trường hợp được dự báo đúng trên tổng số các trường hợp. Độ đo accuracy là độ đo đơn giản nhất và thường được sử dụng. Độ đo accuracy được tính như sau:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

3.6.2. Precision

- Độ đo precision trả lời cho câu hỏi trong số các trường hợp được dự báo là positive thì tỉ lệ dự báo đúng là bao nhiêu. Precision được tính như sau:

$$Precision = \frac{TP}{TP + FP}$$

3.6.3. Recall

- Cũng tương tự như precision, nhưng khác một chút. Trong tất cả các mẫu positive thì recall tính tỉ lệ có bao nhiêu mẫu được dự báo đúng. Công thức tính recall như sau:

$$Recall = \frac{TP}{TP + FN}$$

3.6.4. F1-score

- F1-score là trung bình điều hòa giữa precision và Recall. Vì vậy, F1-score sẽ đại diện cho cả precision và Recall đánh giá một mô hình. F1-score được tính như sau:

$$F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

- Tại sao là trung bình điều hòa mà không là trung bình cộng? Lý giải cho việc này ta xét ví dụ trường có precision = 0.01 và recall = 1.0, nếu ta lấy trung bình cộng thì $f1 = 0.5005$, còn nếu ta lấy trung bình điều hòa thì $f1 = 0.019$. Qua ví dụ trên, ta thấy được trung bình điều hòa sẽ cho ta biết được trường hợp mô hình tệ như thế nào trong trường hợp precision cao và recall thấp hoặc ngược lại. Trong khi đó trung bình cộng sẽ không cho ta biết được điều này.
- So sánh giữa F1-score và Accuracy ta xét ví dụ sau:

		Predict		Recall
		Yes	No	
Actual	Yes	850	50	0.94
	No	60	40	0.4
Precision		0.93	0.44	

Bảng 3.3 Confusion matrix tự tạo của một mô hình với bộ dữ liệu mất cân bằng

Ta tính:

$$+ \text{ Accuracy} = (850+40)/(850+50+60+40) = 89\%$$

+ F1-score:

$$\bullet \text{ F1-score (Yes)} = (2 \times 0.93 \times 0.94)/(0.93+0.94) = 0.93$$

$$\bullet \text{ F1-score (No)} = (2 \times 0.44 \times 0.4)/(0.44+0.4) = 0.41$$

$$\Rightarrow \text{ F1-score} = (0.93+0.41)/2 = 0.67 = 67\%$$

Kết luận: qua ví dụ trên ta thấy được với một bộ dữ liệu mất cân bằng thì F1-score cho độ chính xác là **67%**, trong khi đó Accuracy cho độ chính xác là **89%**, gần như là xấp xỉ với độ chính xác của mẫu chiếm đa số (Yes chiếm

90%) với độ chính xác theo Accuracy là 94%. Nhưng nhìn lại khi dự đoán trên mẫu No thì độ chính xác Accuracy chỉ là 40%. Vì vậy, độ đo Accuracy chưa làm tốt trong trường hợp này. Sự chênh lệch giữa Accuracy và F1-score cho ta thấy với một bộ dữ liệu mất cân bằng thì F1-score sẽ đánh giá khách quan hơn Accuracy.

Chương 4. BỘ DỮ LIỆU

4.1. Bộ dữ liệu

Bộ dữ liệu tên Mall Customer³ là file .csv được lấy từ liên kết Google Drive. Bộ dữ liệu bao gồm 200 điểm dữ liệu với 5 thuộc tính: CustomerID (Mã số định danh khách hàng từ 1 đến 200), Gender (Giới tính khách hàng), Age (Tuổi của khách hàng), Annual Income (k\$) (Thu nhập năm trước), Spending Score (1-100). Thuộc tính Spending Score (1-100) là điểm số được trung tâm mua sắm tính toán dựa chủ yếu vào số tiền khách hàng bỏ ra trong năm. Điểm số từ thấp đến cao nằm trong đoạn từ 1 đến 100.

CustomerID	Gender	Age	Annual Income (k\$)	Spending Score (1-100)
1	Male	19	15	39
2	Male	21	15	81
3	Female	20	16	6
4	Female	23	16	77
5	Female	31	17	40
6	Female	22	17	76
7	Female	35	18	6
8	Female	23	18	94
9	Male	64	19	3
10	Female	30	19	72

Bảng 4.1 Mô tả một số điểm dữ liệu trong tập dữ liệu

4.2. Tiền xử lý

Do chúng chúng ta xây dựng mô hình mà không sử dụng hết các thuộc tính của tập dữ liệu ban đầu nên ta cần tiền xử lý chúng một chút trước khi đưa vào huấn luyện.

³ Bộ dữ liệu Mall Customer được lấy từ liên kết Google Drive:
<https://drive.google.com/file/d/19BOhwz52NUY3dg8XErVYglctpr5sjTy4/view>

Bước 1: Đầu tiên chúng ta, tạo một tập dữ liệu từ file .csv có được: Hàm **read_csv** của thư viện **pandas** được gọi thực hiện để đọc dữ liệu từ file .csv và gán cho dataframe **df**.

```
#Tạo dataset từ file csv
import pandas as pd
df=pd.read_csv('/content/drive/MyDrive/DS102_CK/Mall_Customers.csv')
```

Hình 4.1 Code load dataset

df

	CustomerID	Gender	Age	Annual Income (k\$)	Spending Score (1-100)
0	1	Male	19	15	39
1	2	Male	21	15	81
2	3	Female	20	16	6
3	4	Female	23	16	77
4	5	Female	31	17	40
...
195	196	Female	35	120	79
196	197	Female	45	126	28
197	198	Male	32	126	74
198	199	Male	32	137	18
199	200	Male	30	137	83

200 rows × 5 columns

Hình 4.2 Mô tả tập df ở giai đoạn tiền xử lý bước 1

Bước 2: Do đầu vào của 2 mô hình ta muốn xây dựng là khác nhau nên phần tiền xử lý của 2 mô hình là khác nhau.

Mô hình 1

Trong mô hình này, chúng ta sẽ không sử dụng thuộc tính CustomerID nên thuộc tính này sẽ bị xóa đi. Các thuộc tính còn lại như Gender, Annual Income (k\$) và Spending Score (1-100) có miền giá trị độ khoảng 0 đến 150 nên để dễ dàng trực quan dữ liệu, chúng ta sẽ chuyển thuộc tính Gender về dạng số integer với Male là 100 và Female là 50.

```
#Tiền xử lý dữ liệu
df=df.drop(['CustomerID'],axis=1)
df['Gender'][df.Gender == 'Male'] = 100
df['Gender'][df.Gender == 'Female'] = 50
df['Gender']=df['Gender'].astype(int)
```

Hình 4.3 Code tiền xử lý dữ liệu cho mô hình 1

Mô hình 2

Còn trong mô hình này, chúng ta chỉ sử dụng 2 thuộc tính Annual Income (k\$) và Spending Score (1-100) nên ta xóa các thuộc tính còn lại như CustomerID, Gender, Age.

```
#Tiền xử lý dữ liệu
df=df.drop(['CustomerID',
            'Gender',
            'Age'],axis=1)
```

Hình 4.4 Code tiền xử lý cho mô hình 2

Chương 5. XÂY DỰNG MÔ HÌNH

5.1. Xây dựng mô hình

5.1.1. Mô hình 1

Ở phần này, chúng tôi sẽ trình bày cách xây dựng mô hình phân nhóm khách hàng với thuật toán K-Means, sau đó dùng Naïve Bayes xây dựng mô hình dự đoán nhóm khách hàng các thuộc tính nhân khẩu học như: Gender, Age, Annual Income (k\$), Spending Score (1-100).

	Gender	Age	Annual Income (k\$)	Spending Score (1-100)
0	100	19	15	39
1	100	21	15	81
2	50	20	16	6
3	50	23	16	77
4	50	31	17	40
...
195	50	35	120	79
196	50	45	126	28
197	100	32	126	74
198	100	32	137	18
199	100	30	137	83

200 rows × 4 columns

Hình 5.1 Dữ liệu tập dữ liệu trước khi đưa vào phân nhóm ở mô hình 1

Bước 1: Thêm một số thư viện và module cần thiết.

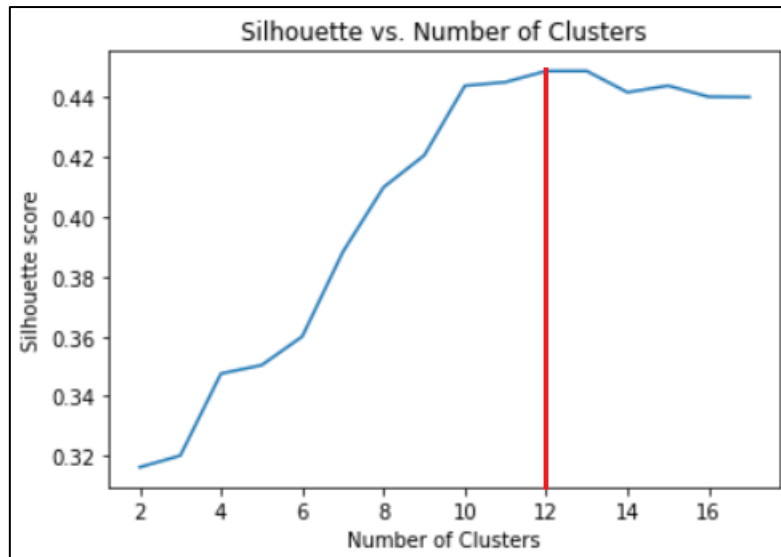
```
#import thư viện
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
import matplotlib.pyplot as plt
import plotly.express as px
```

Hình 5.2 Code thêm các thư viện, module cho quá trình phân nhóm của mô hình 1

Bước 2: Tìm **K** cho mô hình K-Means: Phương pháp silhouette được sử dụng. Đầu tiên tạo một mảng silhouette score có tên **sil_score**. Sau đó, ta cho số lượng cụm chạy từ 2 đến 17 (đã chạy từ 2 đến 51 và thấy rằng đoạn từ 2 đến 17 ta dễ dàng quan sát được số cụm phù hợp) và trực quan nó lên bằng module **pyplot** của thư viện **matplotlib**. Ta thấy được **K=12** cho silhouette score cao nhất.

```
#Tìm K phù hợp cho thuật toán Kmeans
sil_score=[]
for i in range(2,18):
    kmeans = KMeans(n_clusters=i, init= 'k-means++', random_state=42)
    kmeans.fit(df)
    preds = kmeans.predict(df)
    score = silhouette_score(df, preds, metric='euclidean')
    sil_score.append(score)
plt.plot(range(2,18),sil_score)
plt.title('Silhouette vs. Number of Clusters')
plt.xlabel('Number of Clusters')
plt.ylabel('Silhouette score')
plt.show()
```

Hình 5.3 Code tìm K cho thuật toán K-Means trong mô hình 1



Hình 5.4 Trực quan hóa silhouette của mô hình 1

Bước 3: Huấn luyện mô hình K-Means bằng lớp **KMeans** thuộc module **sklearn.cluster** với tham số là **n_clusters** =12 (số cụm), **init** = ‘k-means++’ (khởi tạo) và **random_state** = 42. Các thuộc tính được sử dụng lúc này là Gender, Age, Annual Income (k\$), Spending Score (1-100). Sau đó gán label các cụm lại cho tập **df**. Khởi tạo với ‘k-means++’ sẽ tối ưu được thời gian hội tụ của thuật toán.

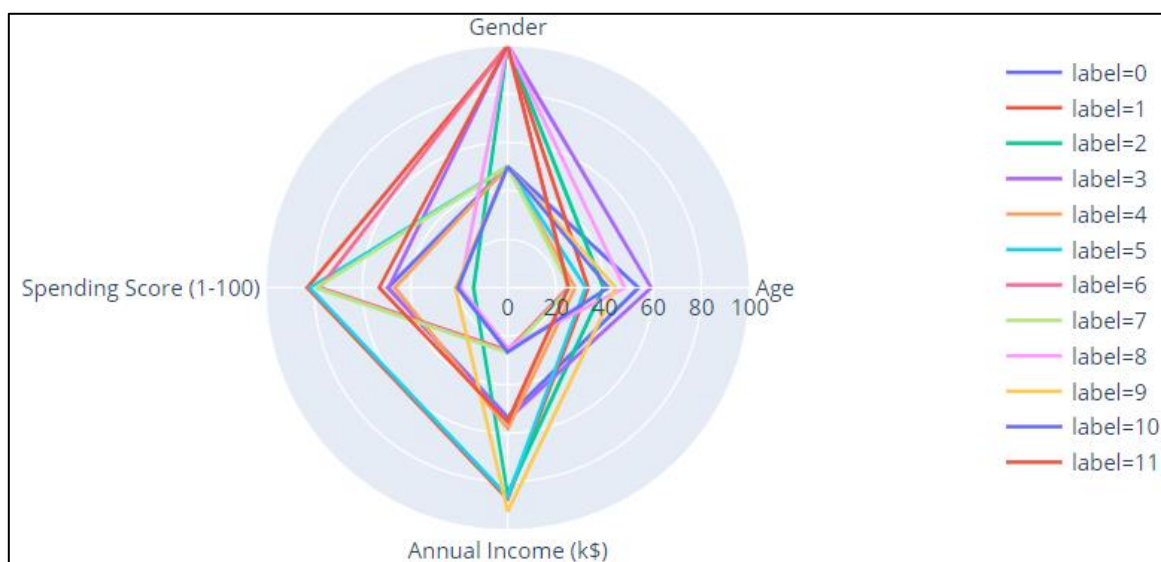
```
#Huấn luyện mô hình
kmeans = KMeans(n_clusters=12, init='k-means++', random_state=42)
kmeans.fit(df)
df['label']=kmeans.labels_
```

Hình 5.5 Code huấn luyện mô hình phân nhóm bằng K-Means trong mô hình 1

Bước 4: Trực quan hóa dữ liệu tập **df**: Thực hiện groupby theo thuộc tính label, tính trung bình các thuộc tính còn lại. Sau đó trực quan bằng hàm **line_polar** của module **plotly.express**.

```
#Trực quan hóa bằng giá trị trung bình các thuộc tính của 12 cụm
polar=df.groupby(by=['label']).mean().reset_index()
polar=pd.melt(polar,id_vars=['label'])
fig4 = px.line_polar(polar, r='value', theta='variable', color='label',
                    line_close=True,height=400,width=700)
fig4.show()
```

Hình 5.6 Code trực quan hóa 12 cụm sau khi phân nhóm của mô hình 1

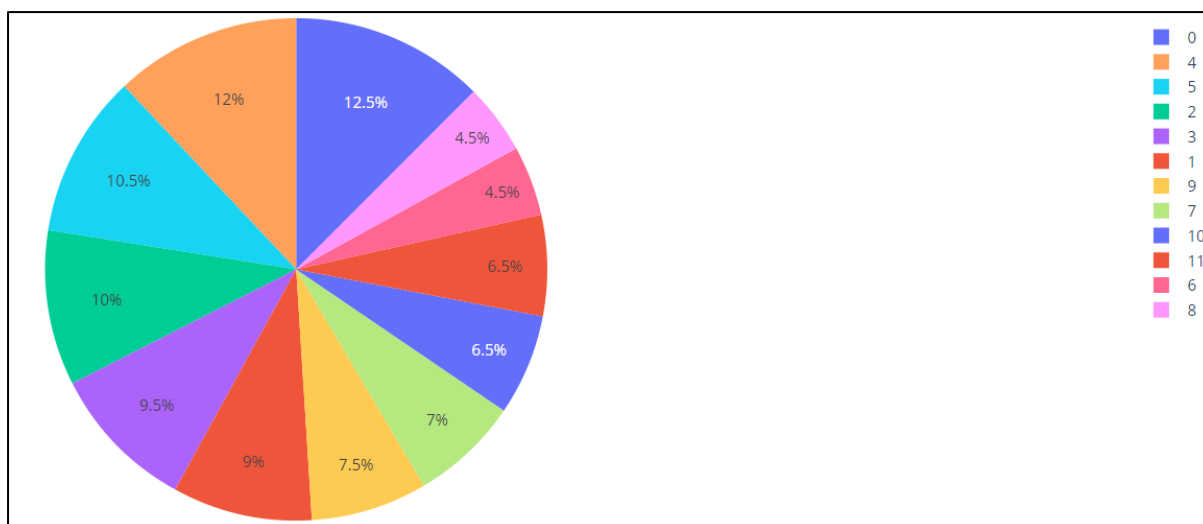


Hình 5.7 Trực quan hóa 12 cụm sau phân nhóm của mô hình 1

Bước 5: Trực quan hóa sự phân bố dữ liệu vào các cluster bằng hàm **pie** của module **plotly.express**.

```
#Trực quan hóa sự phân bố của các điểm dữ liệu vào 12 cụm
pie=df.groupby('label').size().reset_index()
pie.columns=['label','value']
px.pie(pie,values='value',names='label',color='label')
```

Hình 5.8 Code trực quan hóa sự phân bố 200 điểm vào 12 cụm trong mô hình 1



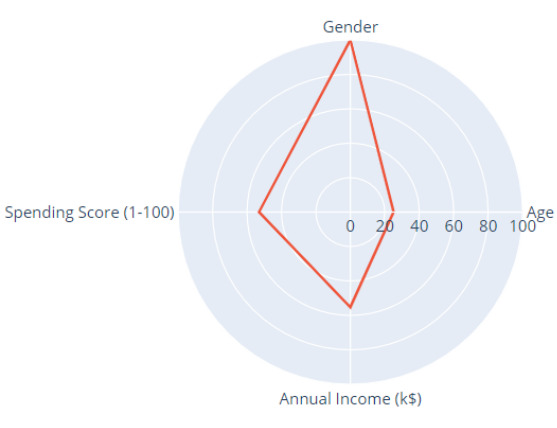
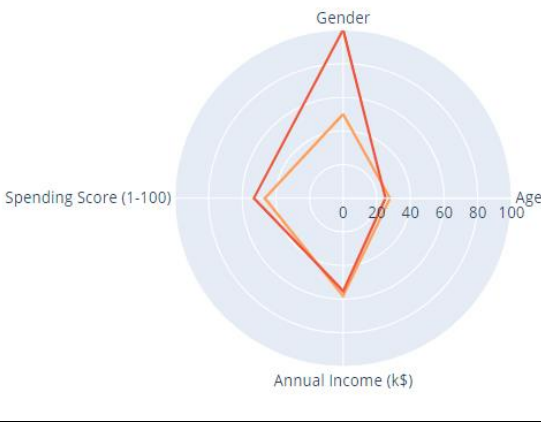
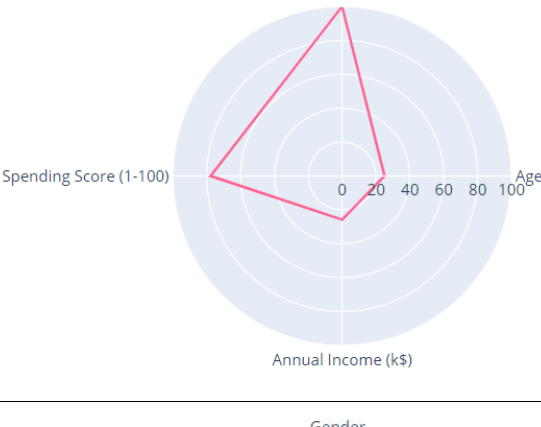
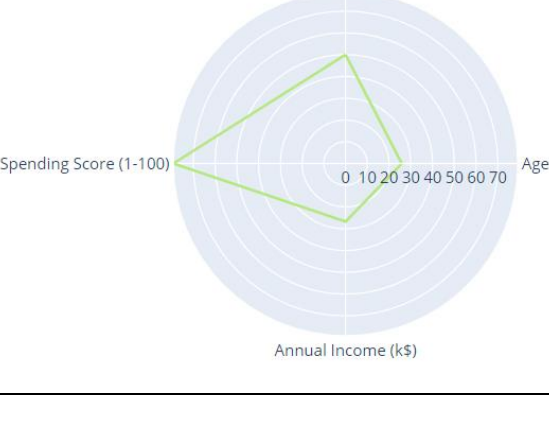
Hình 5.9 Sự phân bố của 200 điểm dữ liệu vào 12 cụm theo tỉ lệ % của mô hình 1

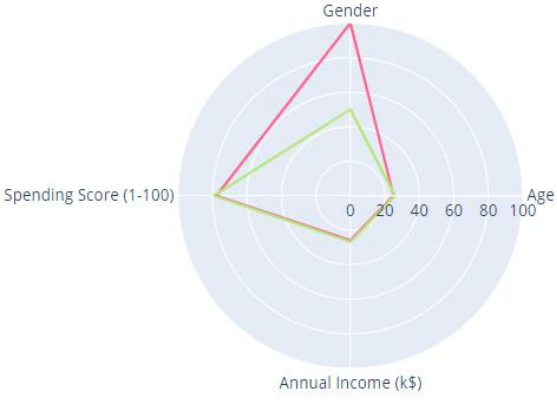
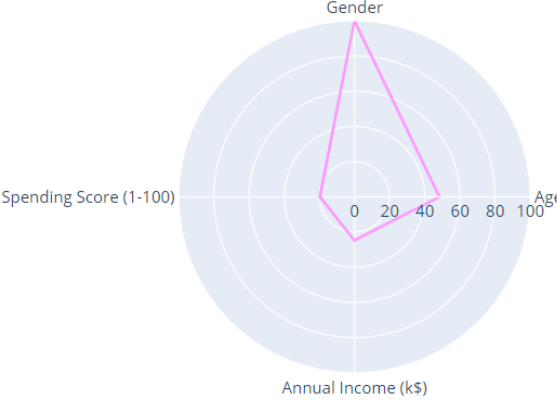
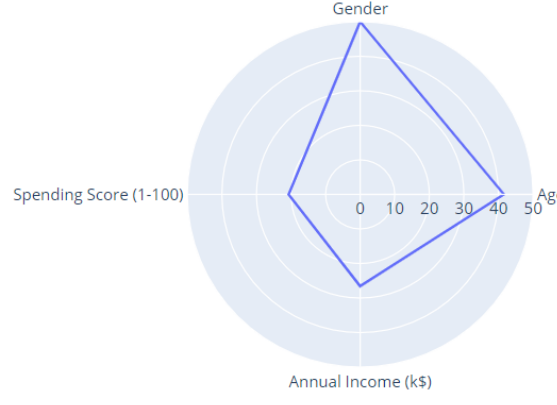
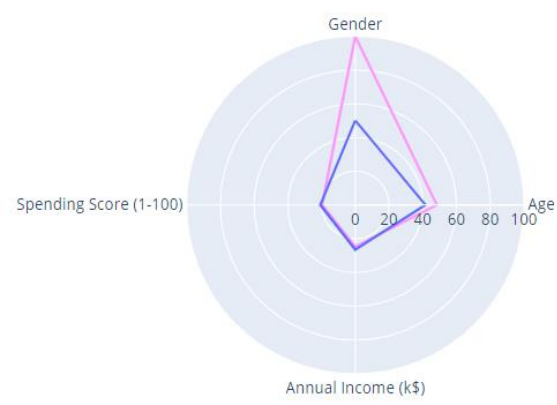
Sau khi dùng Kmeans để phân nhóm, ta được 12 phân khúc khách hàng như sau:

STT	Cluster	Mô tả	Số lượng điểm dữ liệu (Tổng 200)	Hình ảnh trực quan
1	0	Đây là nhóm khách hàng Nữ, có thu nhập trung bình (50-60k\$), chi tiêu trung bình (50), và tuổi trung bình (50-60 tuổi).	25	
2	3	Đây là nhóm khách hàng Nam, có thu nhập trung bình (50-60k\$), chi tiêu trung bình (score: 50), và tuổi trung bình (50-60 tuổi).	19	

<p>Kết luận: Đây là 2 nhóm thường đã có gia đình, an toàn về tài chính và không còn “năng động” với các khuyến mãi => chỉ có thể khai thác thêm ít.</p>				<p>A radar chart with four axes: Gender, Age, Annual Income (k\$), and Spending Score (1-100). The chart shows a purple line forming a diamond shape, indicating high scores across all four dimensions. The scale on the right axis (Age) ranges from 0 to 100.</p>
3	1	<p>Đây là nhóm khách hàng Nam trẻ trung, có độ tuổi trung bình từ 30 đến 40, có thu nhập cao (80-100k\$) và chi tiêu cao (score: 80-100).</p>	18	<p>A radar chart with four axes: Gender, Age, Annual Income (k\$), and Spending Score (1-100). The chart shows a red line forming a diamond shape, indicating high scores across all four dimensions. The scale on the right axis (Age) ranges from 0 to 100.</p>
4	5	<p>Đây là nhóm khách hàng Nữ, độ tuổi từ 30 đến 40, có thu nhập cao (90-100k\$) và chi tiêu cao (score: 80-100).</p>	21	<p>A radar chart with four axes: Gender, Age, Annual Income (k\$), and Spending Score (1-100). The chart shows a cyan line forming a diamond shape, indicating high scores across all four dimensions. The scale on the right axis (Age) ranges from 0 to 100.</p>
<p>Kết luận: Đây là 2 nhóm khách hàng ở độ tuổi trung niên có thu nhập và chi tiêu cao, có thể thấy được họ đang hưởng thụ cuộc sống. Vì vậy các chiến lược kinh doanh có thể nhắm vào nhóm người này.</p>				<p>A radar chart with four axes: Gender, Age, Annual Income (k\$), and Spending Score (1-100). The chart shows a red line forming a diamond shape, indicating high scores across all four dimensions. The scale on the right axis (Age) ranges from 0 to 100.</p>

5	2	Đây là nhóm khách hàng Nam, độ tuổi trung bình khoảng 40, có thu nhập cao (trên 80k\$) và có mức chi tiêu được đánh giá là thấp (điểm dưới 20)	20	<p>A radar chart with four axes: Gender, Age, Annual Income (k\$), and Spending Score (1-100). The chart shows a green line representing the profile of Group 5. The line is at the center for Gender and Age, extends to the 100 mark on the Annual Income axis, and stays very close to the center on the Spending Score axis.</p>
6	9	Đây là nhóm khách hàng Nữ, độ tuổi trung bình khoảng 40-50, có thu nhập cao (trên 90k\$) và có mức chi tiêu được đánh giá là thấp (điểm vào khoảng 20)	15	<p>A radar chart with four axes: Gender, Age, Annual Income (k\$), and Spending Score (1-100). The chart shows an orange line representing the profile of Group 6. The line is at the center for Gender and Age, extends to the 100 mark on the Annual Income axis, and stays very close to the center on the Spending Score axis.</p>
Kết luận: Đây là 2 nhóm khách ở độ tuổi 40-50, độ tuổi thường là đã có gia đình, sự nghiệp ổn định và có dư về tài chính. Vì vậy, họ không mua sắm, vui chơi một mình mà còn cho gia đình họ. Các khuyến mãi, combo nhiều người thường phù hợp với nhóm này. Đây là nhóm khách hàng tiềm năng cần khai thác nhiều hơn.				<p>A radar chart with four axes: Gender, Age, Annual Income (k\$), and Spending Score (1-100). The chart shows two overlapping lines: a green line for Group 6 and an orange line for Group 7. Both lines are at the center for Gender and Age, extend to the 100 mark on the Annual Income axis, and stay very close to the center on the Spending Score axis.</p>
7	4	Đây là nhóm khách hàng Nữ, trẻ trung, độ tuổi trung bình khoảng 20-30, có thu nhập trung bình (50-60k\$) và có mức chi tiêu được đánh giá là trung bình (điểm từ 40-50)	24	<p>A radar chart with four axes: Gender, Age, Annual Income (k\$), and Spending Score (1-100). The chart shows an orange line representing the profile of Group 7. The line is at the center for Gender and Age, extends to the 50 mark on the Annual Income axis, and extends to the 50 mark on the Spending Score axis.</p>

8	11	Đây là nhóm khách hàng Nam, trẻ trung, độ tuổi trung bình khoảng 20-30, có thu nhập trung bình (50-60k\$) và có mức chi tiêu được đánh giá là trung bình (điểm từ 40-50)	13	 <p>A radar chart with four axes: Gender, Age, Annual Income (k\$), and Spending Score (1-100). The chart shows a red line representing the profile of Group 8. The scores are approximately: Gender 100, Age 20, Annual Income 50, and Spending Score 50.</p>
Kết luận: 2 nhóm khách hàng thường ở độ tuổi độc thân nên có những chủ kiến, tư duy, cách mua hàng độc lập. Nhóm khách hàng này hãy quan tâm khi có thể.				 <p>A radar chart with four axes: Gender, Age, Annual Income (k\$), and Spending Score (1-100). The chart shows an orange line representing the profile of Group 11. The scores are approximately: Gender 100, Age 20, Annual Income 50, and Spending Score 50.</p>
9	6	Đây là khách hàng Nam, độ tuổi thanh niên rất trẻ (20-25 tuổi), có thu nhập thấp (khoảng 20k\$) nhưng chi tiêu rất cao (điểm vào khoảng 80).	9	 <p>A radar chart with four axes: Gender, Age, Annual Income (k\$), and Spending Score (1-100). The chart shows a pink line representing the profile of Group 9. The scores are approximately: Gender 100, Age 20, Annual Income 20, and Spending Score 80.</p>
10	7	Đây là khách hàng Nữ, độ tuổi thanh niên rất trẻ (20-25 tuổi), có thu nhập thấp (khoảng 20-30k\$) và chi tiêu rất cao (điểm vào khoảng 80).	14	 <p>A radar chart with four axes: Gender, Age, Annual Income (k\$), and Spending Score (1-100). The chart shows a green line representing the profile of Group 10. The scores are approximately: Gender 100, Age 20, Annual Income 20, and Spending Score 80.</p>

<p>Kết luận: Đây là nhóm khách hàng thường là còn đi học, có thu nhập thấp nhưng chi tiêu cao, thường không có tiềm lực về tài chính. Vì vậy, không cần tập trung vào nhóm này nhiều.</p>				
11	8	<p>Đây là nhóm khách hàng Nam, độ tuổi trung niên, trung bình khoảng 50 tuổi, có thu nhập thấp (khoảng 20k\$) và chi tiêu thấp (khoảng 20 điểm)</p>	9	
12	10	<p>Đây là nhóm khách hàng Nữ, độ tuổi trung niên, trung bình khoảng 50 tuổi, có thu nhập thấp (khoảng 20k\$) và chi tiêu thấp (khoảng 20 điểm).</p>	13	
<p>Kết luận: Đây là nhóm khách hàng đã bão hòa về nhu cầu và thu nhập khó có thể khai thác được nên không cần quá quan tâm.</p>				

Bảng 5.1 Mô tả 12 cụm sau khi phân chia của mô hình 1

Thống kê dữ liệu hiện tại trong các nhãn.

Label	Số điểm dữ liệu	Tỉ lệ (%)
0	25	12.5
1	18	9
2	20	10
3	19	9.5
4	24	12
5	21	10.5
6	9	4.5
7	14	7
8	9	4.5
9	15	7.5
10	13	6.5
11	13	6.5
Tổng	200	100

***Bảng 5.2** Thông kê các điểm trong 12 cụm sau khi phân chia của mô hình 1*

Bước 6: Giờ ta bắt đầu xây dựng mô hình phân lớp khách hàng bằng Naïve Bayes, đầu tiên thêm dữ liệu thư viện vào module cần thiết vào.

```
#Import các thư viện
from sklearn.naive_bayes import GaussianNB
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
import seaborn as sns
from sklearn.metrics import classification_report
```

***Hình 5.10** Code thêm các thư viện, module cho quá trình phân lớp ở mô hình 1*

Bước 7: Kiểm tra lại tập dữ liệu **df** hiện tại.

	Gender	Age	Annual Income (k\$)	Spending Score (1-100)	label
0	100	19	15	39	8
1	100	21	15	81	6
2	50	20	16	6	10
3	50	23	16	77	7
4	50	31	17	40	10
...
195	50	35	120	79	5
196	50	45	126	28	9
197	100	32	126	74	1
198	100	32	137	18	2
199	100	30	137	83	1

200 rows × 5 columns

Hình 5.11 Tập dữ liệu df tại Bước 7 quá trình xây dựng mô hình 1

Bước 8: Phân chia tập train, test cho tập **df**: ta dùng hàng **train_test_split** của module **sklearn.model_selection** phân chia tập **df** thành tập train (X_train, Y_train) và tập test (X_test, Y_test) theo tỉ lệ 8:2 (tương đương tập train có 160 điểm dữ liệu và tập test có 40 điểm dữ liệu)

```
#Chia tập dữ liệu X thành train test với tỉ lệ 8:2
X = df.iloc[:, 0:-1].values
Y = df.iloc[:, -1].values
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, train_size = 0.8,
                                                    random_state = 42)
```

Hình 5.12 Code chia dữ liệu thành tập train, test của mô hình 1

Bước 9: Huấn luyện mô hình bằng lớp **GaussianNB** của module **sklearn.naive_bayes**.

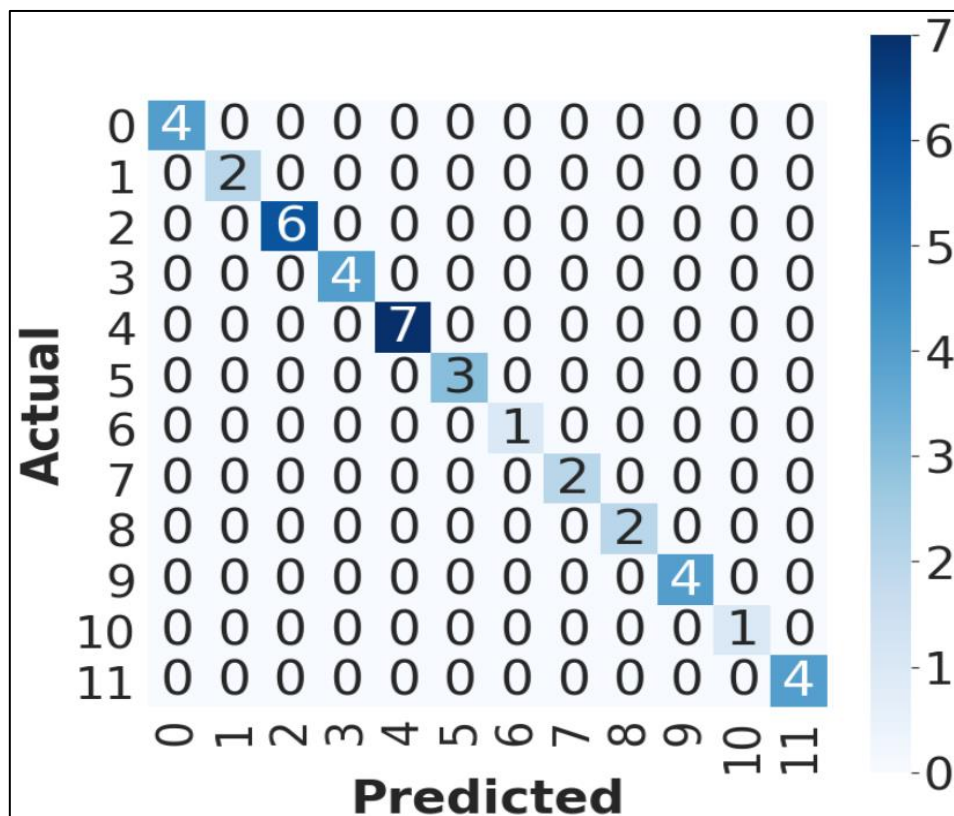
```
#Huấn luyện mô hình
GNB_classifier = GaussianNB()
GNB_classifier.fit(X_train, Y_train)
```

Hình 5.13 Code huấn luyện mô hình phân lớp bằng GaussianNB trong mô hình 1

Bước 10: Sử dụng thư viện **seaborn** và lớp **confusion_matrix** thuộc module **sklearn.metrics** để trực quan confusion matrix.

```
#In ra confusion matrix
cm = confusion_matrix(Y_test, GNB_classifier.predict(X_test))
plt.figure(figsize=(12, 12))
sns.set(font_scale=3.5)
ax = sns.heatmap(cm, cmap=plt.cm.Blues, annot=True, square=True, fmt = 'g')
ax.set_ylabel('Actual', fontsize=40, fontweight = 'bold')
ax.set_xlabel('Predicted', fontsize=40, fontweight = 'bold')
```

Hình 5.14 Code in ra confusion matrix trong mô hình 1



Hình 5.15 Confusion matrix của mô hình 1

Bước 11: Sử dụng hàm `classification_report` thuộc module `sklearn.metrics` để tính toán và in ra các độ đo đánh giá cần thiết cho một mô hình phân lớp.

```
#Classification report
print(classification_report(Y_test, GNB_classifier.predict(X_test)))
```

Hình 5.16 Code in classification report cho mô hình 1

	precision	recall	f1-score	support
0	1.00	1.00	1.00	4
1	1.00	1.00	1.00	2
2	1.00	1.00	1.00	6
3	1.00	1.00	1.00	4
4	1.00	1.00	1.00	7
5	1.00	1.00	1.00	3
6	1.00	1.00	1.00	1
7	1.00	1.00	1.00	2
8	1.00	1.00	1.00	2
9	1.00	1.00	1.00	4
10	1.00	1.00	1.00	1
11	1.00	1.00	1.00	4
accuracy			1.00	40
macro avg	1.00	1.00	1.00	40
weighted avg	1.00	1.00	1.00	40

Hình 5.17 Classification report của mô hình 1

Bước 12: Xây dựng một form và button ‘Predict’ để dự đoán một khách hàng. **Form** có sẵn ở **Google Colab**, còn button được thêm từ thư viện **ipwidgets** và module **IPython.display**. Chú ý, phải bật tính năng ‘Tự thực thi ô khi các trường thay đổi’ của **Form** để có thể luôn cập nhật các giá trị mới được nhập từ người dùng.

```
#@title Nhập các thuộc tính khách hàng { run: "auto" }
Age = 34#@param {type:"integer"}
Gender = "Male" #@param ["Male", "Female"]
Annual_Income = 100#@param {type:"integer"}
Spending_Score = 76#@param {type:"integer"}

#create a object
import numpy as np
if (Gender=='Male'):
    obj=np.array([100, Age, Annual_Income, Spending_Score])
if (Gender=='Female'):
    obj=np.array([50, Age, Annual_Income, Spending_Score])


Message = 'Cluster ' + str(GNB_classifier.predict(obj.reshape(1, -1))[0])

#button 'Predict'
import ipywidgets as widgets
from IPython.display import display
button = widgets.Button(description="Predict")
output = widgets.Output()

def on_button_clicked(b):
    with output:
        print(Message)

button.on_click(on_button_clicked)
display(button, output)
```

Hình 5.18 Code tạo button và form cho mô hình 1



Dự đoán một khách hàng thuộc nhóm nào (Cluster 0-11)

• Nhập các thuộc tính khách hàng

Age: 0

Gender: Male

Annual_Income: 0

Spending_Score: 0

Predict

Hình 5.19 Giao diện dự đoán một khách hàng của mô hình 1

5.1.2. Mô hình 2

Ngoài các nhu cầu đặc biệt về phân nhóm khách hàng, các trung tâm thương mại ngày nay đã thực dụng hơn, họ thường có xu hướng phân nhóm khách hàng theo thu nhập và chi tiêu cá nhân mà không quan tâm đến tuổi và giới tính. Điển hình là chương trình ưu đãi khi tích lũy khi mua hàng, đủ điểm sẽ nhận được phần quà nhất định, hoặc mức chiết khấu theo % cao hơn.

Do đó, ở phần này chúng tôi sẽ trình bày cách xây dựng một mô hình phân nhóm khách hàng với thuật toán K-Means, sau đó dùng Naïve Bayes xây dựng mô hình dự đoán nhóm khách hàng dựa trên 2 thuộc tính của tập dữ liệu: Annual Income (k\$), Spending Score (1-100)

Bước 1: Bắt đầu xây dựng mô hình: thêm các thư viện và module cần thiết.

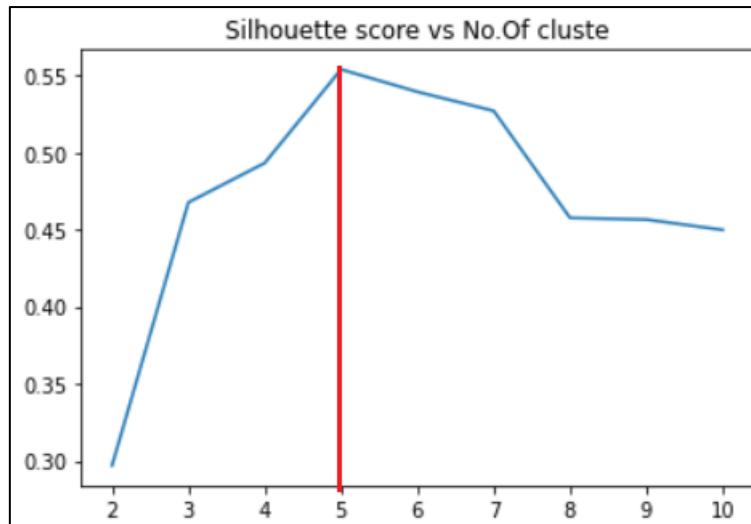
```
#import thư viện
import numpy as np
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
import plotly.express as px
```

Hình 5.20 Code thêm các thư viện, module cho quá trình phân nhóm ở mô hình 2

Bước 2: Tìm **K** cho mô hình K-Means: Đầu tiên tạo một mảng silhouette score có tên **sil_score**. Sau đó, ta cho số lượng cụm chạy từ 2 đến 10 và trực quan nó lên bằng module **pyplot** của thư viện **matplotlib**. Ta thấy được **K=5** cho silhouette score cao nhất.

```
#Tìm K phù hợp cho thuật toán K-mean
from sklearn.metrics import silhouette_score
sil_score=[]
for i in range(2,11):
    kmeans = KMeans(n_clusters=i, random_state=0)
    kmeans.fit(df)
    preds = kmeans.predict(df)
    score = silhouette_score(df, preds, metric='euclidean')
    sil_score.append(score)
plt.plot(range(2,11),sil_score)
plt.title("Silhouette score vs No.Of cluste")
plt.show()
```

Hình 5.21 Code tìm K cho thuật toán K-Means trong mô hình 2



Hình 5.22 Trực quan hóa silhouette score cho mô hình 2

Bước 3: Huấn luyện mô hình bằng lớp **Kmeans** thuộc module **sklearn.cluster** với tham số là **n_clusters = 5** (số cụm), **init = 'k-means++'** (khởi tạo) và **random_state = 42**. Hai thuộc tính được sử dụng là Annual Income (k\$) và Spending Score (1-100). Cũng như mô hình trên, khởi tạo bằng 'kmeans++' để tăng tốc độ hội tụ của các cụm.

```
#Huấn luyện mô hình
kmeans = KMeans(n_clusters= 5, init='k-means++', random_state=42)
kmeans.fit(df)
df['label']=kmeans.labels_
y_kmeans= kmeans.fit_predict(df)
```

Hình 5.23 Code huấn luyện mô hình phân nhóm bằng K-Means cho mô hình 2

Bước 4: Trực quan hóa dữ liệu đã phân cụm với module **matplotlib.pyplot**.

```
#Trực quan hóa các điểm dữ liệu trong 5 cụm
plt.figure(figsize=(14,7))
plt.title('Annual Income - Spending Score Segmentation')
plt.xlabel('Annual Income')
plt.ylabel('Spending Score')
colors=['C0','C1','C2','C3','C4']
for i in range(len(np.unique(y_kmeans))):
    plt.scatter(df.loc[y_kmeans == i, 'Annual Income (k$)'],
                df.loc[y_kmeans == i, 'Spending Score (1-100)'],
                s = 100, c=colors[i], label = 'Cluster {}'.format(i))
plt.scatter(kmeans.cluster_centers_[ :, 0],
            kmeans.cluster_centers_[ :, 1],
            s = 200, marker='x', c = 'black', linewidth=6, label = 'Centroids')

plt.legend()
plt.show()
```

Hình 5.24 Code trực quan hóa 5 cụm sau khi phân nhóm của mô hình 2

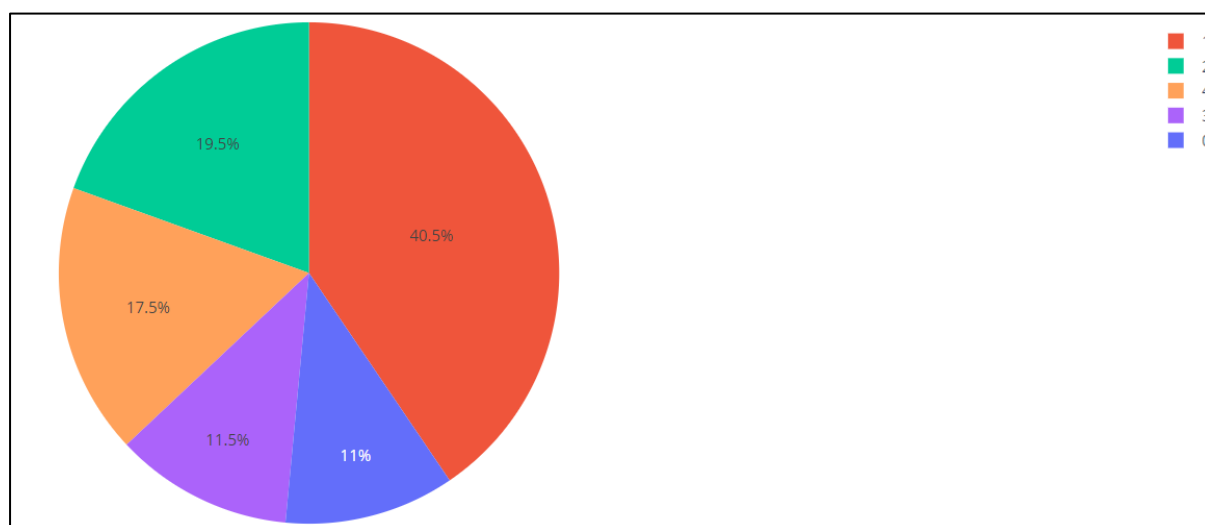


Hình 5.25 Trực quan hóa dữ liệu trong 5 cụm được phân của mô hình 2

Bước 5: Trực quan hóa sự phân bố 200 điểm dữ liệu vào các cụm bằng hàng **pie** của module **plotly.express**.

```
#Trực quan hóa sự phân bố các điểm dữ liệu vào 5 cụm
pie=df.groupby('label').size().reset_index()
pie.columns=['label','value']
px.pie(pie,values='value',names='label',color='label')
```

Hình 5.26 Code sự phân bố 200 điểm dữ liệu vào 5 cụm của mô hình 2



Hình 5.27 Sự phân bố 200 điểm dữ liệu vào 5 cụm của mô hình 2

Sau khi sử dụng mô hình 2 phân nhóm ta thu được 5 phân khúc khách hàng như sau:

STT	Cluster	Mô tả	Số lượng
1	0	Đây là nhóm khách hàng có thu nhập thấp (dưới 40 k\$) và điểm chi tiêu cao (60-100 điểm)	22
2	1	Đây là nhóm khách hàng có thu nhập trung bình (khoảng từ 40-dưới 80 k\$) và điểm chi tiêu trung bình (30-60 điểm)	81
3	2	Đây là nhóm khách hàng có thu nhập cao (60-140 k\$) và điểm chi tiêu cao (60-100 điểm)	39

4	3	Đây là nhóm khách hàng có thu nhập thấp (dưới 40 k\$) và điểm chi tiêu thấp (0-40 điểm)	23
5	4	Đây là nhóm khách hàng có thu nhập cao (trên 60-140 k\$) và điểm chi tiêu thấp (0-40 điểm)	35
Tổng			200

Bảng 5.3 Mô tả 5 cụm sau khi phân chia của mô hình 2

Thống kê dữ liệu trong các nhãn.

Label	Số điểm dữ liệu	Tỉ lệ (%)
0	22	11
1	81	40.5
2	39	19.5
3	23	11.5
4	35	17.5
Tổng	200	100

Bảng 5.4 Thống kê số điểm dữ liệu trong 5 cụm đã phân chia của mô hình 2

Bước 6: Bây giờ chúng ta sẽ xây dựng mô hình phân lớp dữ liệu với các nhãn vừa phân nhóm được bằng Naïve Bayes, đầu tiên ta thêm các thư viện cần thiết vào.

```
#Import các thư viện
from sklearn.naive_bayes import GaussianNB
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
import seaborn as sns
```

Hình 5.28 Code thêm các thư viện, module của mô hình 2

Bước 7: Kiểm tra lại tập dữ liệu **df** hiện tại.

	Gender	Age	Annual Income (k\$)	Spending Score (1-100)	label
0	100	19	15	39	8
1	100	21	15	81	6
2	50	20	16	6	10
3	50	23	16	77	7
4	50	31	17	40	10
...
195	50	35	120	79	5
196	50	45	126	28	9
197	100	32	126	74	1
198	100	32	137	18	2
199	100	30	137	83	1
200 rows × 5 columns					

Hình 5.29 Tập dữ liệu tại bước 7 của mô hình 2

Bước 8: Chia dữ liệu thành 2 tập train (X_train, Y_train) và test (X_test, Y_test) theo tỉ lệ 8:2 (tương đương tập train có 160 điểm dữ liệu và tập test có 40 điểm dữ liệu) bằng lớp **train_test_split** thuộc module **sklearn model_selection**.

```
#Chia thành tập train test theo tỉ lệ 8:2
X = df.iloc[:, 0:-1].values
Y = df.iloc[:, -1].values
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, train_size = 0.8,
                                                    random_state = 42)
```

Hình 5.30 Code chia tập dữ liệu thành train, test cho mô hình 2

Bước 9: Huấn luyện mô hình phân lớp cho mô hình 2 với lớp **GaussianNB** thuộc module **sklearn.naive_bayes**.

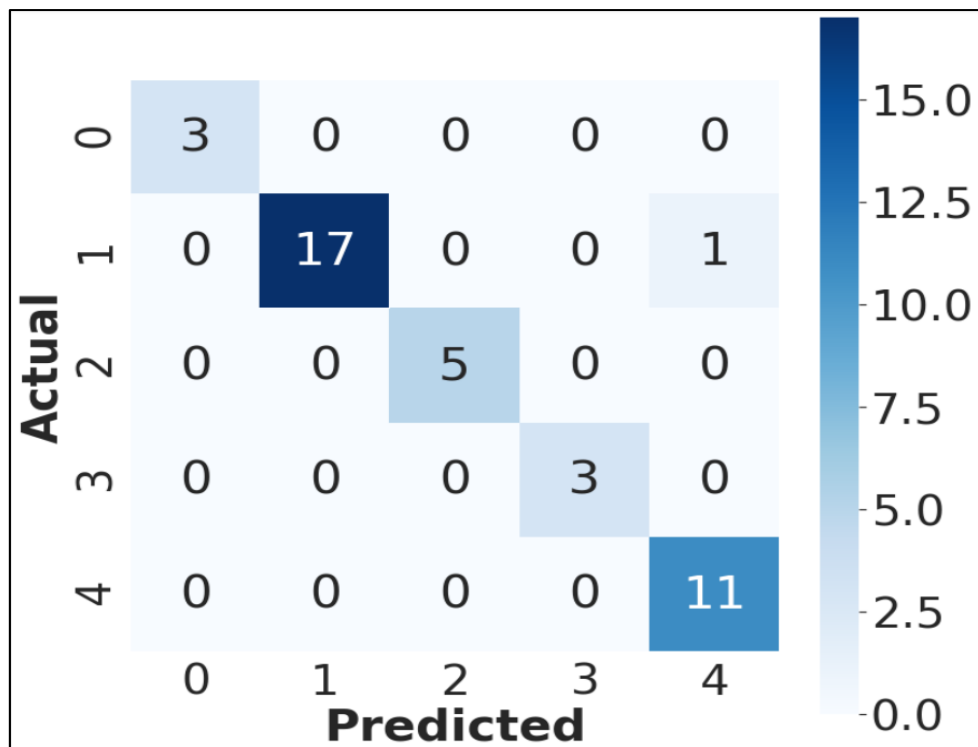
```
#Huấn luyện mô hình
GNB_classifier = GaussianNB()
GNB_classifier.fit(X_train, Y_train)
```

Hình 5.31 Code huấn luyện mô hình phân lớp bằng GaussianNB cho mô hình 2

Bước 10: Sử dụng thư viện **seaborn** và lớp **confusion_matrix** thuộc module **sklearn.metrics** để trực quan confusion matrix.

```
#In ra confusion matrix
cm = confusion_matrix(Y_test, GNB_classifier.predict(X_test))
plt.figure(figsize=(12, 12))
sns.set(font_scale=3.5)
ax = sns.heatmap(cm, cmap=plt.cm.Blues, annot=True, square=True, fmt = 'g')
ax.set_ylabel('Actual', fontsize=40, fontweight = 'bold')
ax.set_xlabel('Predicted', fontsize=40, fontweight = 'bold')
```

Hình 5.32 Code in ra confusion matrix cho mô hình 2



Hình 5.33 Confusion matrix của mô hình 2

Bước 11: Sử dụng hàm `classification_report` thuộc module `sklearn.metrics` để tính toán và in ra các độ đo đánh giá cần thiết cho một mô hình phân lớp.

```
#Classification report
print(classification_report(Y_test, GNB_classifier.predict(X_test)))
```

Hình 5.34 Code in ra classification report cho mô hình 2

	precision	recall	f1-score	support
0	1.00	1.00	1.00	3
1	1.00	0.94	0.97	18
2	1.00	1.00	1.00	5
3	1.00	1.00	1.00	3
4	0.92	1.00	0.96	11
accuracy			0.97	40
macro avg	0.98	0.99	0.99	40
weighted avg	0.98	0.97	0.98	40

Hình 5.35 Classification report của mô hình 2

Bước 12: Xây dựng một form và button ‘Predict’ để dự đoán một khách hàng. Cũng như mô hình trước đó, **Form** có sẵn ở **Google Colab**, còn button được thêm từ thư viện **ipywidgets** và module **IPython.display**. Chú ý, phải bật tính năng ‘Tự thực thi ô khi các trường thay đổi’ của **Form** để có thể luôn cập nhật các giá trị mới được nhập từ người dùng.

```
#@title Nhập các thuộc tính khách hàng { run: "auto" }
Annual_Income = 20#@param {type:"integer"}
Spending_Score = 18#@param {type:"integer"}

obj=np.array([Annual_Income, Spending_Score])

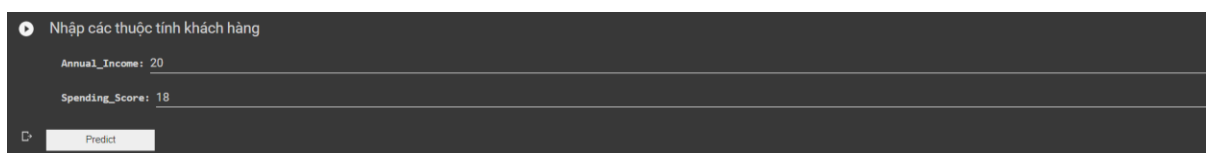
Message = 'Cluster ' + str(GNB_classifier.predict(obj.reshape(1, -1))[0])

#button 'Predict'
import ipywidgets as widgets
from IPython.display import display
button = widgets.Button(description="Predict")
output = widgets.Output()

def on_button_clicked(b):
    with output:
        print(Message)

button.on_click(on_button_clicked)
display(button, output)
```

Hình 5.36 Code tạo button và form để dự đoán khách hàng cho mô hình 2

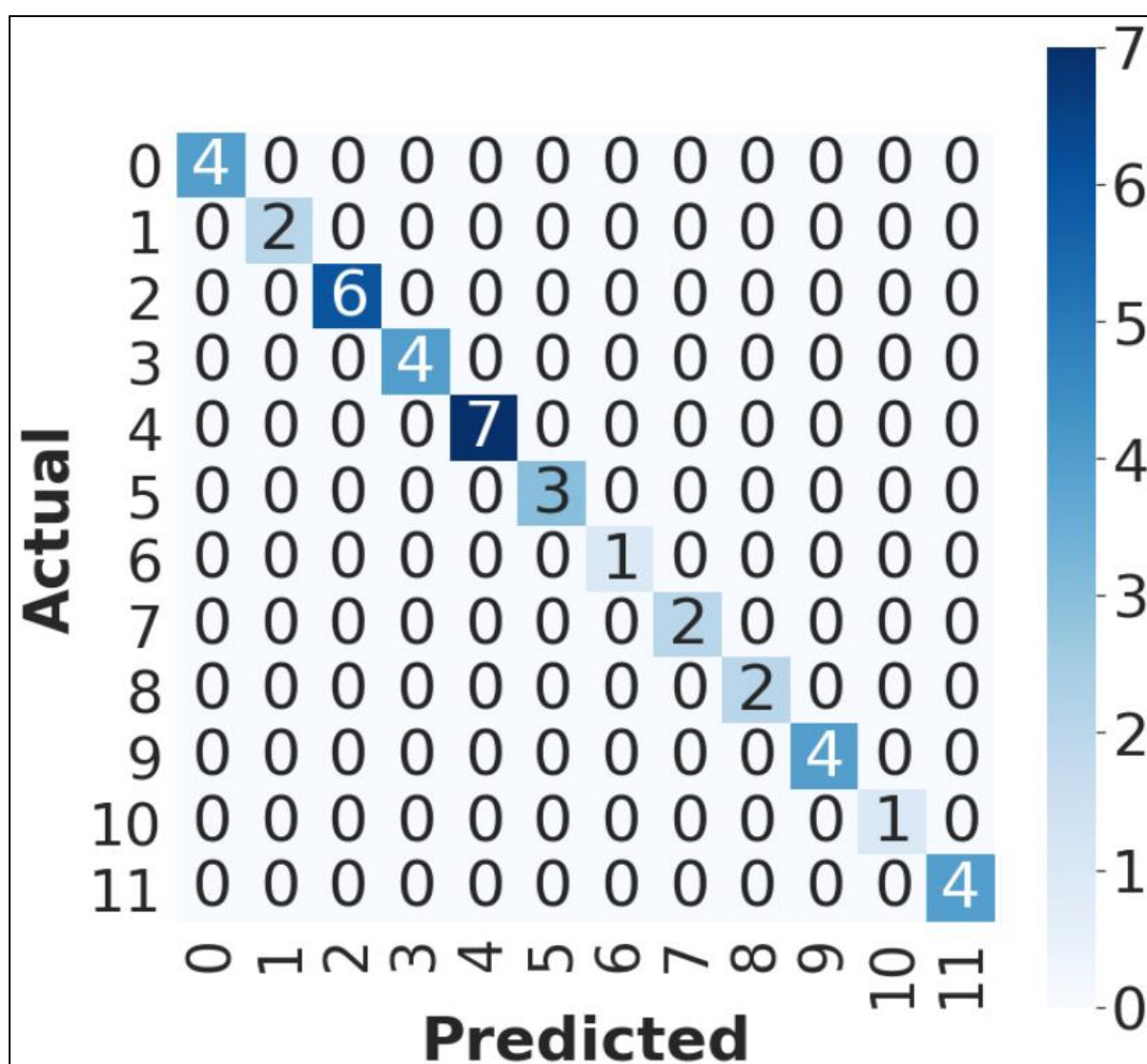
The screenshot shows a Google Colab notebook interface. At the top, there is a title bar that says "Nhập các thuộc tính khách hàng". Below the title bar, there are two input fields: "Annual_Income: 20" and "Spending_Score: 18". At the bottom of the form, there is a button labeled "Predict". The entire interface is displayed within a dark-themed window.

Hình 5.37 Giao diện dự đoán một khách hàng của mô hình 2

Chương 6. HIỆU SUẤT MÔ HÌNH

Các độ đo được sử dụng trong phần hiệu suất mô hình này lần là Accuracy, Precision, Recall và F1-score. Trong đó, độ đo Accuracy là độ đo đơn giản nhất, thường được sử dụng để đánh giá một mô hình; các độ đo Precision, Recall, F1-score sẽ đánh giá khách quan hơn cho mô hình được huấn luyện với bộ dữ liệu mất cân bằng, gây ra việc độ chính xác của các mẫu chiếm đa số gần như là độ chính xác của cả mô hình nếu chỉ xem xét với Accuracy. Vì vậy, nó phù hợp với 2 mô hình trong báo cáo này khi bộ dữ liệu được sử dụng bị mất cân bằng nhẹ.

6.1. Mô hình 1



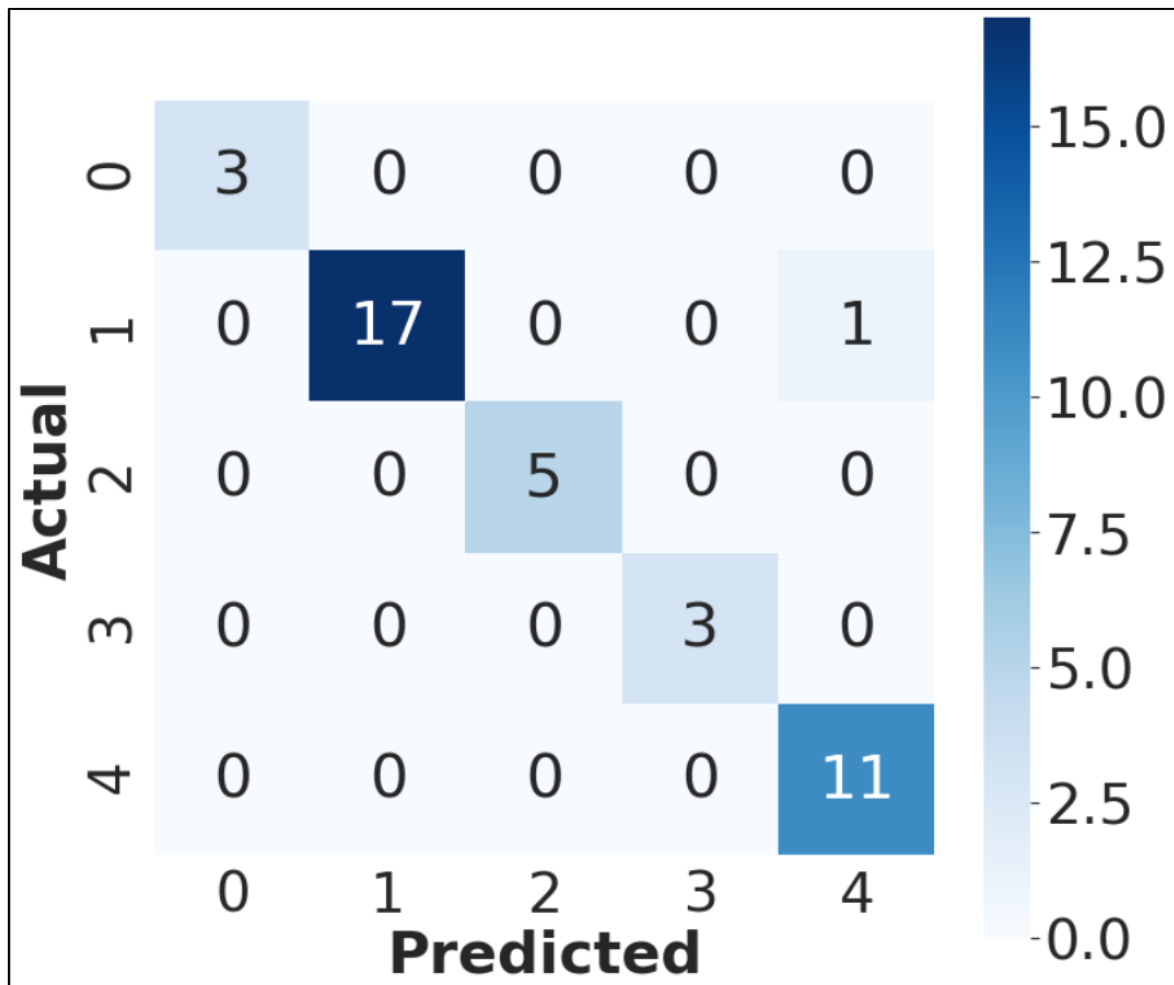
Hình 6.1 Confusion matrix cho mô hình 1

- Số điểm phân loại đúng là $4+2+6+4+7+3+1+2+2+4+1+4 = 40$.
- Số điểm phân loại sai là 0.
- Tỷ lệ điểm dữ liệu phân loại đúng là $40/40=100\%$.

Label	Precision	Recall	F1-score
0	1.00	1.00	1.00
1	1.00	1.00	1.00
2	1.00	1.00	1.00
3	1.00	1.00	1.00
4	1.00	1.00	1.00
5	1.00	1.00	1.00
6	1.00	1.00	1.00
7	1.00	1.00	1.00
8	1.00	1.00	1.00
9	1.00	1.00	1.00
10	1.00	1.00	1.00
11	1.00	1.00	1.00
Trung bình	1.00	1.00	1.00

***Bảng 6.1** Classification report mô hình 1*

6.2. Mô hình 2



Hình 6.2 Confusion matrix của mô hình 2

- Số điểm phân loại đúng là $3+17+5+3+11 = 39$.
- Số điểm phân loại sai là 1.
- Tỷ lệ điểm dữ liệu phân loại đúng là $39/40=97.5\%$.

Label	Precision	Recall	F1-score
0	1.00	1.00	1.00
1	1.00	0.94	0.97
2	1.00	1.00	1.00
3	1.00	1.00	1.00
4	0.92	1.00	0.96
Trung bình	0.984	0.988	0.986

Bảng 6.2 Classification report mô hình 2

Chương 7. KẾT LUẬN

Qua hiệu suất mô hình 1 ta thấy được kết quả vô cùng cao với độ chính xác là 100% với mọi độ đo Accuracy, Precision, Recall, F1-score. Một kết quả vô cùng tuyệt vời. Ở mô hình 2, kết quả cũng rất khả quan, độ chính xác lần lượt là 97.5% cho Accuracy, 98.4% cho Precision, 98.8% cho Recall, và 98.6% cho F1-score; các nhãn 0, 2, 3 cho độ chính xác theo F1-score là 100%, hai nhãn 1 và 4 cũng cho kết quả rất cao lần lượt là 97% và 96% theo độ đo F1-score. Kết quả cao như vậy là do nét tương đồng về mặt toán học khi quyết định một điểm dữ liệu thuộc cụm nào/lớp nào giữa Gaussian Naïve Bayes và K-Means sử dụng khoảng cách Euclide. Cụ thể là hai thuật toán đều sử dụng các bình phương hiệu giữa thuộc tính thứ i của điểm dữ liệu và giá trị kỳ vọng trung bình thuộc tính thứ i của tâm cụm để đánh giá cho điểm dữ liệu mới. *(Tham khảo thêm tại phần phụ lục[I])*

Qua bài báo cáo, chúng tôi đã trình bày được cách phân khúc khách hàng ở trung tâm thương mại bằng machine learning thông qua thuật toán K-Means. Sau đó, chúng tôi gán nhãn theo nhóm và tiến hành huấn luyện mô hình dự đoán khách hàng thuộc nhóm nào thông qua Gaussian Naïve Bayes. Thách thức thứ nhất trong cách phân khúc này là nhà kinh doanh phải hiểu được và tìm ra đặc điểm của các cụm, vì thuật toán K-Means chỉ tìm ra sự tương đồng của các khách hàng, sau đó gom những khách hàng đó lại. Thách thức thứ hai là số lượng điểm dữ liệu trong báo cáo này còn ít và mất cân bằng, vì vậy, trong tương lai để đánh giá mô hình này khách quan và chặt chẽ hơn ta sẽ cần một bộ dữ liệu tốt hơn.

Từ đó chúng ta thấy được, machine learning đã giúp ích rất nhiều cho con người trong phân khúc khách hàng như giảm thiểu thời gian, tối ưu độ chính xác và có thể tìm ra các ngoại lệ cho các phân khúc khách hàng, từ đó giúp cho nhà kinh doanh triển khai các chiến lược phù hợp với các đối tượng khách hàng (khách hàng tiềm năng, khách hàng thân thiết, khách hàng lợi nhuận không cao,...) để tối ưu được lợi nhuận và công sức. Không những trong phân khúc khách hàng mà còn trong rất nhiều lĩnh vực trong cuộc sống. Machine learning đang cho thấy sự cần thiết và tiềm năng của nó có thể mang lại cho con người. Đây là một lĩnh vực quan trọng và hứa hẹn sẽ làm được nhiều thứ hơn nữa cho tương lai. Tuy nhiên nó cũng đòi hỏi những con người sáng tạo,

nhảy bèn và kiên trì để nghiên cứu, phát triển và ứng dụng nó vào cuộc sống hằng ngày của chúng ta.

TÀI LIỆU THAM KHẢO

- [1] “1.9. Naive bayes”. [Trực tuyến]. Địa chỉ: https://scikit-learn.org/stable/modules/naive_bayes.html. [Truy cập lần cuối 27/06/2021].
- [2] A.C. Bepary, Z. Ferdous. “*Customer Segmentation by Using Machine Learning and E-commerce Solution*”, 2020. [Trực tuyến]. Địa chỉ: <http://dspace.daffodilvarsity.edu.bd:8080/handle/123456789/5319>. [Truy cập lần cuối 25/06/2021].
- [3] A. Chaerudin, D.T. Murdiansyah, M. Imrona. “*Implementation of K-Means++ Algorithm for Store Customers Segmentation Using Neo4J*”. [Trực tuyến]. Địa chỉ: <https://socj.telkomuniversity.ac.id/ojs/index.php/indojc/article/view/547>. [Truy cập lần cuối 25/06/2021].
- [4] “*Bài 4: K-Means Clustering*”, 2017. [Trực tuyến]. Địa chỉ: <https://machinelearningcoban.com/2017/01/01/kmeans/?fbclid=IwAR07ZODtK6pBv5kGJoSltrdyd9D3RleD1l2-4sjrBtrmNALgSbMyjYZj-bc>. [Truy cập lần cuối 25/06/2021].
- [5] “*Bài 31: Maximum Likelihood và Maximum A Posteriori estimation*”, [Trực tuyến]. 2017. Địa chỉ: <https://machinelearningcoban.com/2017/07/17/mlemap/>. [Truy cập lần cuối 25/06/2021].
- [6] “*Bài 32: Naive Bayes Classifier*”, 2017. [Trực tuyến]. Địa chỉ: <https://machinelearningcoban.com/2017/08/08/nbc/>. [Truy cập lần cuối 25/06/2021].
- [7] “*Bài 33: Phương pháp đánh giá một hệ thống phân lớp*”, 2018. [Trực tuyến]. Địa chỉ: https://machinelearningcoban.com/2017/08/31/evaluation/?fbclid=IwAR35mrLi1x9c5_xtGO_Q9eXSmQmzGdcwFUmcEe-z8DV9JDZSKhpWKdyjeCo. [Truy cập lần cuối 25/06/2021].
- [8] “*Determining The Optimal Number Of Clusters: 3 Must Know Methods*”. [Trực tuyến]. Địa chỉ: <https://www.datanovia.com/en/lessons/determining-the-optimal-number-of-clusters-3-must-know->

- [methods/?fbclid=IwAR0Sjj8EhAY5xbDLuXQsL8C7zwKbP0E0csUKjFXOIFlBzhisqGiKNGLvwU#elbow-method](https://colab.research.google.com/notebooks/forms.ipynb#scrollTo=3jKM6GfzlgpS). [Truy cập lần cuối 25/06/2021].
- [9] D. Pham. “*Phân khúc khách hàng để Marketing hiệu quả*”, 2017. [Trực tuyến]. Địa chỉ: https://subiz.com.vn/blog/phan-khuc-khach-hang.html?fbclid=IwAR0GyUY18p51Z4a6f2iStAGLR_dSmwECsxlbyPiezD1vA84ufyz4WYfz9dw. [Truy cập lần cuối 25/06/2021].
- [10] Đ.T. Minh, L.V.L. Oanh. “*Phân khúc khách hàng mua sắm dựa trên thuộc tính của các trung tâm thương mại tại Thành phố Hồ Chí Minh*”, 2021. [Trực tuyến]. Địa chỉ: http://tapchicongthuong.vn/bai-viet/phan-khuc-khach-hang-mua-sam-dua-tren-thuoc-tinh-cua-cac-trung-tam-thuong-mai-tai-thanh-pho-ho-chi-minh-78672.htm?fbclid=IwAR1mI3CLp8UJ_E1glT2p6W1EAewzqtuRIIdS34_5gA0j51rhrWiVdOMnA3Qo. [Truy cập lần cuối 25/06/2021].
- [11] “*Forms*”. [Trực tuyến]. Địa chỉ: <https://colab.research.google.com/notebooks/forms.ipynb#scrollTo=3jKM6GfzlgpS>. [Truy cập lần cuối 26/06/2021].
- [12] “*Google Colab - Adding Forms*”. [Trực tuyến]. Địa chỉ: https://www.tutorialspoint.com/google_colab/google_colab_adding_forms.htm. [Truy cập lần cuối 26/06/2021].
- [13] J.R. Johansson. “*matplotlib - 2D and 3D plotting in Python*”, 2016. [Trực tuyến]. Địa chỉ: https://www.southampton.ac.uk/~feeeg1001/notebooks/Matplotlib.html?fbclid=IwAR0GyUY18p51Z4a6f2iStAGLR_dSmwECsxlbyPiezD1vA84ufyz4WYfz9dw. [Truy cập lần cuối 25/06/2021].
- [14] M.G. Pradana, H.T. Ha. “*Maximizing Strategy Improvement In Mall Customer Segmentation Using K-Means Clustering*”, 2021. [Trực tuyến]. Địa chỉ: <http://bright-journal.org/Journal/index.php/JADS/article/view/18>. [Truy cập lần cuối 26/06/2021].
- [15] M. Letelier. “*Clustering With More Than Two Features? Try This To Explain Your Findings*”, 2020. [Trực tuyến]. Địa chỉ: <https://towardsdatascience.com/clustering-with-more-than-two-features-try->

- [this-to-explain-your-findings-b053007d680a](#). [Truy cập lần cuối 26/06/2021].
- [16] Nathan. “*Xây Dựng Clustering Model Bằng Giải Thuật K-Means Với Thư Viện Scikit-Learn Skills AI*”, 2020. [Trực tuyến]. Địa chỉ: <https://insights.magestore.com/posts/xay-dung-clustering-model-bang-giai-thuat-k-means-voi-thu-vien-scikit-learn-skills-ai?fbclid=IwAR0WXr53T3ZfUvmnLCYyeLZIJLosxtdycPxSExWBg8QJo3BNiy-A5Icp6HQ>. [Truy cập lần cuối 25/06/2021].
- [17] N. Singh, P.Singh, K.K. Singh, A. Singh. “*Machine learning based classification and segmentation techniques for CRM: a customer analytics*”, 2020. [Trực tuyến]. Địa chỉ: <https://www.inderscienceonline.com/doi/abs/10.1504/IJBFMI.2020.109878>. [Truy cập lần cuối 25/06/2021].
- [18] N.V. Hiếu. “*Thuật toán K-Means (K-Means clustering) và ví dụ*”. [Trực tuyến]. Địa chỉ: https://nguyenvanhieu.vn/thuat-toan-phan-cum-k-means/?fbclid=IwAR1NlrcEuikFBQCpcu_dtNmjkRFULtO9Mqcw2BCBZxbitxEfYvIEH65UhxI#gioi-thieu-ve-k-means. [Truy cập lần cuối 25/06/2021].
- [19] “sklearn.cluster.Kmeans”. [Trực tuyến]. Địa chỉ: <https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html>. [Truy cập lần cuối 27/06/2021].
- [20] S. Kumar. “*Silhouette Method — Better than Elbow Method to find Optimal Clusters*”, 2020. [Trực tuyến]. Địa chỉ: <https://towardsdatascience.com/silhouette-method-better-than-elbow-method-to-find-optimal-clusters-378d62ff6891>. [Truy cập lần cuối 25/06/2021].
- [21] T.D. Tan. “*Sơ lược về Maximum Likelihood Estimation*”, 2019. [Trực tuyến]. Địa chỉ: <https://viblo.asia/p/so-luoc-ve-maximum-likelihood-estimation-1Je5EvrYKnL>. [Truy cập lần cuối 27/06/2021].
- [22] T.T. Hmwe, N.Y.T. Thein, K.M. Cho. “*Improving Clustering Quality Using Silhouette Score*”, 2020. [Trực tuyến]. Địa chỉ:

https://www.ucstgi.edu.mm/wp-content/uploads/2020/10/JCAR2020_58_62.pdf. [Truy cập lần cuối 25/06/2021].

- [23] X. Hồng. “*PHÂN KHÚC KHÁCH HÀNG THỜI ĐẠI CÔNG NGHỆ SỐ 4.0*”, 2019. [Trực tuyến]. Địa chỉ: <https://gemdigital.vn/phan-khuc-khach-hang/?fbclid=IwAR0MnFC0HXyASGVyb2THziNjTpQgFPIiKS5F6osUOMOdMcnOwp-3WEDCA34>. [Truy cập lần cuối 25/06/2021].

PHỤ LỤC

I. Nét tương đồng về mặt toán học khi quyết định một điểm dữ liệu thuộc cụm nào/lớp nào giữa K-Means sử dụng khoảng cách Euclide và Gaussian Naïve Bayes

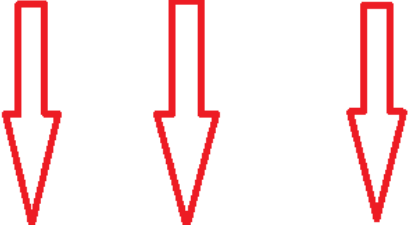
Ta thấy được các giá trị thuộc tính của một centroid chính là giá trị kỳ vọng trung bình của các thuộc tính. Giả sử ta có:

- Cluster **A** có centroid **B** $(\mu_1, \mu_2, \dots, \mu_d)$.
- Điểm dữ liệu mới **x** (x_1, x_2, \dots, x_d) .
- **p(x)** là xác suất tìm được x trong tập dữ liệu.
- Lớp **A** với **p(A)** là xác suất một điểm rơi vào lớp A và kỳ vọng, phương sai của thuộc tính thứ **i** là μ_i, σ_i^2 .

Khoảng cách từ điểm dữ liệu x đến centroid B:

$$(x_1 - \mu_1)^2 + (x_2 - \mu_2)^2 + \dots + (x_d - \mu_d)^2$$

Xác suất lớp A là lớp của x theo Gaussian Naïve Bayes:

$$\begin{aligned} & \frac{p(A)}{p(x)} \times p(x | A) \\ &= \frac{p(A)}{p(x)} \times \prod_{i=1}^d p(x_i | c) \\ &= \frac{p(A)}{p(x)} \times \left[\frac{1}{\sqrt{2\pi\sigma_1^2}} \times \frac{1}{\sqrt{2\pi\sigma_2^2}} \times \dots \times \frac{1}{\sqrt{2\pi\sigma_d^2}} \right] \times e^{-\left[\frac{(x_1 - \mu_1)^2}{2\sigma_1^2} + \frac{(x_2 - \mu_2)^2}{2\sigma_2^2} + \dots + \frac{(x_d - \mu_d)^2}{2\sigma_d^2} \right]} \end{aligned}$$


Kết luận: Ta thấy khi chọn một lớp/một cụm cho điểm dữ liệu mới giữa K-Means và Gaussian Naïve Bayes có nét tương đồng ở chỗ là đều sử dụng các bình phương hiệu của thuộc tính thứ **i** của **x** và thuộc tính thứ **i** centroid **B**. Và ta thấy hàm xác suất sẽ giảm khi các bình phương hiệu này tăng với cùng phương sai và **p(A)**. **Vì vậy,**

khoảng cách một điểm dữ liệu và một tâm càng nhỏ thì khả năng cao điểm dữ liệu này sẽ thuộc cụm này. Nhưng vẫn có ngoại lệ cho trường hợp một điểm dữ liệu gần một tâm khác hơn nhưng vẫn được mô hình Gaussian Naïve Bayes dự đoán thuộc cụm này, lý do là biểu thức xác suất không chỉ phụ thuộc vào khoảng cách giữa điểm dữ liệu này và các tâm cụm mà còn phụ thuộc vào phương sai trung bình (khoảng cách trung bình giữa các điểm dữ liệu trong cụm với tâm cụm đó) và $\mathbf{p}(\mathbf{A})$ (xác suất một điểm bất kỳ thuộc cụm \mathbf{A}) của các cụm. Minh chứng ngoại lệ này là mô hình 2 vẫn có điểm dữ liệu bị dự đoán sai.