Họ và tên: Nguyễn Vũ Khánh Huy
MSSV: 16025591

Đề tài: Ứng dụng cây quyết định để xây dựng mô hình phân loại dữ liệu. Đánh giá, so sánh các mô hình và áp dụng mô hình để dự báo trên tập dữ liệu. Viết ứng dụng minh họa.

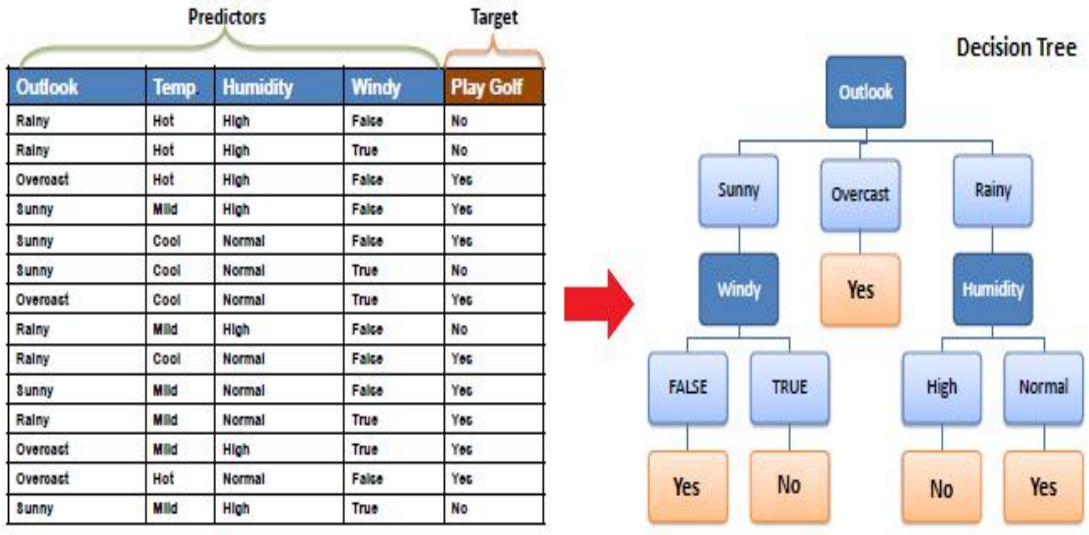
Table of contents

Decision tree (cây quyết định)

1.Giới thiệu sơ lược

Cây quyết định thuộc dạng **supervised learning** (máy học có giám sát) có thể được áp dụng vào cả bài toán classification và hồi quy. Việc xây dựng một cây quyết định trên dữ liệu huấn luyện cho trước là việc đi xác định các câu hỏi và thứ tự của chúng. Một điểm đáng lưu ý của decision tree là nó có thể làm việc với các đặc trưng (trong các tài liệu về decision tree, các đặc trưng thường được gọi là thuộc tính - attribute) dạng categorial, thường là rời rạc và không có thứ tự. Ví dụ, mưa nắng hay xanh đỏ, ... Cây quyết định cũng làm việc với dữ liệu có vector đặc trưng bao gồm cả thuộc tính dạng categorial và liên tục (numeric). Một điểm đáng chú ý nữa là cây quyết định ít yêu cầu việc chuẩn hoá dữ liệu.

Cây quyết định được sử dụng cho các bài toán **phân loại dữ liệu**. Dễ xây dựng cây và dễ hiểu là thế mạnh của cách tiếp cận bài toán bằng cây quyết định. Một cây quyết định bao gồm **nodes** (điểm trên cây), **branches** (nhánh) và l**eaf nodes** (node lá). Mỗi node là một đại diện cho một phép thử **logic** hay **toán học** trên từng thuộc tính trong tập dữ liệu. Mục tiêu cần đạt được là phân tách tập dữ liệu một cách rõ ràng để chỉ ra được sự liên quan giữa các biến số. Kết quả của từng phép thử chính là hướng đi của từng node. Node cha có thể có hai hoặc nhiều node con, tuỳ thuộc vào thuật toán đã chọn. Node cha và các node con được liên kết với nhau thông qua các nhánh, mỗi nhánh là đại diện cho kết quả của mỗi phép thử ở node cha. Node lá thì không có node con và chính là đại diện cho một class.



Hình 1.1 Ví dụ xây dựng cây quyết định

2.Mục tiêu.

Bài luận văn này tập trung vào việc tìm hiểu các bài toán trong thực tế và áp dụng cây quyết định để phân loại các tập dữ liệu sau đó trả về kết quả phù hợp nhất. Đặc điểm cấu tạo của cây quyết định giúp truyền tải ý tưởng từ bài toán vào thuật toán một cách tự nhiên nhất, không những vậy cây quyết định thường xuyên được sử dụng vào các bài toán phân loại trong thực tế như kinh tế, tài chính, y tế, nông nghiệp, sinh học.

3. Khai phá dữ liệu.

3.1.Khái niệm khai phá dữ liệu.

Khai phá dữ liệu (Data Mining) là một khái niệm ra đời vào những năm cuối của thập kỷ 1980. Nó là quá trình khám phá thông tin ẩn được tìm thấy trong các cơ sở dữ liệu và có thể xem như là một bước trong quá trình khám phá tri thức. Data Mining là giai đoạn quan trọng nhất trong tiến trình khai phá tri thức từ cơ sở dữ liệu, các tri thức này hỗ trợ trong việc ra quyết định trong khoa học và kinh doanh, ...

Giáo sư Tom Mitchell đã đưa ra định nghĩa của Khai phá dữ liệu như sau: “Khai phá dữ liệu là việc sử dụng dữ liệu lịch sử để khám phá những qui tắc và cải thiện những quyết định trong tương lai.” Với một cách tiếp cận ứng dụng hơn, Tiến sĩ Fayyad đã phát biểu: “Khai phá dữ liệu, thường được xem là việc khám phá tri thức trong các cơ sở dữ liệu, là một quá trình trích xuất những thông tin ẩn, trước đây chưa biết và có khả năng hữu ích, dưới dạng các qui luật, ràng buộc, qui tắc trong cơ sở dữ liệu.” hay nói cách khác “Khai phá dữ liệu – Data Mining là tiến trình khám phá tri thức tiềm ẩn trong các cơ sở dữ liệu. Cụ thể hơn, đó là tiến trình trích lọc, sản sinh những tri thức hoặc các mẫu tiềm ẩn, chưa biết nhưng hữu ích từ cơ sở dữ liệu lớn” .

Nói tóm lại, Khai phá dữ liệu là một quá trình học tri thức mới từ những dữ liệu đã thu thập được.

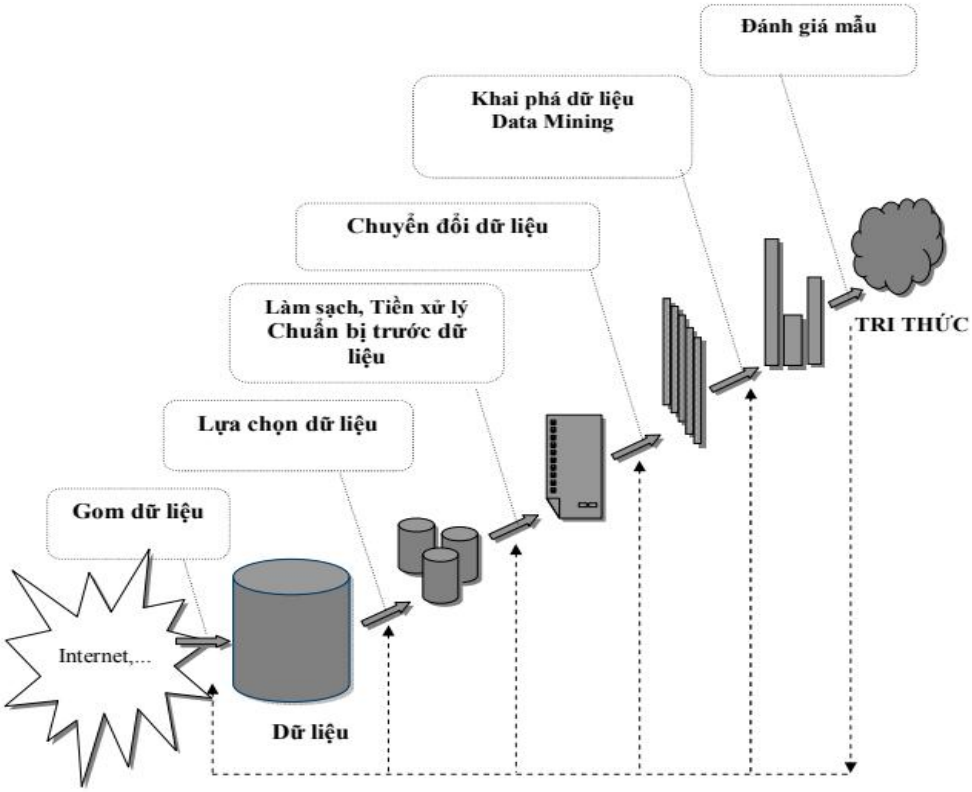
Khai phá dữ liệu là tiến trình khái quát các sự kiện rời rạc trong dữ liệu thành các tri thức mang tính khái quát, tính quy luật hỗ trợ tích cực cho các tiến trình ra quyết định. Khai phá dữ liệu là việc trích rút tri thức một cách tự động và hiệu quả từ một khối dữ liệu rất lớn. Tri thức đó thường ở dạng các mẫu tin có tính chất không tầm thường, không tường minh (ẩn), chưa được biết đến và có tiềm năng mang lại lợi ích.

Để hình dung vấn đề này ta có thể sử dụng một ví dụ đơn giản như sau: Khai phá dữ liệu được ví như tìm một cây kim trong đống cỏ khô. Trong ví dụ này, cây kim là một mảnh nhỏ tri thức hoặc một thông tin có giá trị và đống cỏ khô là một kho cơ sở dữ liệu rộng lớn. Như vậy, những thông tin có giá trị tiềm ẩn trong kho cơ sở dữ liệu sẽ được chiết xuất ra và sử dụng một cách hữu ích nhờ khai phá dữ liệu.

Chức năng khai phá dữ liệu gồm có gộp nhóm phân loại, dự báo, dự đoán và phân tích các liên kết. Năm 1989, Fayyad, Smyth và Piatestsky-Shapiro đã dùng khái niệm Phát hiện tri thức từ cơ sở dữ liệu (Knowledge Discovery in Database-KDD).

Trong đó, khai phá dữ liệu là một giai đoạn rất đặc biệt trong toàn bộ quá trình, nó sử dụng các kỹ thuật để tìm ra các mẫu từ dữ liệu. Có thể coi khai phá dữ liệu là cốt lõi của quá trình phát hiện tri thức.

Quá trình khai phá dữ liệu sẽ tiến hành qua 6 giai đoạn như **hình** sau



Bắt đầu của quá trình là kho dữ liệu thô và kết thúc với tri thức được chiết xuất

ra. Về lý thuyết thì có vẽ rất đơn giản nhưng thực sự đây là một quá trình rất khó khăn gặp phải rất nhiều vướng mắc như: quản lý các tập dữ liệu, phải lặp đi lặp lại toàn bộ quá trình, ...

1. Gom dữ liệu (Gathering): Tập hợp dữ liệu là bước đầu tiên trong quá trình khai phá dữ liệu. Đây là bước được khai thác trong một cơ sở dữ liệu, một kho dữ liệu và thậm chí các dữ liệu từ các nguồn ứng dụng Web.

2. Trích lọc dữ liệu (Selection): Ở giai đoạn này dữ liệu được lựa chọn hoặc phân
chia theo một số tiêu chuẩn nào đó, ví dụ chọn tất cả những người có tuổi đời từ 25 – 35 và có trình độ đại học.

3. Làm sạch, tiền xử lý và chuẩn bị trước dữ liệu (Cleaning, Pre-processing and Preparation): Giai đoan thứ ba này là giai đoạn hay bị sao lãng, nhưng thực tế nó là một bước rất quan trọng trong quá trình khai phá dữ liệu. Một số lỗi thường mắc phải trong khi gom dữ liệu là tính không đủ chặt chẽ, logíc. Vì vậy, dữ liệu thường chứa các giá trị vô nghĩa và không có khả năng kết nối dữ liệu. Ví dụ: tuổi = 273. Giai đoạn này sẽ tiến hành xử lý những dạng dữ liệu không chặt chẽ nói trên. Những dữ liệu dạng này được xem như thông tin dư thừa, không có giá trị. Bởi vậy, đây là một quá trình rất quan trọng vì dữ liệu này nếu không được “làm sạch - tiền xử lý - chuẩn bị trước” thì sẽ gây nên những kết quả sai lệch nghiêm trọng.

4. Chuyển đổi dữ liệu (Transformation): Tiếp theo là giai đoạn chuyển đổi dữ
liệu, dữ liệu đưa ra có thể sử dụng và điều khiển được bởi việc tổ chức lại nó. Dữ liệu đã được chuyển đổi phù hợp với mục đích khai thác.

5. Phát hiện và trích mẫu dữ liệu (Pattern Extraction and Discovery): Đây là bước mang tính tư duy trong khai phá dữ liệu. Ở giai đoạn này nhiều thuật toánkhác nhau đã được sử dụng để trích ra các mẫu từ dữ liệu. Thuật toán thường dùng là nguyên tắc phân loại, nguyên tắc kết hợp hoặc các mô hình dữ liệu tuần tự, ...

6. Đánh giá kết quả mẫu (Evaluation of Result): Đây là giai đoạn cuối trong quá
trình khai phá dữ liệu. Ở giai đoạn này, các mẫu dữ liệu được chiết xuất ra bởi phần mềm khai phá dữ liệu. Không phải bất cứ mẫu dữ liệu nào cũng đều hữu ích, đôi khi nó còn bị sai lệch. Vì vậy, cần phải ưu tiên những tiêu chuẩn đánh giá để chiết xuất ra các tri thức (Knowledge).

Trên đây là 6 giai đoạn trong quá trình khai phá dữ liệu, trong đó **giai đoạn 5** là giai đoạn được quan tâm nhiều nhất, đó là khai phá dữ liệu.

3.2.Kiến trúc của một hệ thống khai phá dữ liệu.

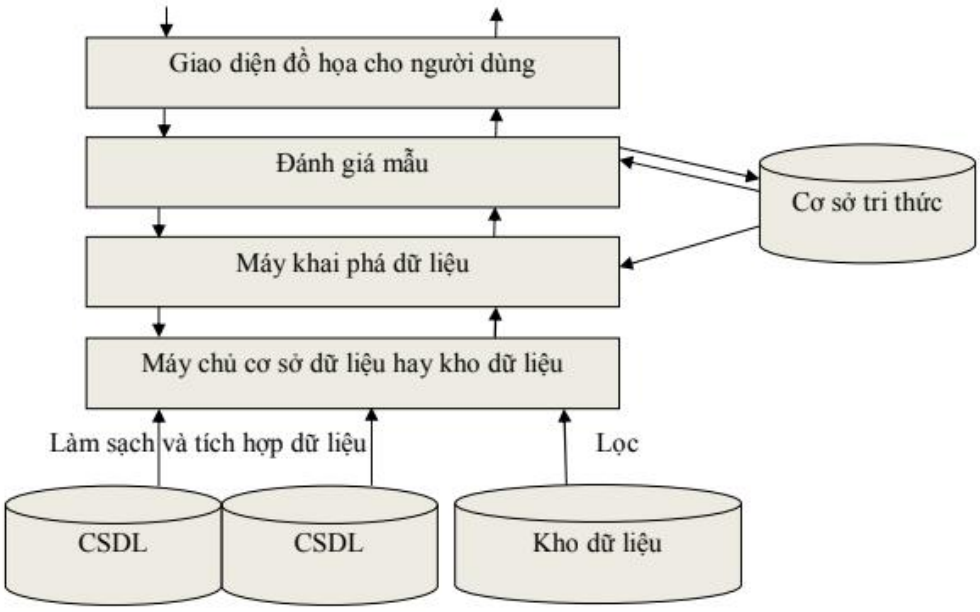
- Máy chủ cơ sở dữ liệu hay máy chủ kho dữ liệu (Database or warehouse server): Máy chủ này có trách nhiệm lấy dữ liệu thích hợp dựa trên những yêu cầu khai phá của người dùng.

- Cơ sở tri thức (Knowledge base): Đây là miền tri thức được dùng để tìm kiếm hay đánh giá độ quan trọng của các hình mẫu kết quả.

- Máy khai phá dữ liệu (Data mining engine): Một hệ thống khai phá dữ liệu cần phải có một tập các modun chức năng để thực hiện công việc, chẳng hạn như đặc trưng hóa, kết hợp, phân lớp, phân cụm, phân tích sự tiến hoá...

- Modun đánh giá mẫu (Pattern evaluation): Bộ phận này tương tác với các modun khai phá dữ liệu để tập trung vào việc duyệt tìm các mẫu đáng được quan tâm. Cũng có thể modun đánh giá mâu được tích hợp vào modun khai phá tuỳ theo sự cài đặt của phương pháp khai phá được dùng.

- Giao diện đồ họa cho người dùng (Graphical user interface): Thông qua giao diện này, người dùng tương tác với hệ thống bằng cách đặc tả một yêu cầu khai phá hay một nhiệm vụ, cung cấp thông tin trợ giúp cho việc tìm kiếm và thực hiện khai phá thăm dò trên các kết quả khai phá trung gian.



Hình trên là kiến trúc của một hệ thống khai phá dữ liệu

3.3.Một số kỹ thuật khai phá dữ liệu.

Các kĩ thuật khai phá dữ liệu thường được chia thành 2 nhóm chính:

- Kĩ thuật khai phá dữ liệu mô tả: có nhiệm vụ mô tả về các tính chất hoặc các đặc tính chung của dữ liệu trong CSDL hiện có. Các kĩ thuật này gồm có: phân cụm (clustering), tóm tắt (summarization), trực quan hóa (visualization), phân tích sự phát triển và độ lệch (Evolution and deviation analysis), phát hiện luật kết hợp (association rules), ...
- Kĩ thuật khai phá dữ liệu dự đoán: có nhiệm vụ đưa ra các dự đoán dựa vào các suy diễn trên dữ liệu hiện thời. Các kĩ thuật này gồm có: phân lớp (classification), hồi quy (regression), ...

Tuy nhiên, do khuôn khổ có hạn nên tôi chỉ giới thiệu 3 phương pháp thông dụng nhất là: phân lớp dữ liệu, phân cụm dữ liệu và khai phá luật kết hợp.

3.3.1 Phân lớp.

Phân lớp dữ liệu (Classification) là chia các đối tượng dữ liệu thành các lớp dựa trên các đặc trưng của tập dữ liệu. Với một tập các dữ liệu huấn luyện cho trước và sự huấn luyện của con người, các giải thuật phân loại sẽ học ra bộ phân loại (classifier) dùng để phân các dữ liệu mới vào một trong những lớp (còn gọi là loại) đã được xác định trước. Phương pháp này rất có ích trong giai đoạn đầu của quá trình nghiên cứu khi ta biết rất ít về đối tượng cần nghiên cứu, nó là tiền đề để tiến hành các phương pháp phát hiện tri thức. Có nhiều phương pháp phân lớp: phân lớp dựa trên cây quyết định, phân lớp Bayesian, ... Quá trình phân lớp dữ liệu thường gồm hai bước:

Bước 1: Xây dựng mô hình dựa trên việc phân tích các mẫu dữ liệu có sẵn. Mỗi mẫu tương ứng với một lớp, được quyết định bởi một thuộc tính gọi là thuộc tính phân lớp. Các mẫu dữ liệu này còn được gọi là tập dữ liệu huấn luyện (training dataset). Nhãn lớp của tập dữ liệu huấn luyện phải được xác định trước khi xây dựng mô hình, vì vậy phương pháp này còn được gọi là học có giám sát (supervised learning).

Bước 2: Sử dụng mô hình để phân lớp dữ liệu. Chúng ta phải tính độ chính xác của mô hình, nếu độ chính xác là chấp nhận được thì mô hình sẽ được sử dụng để dự đoán lớp cho các mẫu dữ liệu khác trong tương lai.

3.3.2 Phân cụm.

Phân cụm (Clustering) là việc nhóm các đối tượng dữ liệu thành các lớp đối tượng có sự tương tự nhau dựa trên các thuộc tính của chúng. Mỗi lớp đối tượng được gọi là một cụm (cluster). Một cụm bao gồm các đối tượng mà giữa bản thân chúng có sự ràng buộc lẫn nhau và khác biệt so với các lớp đối tượng khác. Phân cụm dữ liệu là một ví dụ của phương pháp học không có giám sát (unsupervised learning). Phân cụm dữ liệu không đòi hỏi phải định nghĩa trước các mẫu dữ liệu huấn luyện. Vì thế, có thể coi phân cụm dữ liệu là một cách học bằng quan sát (learning by observation), trong khi phân lớp dữ liệu là học qua ví dụ (learning by example). Trong phương pháp này ta không thể biết kết quả các cụm thu được sẽ như thế nào khi bắt đầu quá trình. Các cụm có thể tách riêng hay phân cấp hoặc gối lên nhau, có nghĩa là một mục dữ liệu có thể vừa thuộc cụm này vừa thuộc cụm kia. Vì vậy, thông thường cần có một chuyên gia về lĩnh vực đó để đánh giá các cụm thu được.

Phân cụm dữ liệu được sử dụng nhiều trong các ứng dụng về phân loại thị trường, phân loại khách hàng, nhận dạng mẫu, phân loại trang Web, ... Ngoài ra, phân cụm còn được sử dụng như một bước tiền xử lý cho các thuật toán khai phá dữ liệu khác.

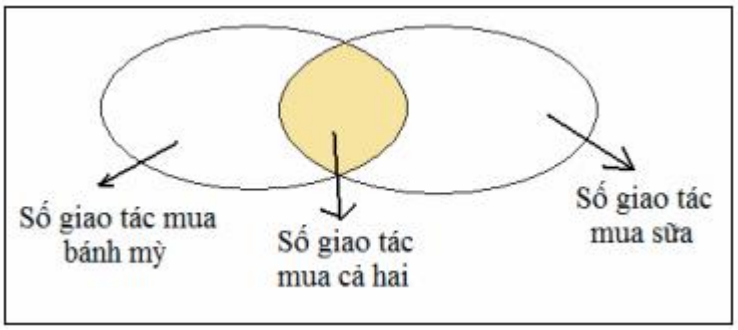
3.3.3 Luật kết hợp.

Phương pháp phát hiện các luật kết hợp (Association Rules) nhằm phát hiện ra các luật kết hợp giữa các thành phần dữ liệu trong cơ sở dữ liệu [4]. Các giải thuật Tìm luật liên kết tìm kiếm các mối liên kết giữa các phần tử dữ liệu, ví dụ như nhóm cácmón hàng thường được mua kèm với nhau trong siêu thị. Đầu ra của thuật toán là tập luật kết hợp tìm được. Cho trước một tập các giao tác, trong đó mỗi giao tác là một tập các mục, tìm sự tương quan giữa các mục như là một luật và kết quả của giải thuật khai phá dữ liệu là tập luật kết hợp tìm được. Luật kết hợp thường có dạng X => Y. Trong đó: X là tiền đề, Y là hệ quả (X, Y là hai tập của mục). Ý nghĩa trực quan của luật là các giao tác của cơ sở dữ liệu mà trong đó nội dung X có khuynh hướng đến nội dung Y.

Có hai thông số quan trọng của luật kết hợp là độ hỗ trợ (support) và độ tin cậy (confidence). Độ hỗ trợ và độ tin cậy là hai độ đo của sự đáng quan tâm của luật. Chúng tương ứng phản ánh sự hữu ích và sự chắc chắn của luật đã khám phá. Khai phá các luật kết hợp từ cơ sở dữ liệu là việc tìm các luật có độ hỗ trợ và độ tin cậy lớn hơn ngưỡng mà người dùng xác định trước.

Ví dụ: Phân tích giỏ hàng của người mua hàng trong một siêu thị ta thu được

luật: “68% khách hàng mua sữa thì cũng mua bánh mỳ, 21% mua cả hai thứ. Trong ví dụ trên thì 68% là độ tin cậy của luật (số phần trăm giao dịch thỏa mãn vế trái thì thỏa mãn vế phải), 21% là độ hỗ trợ (số phần trăm giao dịch thỏa mãn cả hai vế trái và phải).



Mô tả luật kết hợp

Luật kết hợp mang lại những thông tin vô cùng quan trọng, nó hỗ trợ không nhỏ trong quá trình ra quyết định. Phương pháp này được sử dụng rất nhiều trong các lĩnh vực như marketing có chủ đích, phân tích thị trường, quản lý kinh doanh, ... Khai phá luật kết hợp được thực hiện qua hai bước:

Bước 1: Tìm tất cả các tập mục phổ biến, một tập mục phổ biến được xác định thông qua việc tính độ hỗ trợ và thỏa mãn độ hỗ trợ cực tiểu.

Bước 2: Sinh ra các luật kết hợp mạnh từ tập mục phổ biến, các luật này phải thỏa mãn độ hỗ trợ cực tiểu và độ tin cậy cực tiểu.

Phương pháp này được sử dụng rất hiệu quả trong các lĩnh vực như marketing có chủ đích, phân tích quyết định, quản lí kinh doanh, phân tích thị trường, ...

3.4. Lựa chọn phương pháp khai phá dữ liệu.

Cấu trúc của thuật toán khai phá dữ liệu có ba thành phần chính sau: Biểu diễn mô hình, đánh giá mô hình và phương pháp tìm kiếm.

Biểu diễn mô hình: Mô hình được biểu diễn bằng ngôn ngữ L nào đó để mô tả các mẫu có thể khai phá được. Nếu việc biểu diễn mô hình hạn chế thì không có thời gian học tập hoặc không có các mẫu để tạo ra mô hình chính xác cho dữ liệu. Người phân tích dữ liệu cần phải hiểu đầy đủ các giả thiết mô tả, người thiết kế thuật toán phải diễn tả được giả thiết mô tả nào được tạo ra bởi thuật toán nào một cách rõ ràng.

Đánh giá mô hình: Đánh giá xem mẫu có đáp ứng được các tiêu chuẩn của quá trình phát hiện tri thức hay không. Đánh giá độ chính xác dự đoán dựa

trên đánh giá chéo.

Phương pháp tìm kiếm:

Tìm kiếm tham số: Các thuật toán tìm kiếm các tham số để tối ưu hoá các tiêu chuẩn đánh giá mô hình với dữ liệu quan sát được và với một biểu diễn mô hình đã định.

Tìm kiếm mô hình: Giống như một vòng lặp qua phương pháp tìm kiếm tham số, biểu diễn mô hình bị thay đổi tạo nên họ các mô hình. Với một biểu diễn mô hình, phương pháp tìm kiếm tham số được áp dụng để đánh giá chất lượng mô hình.

Hiện nay, người ta chưa đưa ra được một tiêu chuẩn nào trong việc quyết định sử dụng phương pháp nào vào trong trường hợp nào thì có hiệu quả, có nhiều kỹ thuật và mỗi kỹ thuật được sử dụng cho nhiều bài toán khác nhau. Các thuật toán khai phá dữ liệu tự động chỉ đang ở giai đoạn phát triển ban đầu, các kỹ thuật khai phá dữ liệu còn mới với lĩnh vực kinh doanh. Rõ ràng là để trả lời câu hỏi “khai phá dữ liệu dùng kỹ thuật nào là tốt?” thật không đơn giản vì mỗi phương pháp thì có điểm mạnh và điểm yếu riêng, thậm chí chúng ta còn phải kết hợp các phương pháp trong quá trình khai phá.

4. Ưu điểm và nhược điểm cây quyết định.

- Dự đoán từ thuật toán đưa ra đem lại kết quả tốt kể cả trên các tập dữ liệu chứa các đặc trưng độc lập, đây chính là ưu điểm lớn khi mà số lượng lớn các bài toán trong thực tế chỉ có các tập dữ liệu là các đặc trưng độc lập.
- Xử lý được trên tập dữ liệu categorical và numerical.
- Tốn ít công sức cho hoạt động chuẩn hoá dữ liệu, tuy vậy cây quyết định không sử dụng được trên các điểm dữ liệu trống.
- Khả năng giải các bài toán cho ra nhiều output.
- Kiểm tra tính đúng đắn của mô hình dưa trên thống kê.

Tuy vậy, cây quyết định cũng có những hạn chế nhất định như sau:

- Nếu tập dữ liệu có nhiều biến liên hệ với nhau thì cây quyết định không hoạt động được, cụ thể hơn là nếu training trên các bộ dữ liệu phức tạp, nhiều biến và thuộc tính khác nhau có thể dẫn đến mô hình bị overfit, quá khớp với dữ liệu training dẫn đến hậu quả là mô hình khi đem để thử trên mẫu dữ liệu mới sẽ không cho kết quả chính xác.
- Khi tập dữ liệu được phân chia ra thành các đặc trưng và sự chênh lệch giữa các đặc trưng là nhiều thì mô hình từ thuật toán cây quyết định bị **bias,**

phân nhánh đơn giản chỉ chú ý đến các giá trị tiêu biểu và không kiểm soát hết các khả năng phân loại dữ liệu.

- Tối ưu việc học trên cây quyết định được liệt vào dạng các bài toán NP-complete, thuật toán tham lam và heuristic thường xuyên được sử dụng trong việc giải quyết các bài toán khi việc tối ưu chỉ diễn ra tại mỗi node mà từ đó trả về các kết quả không tối ưu. Nhược điểm này có thể được cải thiện trên kỹ thuật random forest.

5.Thuật toán.

5.1. ID3

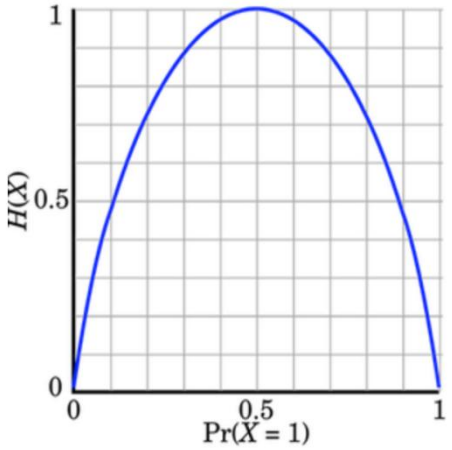
5.1.1. Định nghĩa về Entropy.

Thuật ngữ **entropy** được các nhà khoa học mượn từ lĩnh vực vật lý trong quá trình xây dựng các phương pháp phân loại trong khoa học máy tính. **Entropy** được dùng để đo độ vẩn đục của tập dữ liệu.

Cho một phân phối xác suất của một biến rời rạc x có thể nhận n giá trị khác nhau
$x_1, x_2, ..., x_n$. . Giả sử rằng xác suất để x nhận các giá trị này là $p_i = p\ (x = x_i)$.
Ký hiệu phân phối này là $\boldsymbol{p} = (p_1, p_2, ..., p_n)$.
Entropy của phân phối này là:

$$\mathrm{H}(\mathbf{p}) = -\sum_{\mathrm{i}=1}^{\mathrm{n}} (\mathrm{p_i} \log_2 p_i) \qquad (\mathbf{1})$$

Hàm Entropy được biểu diễn dưới dạng đồ thị như sau:



Từ đồ thị ta thấy hàm Entropy với n > 2 thì

✓ Đạt giá trị **nhỏ nhất** nếu có một giá trị $p_i = 1$.

✓ Đạt giá trị **lớn nhất** nếu tất cả các p_i bằng nhau.

Tổng kết cho định nghĩa về hàm số Entropy.

- Entropy biểu thị cho sự hỗn độn, độ bất định, độ phức tạp của thông tin.
- Thông tin càng phức tạp thì entropy càng cao.
- Entropy nhạy cảm với việc thay đổi sác xuất nhỏ, khi hai phân bố càng giống nhau thì entropy càng giống nhau và ngược lại.
- Mục tiêu khi xây dựng cây quyết định là cho ta nhiều thông tin nhất tức là chọn entropy cao nhất.

Những tính chất này của hàm entropy khiến nó được sử dụng trong việc đo độ vẩn đục của một phép phân chia của ID3. Vì lý do này, ID3 còn được gọi là **entropy-based decision tree**.

5.1.2 Thuật toán ID3 (Iterative Dichotomiser 3)

Thuật toán ID3 lần đầu được công bố bởi Ross Quinlan vào năm 1986, thuật toán hoạt động dựa trên hàm số entropy.

Trong ID3, *tổng có trọng số của entropy tại các leaf-node* sau khi xây dựng decision tree được coi là hàm mất mát của decision tree đó .Các trọng số ở đây tỉ lệ với số điểm dữ liệu được phân vào mỗi node. Công việc của ID3 là tìm các cách phân chia hợp lý (thứ tự chọn thuộc tính hợp lý) sao cho hàm mất mát cuối cùng đạt giá trị càng nhỏ càng tốt. Như đã đề cập, việc này đạt được bằng cách chọn ra thuộc tính sao cho nếu dùng thuộc tính đó để phân chia, entropy tại mỗi bước giảm đi một lượng lớn nhất. Bài toán xây dựng một decision tree bằng ID3 có thể chia thành các bài toán nhỏ, trong mỗi bài toán, ta chỉ cần chọn ra thuộc tính giúp cho việc phân chia đạt kết quả tốt nhất. Mỗi bài toán nhỏ này tương

ứng với việc phân chia dữ liệu trong một *non-leaf node*. Chúng ta sẽ xây dựng phương pháp tính toán dựa trên mỗi node này.

Xét một bài toán với **C** class khác nhau. Giả sử ta đang làm việc với một *non-leaf node* với các
điểm dữ liệu tạo thành một tập S với số phần tử là $|S| = N$.
Giả sử thêm rằng trong số N điểm dữ liệu này N_c, c = 1 , 2 , ..., C điểm thuộc vào class **c**.
Xác suất để mỗi điểm dữ liệu rơi vào một class c được xấp xỉ bằng

$\frac{N_c}{N}$ (maximum likelihood estimation).

Như vậy, entropy tại node này được tính bởi:

$$\mathrm{H(S)} = -\sum_{\mathrm{c=1}}^{\mathrm{C}} \frac{\mathrm{N_c}}{\mathrm{N}} \log \frac{\mathrm{N_c}}{\mathrm{N}} \qquad (\mathbf{2})$$

Tiếp theo, giả sử thuộc tính được chọn là x. Dựa trên x, các điểm dữ liệu trong S được phân ra thành K child node $S_1, S_2, ..., S_k$ với số điểm trong mỗi child node lần lượt là
$m_1, m_2, ..., m_k$. Ta định nghĩa.

$$\mathrm{H}(x, \mathrm{S}) = \sum_{\mathrm{k=1}}^{\mathrm{K}} \frac{\mathrm{m_k}}{\mathrm{N}} \mathrm{H}(S_k) \qquad (\mathbf{3})$$

là tổng có trọng số entroy của mỗi child node–được tính tương tự như (2). Việc lấy trọng số này là quan trọng vì các node thường có số lượng điểm khác nhau.

Tiếp theo, ta định nghĩa i**nformation gain** dựa trên thuộc tính x:

$$x^* = \underset{x}{arg\,max}\, G(x, S) = \underset{x}{arg\,min}\, H(x, S)$$

Tức thuộc tính khiến cho information gain đạt giá trị lớn nhất.

5.1.3 Ví dụ thuật toán ID3.

Dưới đây là tập dữ liệu mô tả quan hệ thời tiết trong 14 ngày gồm bốn thuộc tính **outlook**, **temperature**, **humidity**, **wind.** Cột **play** chính là target mà ta phải dự đoán nếu đã biết giá trị của bốn cột còn lại.

id	outlook	temperature	humidity	wind	play
1	sunny	hot	high	weak	no
2	sunny	hot	high	strong	no
3	overcast	hot	high	weak	yes
4	rainy	mild	high	weak	yes
5	rainy	cool	normal	weak	yes
6	rainy	cool	normal	strong	no
7	overcast	cool	normal	strong	yes
8	sunny	mild	high	weak	no
9	sunny	cool	normal	weak	yes
10	rainy	mild	normal	weak	yes
11	sunny	mild	normal	strong	yes
12	overcast	mild	high	strong	yes
13	overcast	hot	normal	weak	yes
14	rainy	mild	high	strong	no

Dữ liệu về thời tiết ảnh hưởng đến quyết định chơi bóng.

Sơ lược về tập dữ liệu:

Tập dữ liệu gồm bốn thuộc tính thời tiết:
1. **Outlook** có ba giá trị sunny, overcast, rainy.
2. **Temperature** nhận một trong ba giá trị hot, cool, mild.
3. **Humidity** nhận một trong hai giá trị high, normal.
4. **Wind** nhận một trong hai giá trị weak và strong.

Target chính là cột play, đây chính là cột phải đưa quyết định dựa trên các thuộc tính trên gồm có hai giá trị là yes và no.

Tập dữ liệu trên có **14** giá trị kết quả trong đó có **9** giá trị **yes** và **5** giá trị **no**.
Entropy tại root node của bài toán tính theo **(1)** là:

$$\mathbf{H(S)} = -\left(\frac{9}{14}\log\frac{9}{14} + \frac{5}{14}\log\frac{5}{14}\right) \approx 0.65$$

Tính tổng có trọng số entropy của các child node nếu chọn một trong các thuộc tính outlook, temperature, humidity, wind, play để phân chia dữ liệu.
***Lưu ý các phép tính trong machine learning, ngôn ngữ lập trình khi nói đến log là chỉ đến ln.**

- Xét thuộc tính **outlook**. Thuộc tính này có thể nhận một trong ba giá trị sunny, overcast, rainy. Mỗi một giá trị sẽ tương ứng một child node. Gọi **tập hợp các điểm** trong **mỗi child node** này lần lượt là $\boldsymbol{S_s, S_o, S_r}$ với tương ứng $\boldsymbol{m_s, m_o, m_r}$.

Sắp xếp bảng theo thuộc tính **outlook** ta được.

id	outlook	temperature	humidity	wind	play
1	sunny	hot	high	weak	no
2	sunny	hot	high	strong	no
8	sunny	mild	high	weak	no
9	sunny	cool	normal	weak	yes
11	sunny	mild	normal	strong	yes

id	outlook	temperature	humidity	wind	play
3	overcast	hot	high	weak	yes
7	overcast	cool	normal	strong	yes
12	overcast	mild	high	strong	yes
13	overcast	hot	normal	weak	yes

id	outlook	temperature	humidity	wind	play
4	rainy	mild	high	weak	yes
5	rainy	cool	normal	weak	yes
6	rainy	cool	normal	strong	no
10	rainy	mild	normal	weak	yes
14	rainy	mild	high	strong	no

Theo công thức tính Entropy **(2)** ta có được các giá trị:

Tính **Entropy** cho child node **sunny** của thuộc tính **outlook** với **2** giá trị **yes, 3** giá trị **no** và $\boldsymbol{m_s = 5}$.

$$\mathbf{H}(S_s) = -\left(\frac{2}{5}\log\frac{2}{5} + \frac{3}{5}\log\frac{3}{5}\right) \approx 0.673$$

Tính **Entropy** cho child node **overcast** của thuộc tính **outlook** với **4** giá trị **yes, 0** giá trị **no** và $\boldsymbol{m_o = 4}$.

$$\mathbf{H}(S_o) = -\left(\frac{4}{4}\log\frac{4}{4} + \frac{0}{4}\log\frac{0}{4}\right) \approx 0$$

Tính **Entropy** cho child node **rainy** của thuộc tính **outlook** với **3** giá trị **yes, 2** giá trị **no** và $\boldsymbol{m_r = 5}$.

$$\mathbf{H}(S_r) = -\left(\frac{3}{5}\log\frac{3}{5} + \frac{2}{5}\log\frac{2}{5}\right) \approx 0.673$$

Tính tổng trọng số Entropy của mỗi child node theo công thức **(3)** được.

$$\mathbf{H}(\,\mathbf{outlook}, \boldsymbol{S}\,) = \frac{5}{14}\mathrm{H}(S_s) + \frac{4}{14}\mathrm{H}(S_o) + \frac{5}{14}\mathrm{H}(S_r)$$

$$= \frac{5}{14}\,0.673 + \frac{4}{14}\,0 + \frac{5}{14}\,0.673 \approx 0.48$$

❖ Xét thuộc tính **temperature**. Thuộc tính này có thể nhận một trong ba giá trị **hot**, **mild**, **cool** . Mỗi một giá trị sẽ tương ứng một child node. Gọi **tập hợp các điểm** trong **mỗi child node** này lần lượt là S_h, S_m, S_c với tương ứng m_h, m_m, m_c .

Sắp xếp dữ liệu theo thuộc tính **temperature** ta được

id	outlook	temperature	humidity	wind	play
1	sunny	hot	high	weak	no
2	sunny	hot	high	strong	no
3	overcast	hot	high	weak	yes
13	overcast	hot	normal	weak	yes

id	outlook	temperature	humidity	wind	play
4	rainy	mild	high	weak	yes
8	sunny	mild	high	weak	no
10	rainy	mild	normal	weak	yes
11	sunny	mild	normal	strong	yes
12	overcast	mild	high	strong	yes
14	rainy	mild	high	strong	no

id	outlook	temperature	humidity	wind	play
5	rainy	cool	normal	weak	yes
6	rainy	cool	normal	strong	no
7	overcast	cool	normal	strong	yes
9	sunny	cool	normal	weak	yes

Theo công thức tính Entropy **(2)** ta có được các giá trị:

Tính **Entropy** cho child node **hot** của thuộc tính **temperature** với **2** giá trị **yes, 2** giá trị **no** và m_h **= 4.**

$$\mathbf{H}(S_h) = -\left(\frac{2}{4}\log\frac{2}{4} + \frac{2}{4}\log\frac{2}{4}\right) \approx 0.693$$

Tính **Entropy** cho child node **mild** của thuộc tính **temperature** với **4** giá trị **yes, 2** giá trị **no** và m_m **= 6.**

$$\mathbf{H}(S_m) = -\left(\frac{4}{6}\log\frac{4}{6} + \frac{2}{6}\log\frac{2}{6}\right) \approx 0.637$$

Tính **Entropy** cho child node **cool** của thuộc tính **temperature** với **3** giá trị **yes, 1**giá trị **no** và m_c **= 4.**

$$\mathbf{H}(S_c) = -\left(\frac{3}{4}\log\frac{3}{4} + \frac{1}{4}\log\frac{1}{4}\right) \approx 0.562$$

Tính tổng trọng số Entropy của mỗi child node theo công thức **(3)** được.

$$\mathbf{H}(\text{ temperature , } \boldsymbol{S}\ ) = \frac{4}{14}\,\mathrm{H}(S_h) + \frac{6}{14}\,\mathrm{H}(S_m) + \frac{4}{14}\,\mathrm{H}(S_c)$$

$$= \frac{4}{14}\,0.693 + \frac{6}{14}\,0.637 + \frac{4}{14}\,0.562 \approx 0.631$$

- Xét thuộc tính **humidity**. Thuộc tính này có thể nhận một trong ba giá trị **high**, **normal** . Mỗi một giá trị sẽ tương ứng một child node. Gọi **tập hợp các điểm** trong **mỗi child node** này lần lượt là $\boldsymbol{S_h}$, $\boldsymbol{S_n}$ với tương ứng $\boldsymbol{m_h}$, $\boldsymbol{m_n}$.

Sắp xếp dữ liệu theo thuộc tính **humidity** ta được

id	outlook	temperature	humidity	wind	play
1	sunny	hot	high	weak	no
2	sunny	hot	high	strong	no
3	overcast	hot	high	weak	yes
4	rainy	mild	high	weak	yes
8	sunny	mild	high	weak	no
12	overcast	mild	high	strong	yes
14	rainy	mild	high	strong	no

id	outlook	temperature	humidity	wind	play
5	rainy	cool	normal	weak	yes
6	rainy	cool	normal	strong	no
7	overcast	cool	normal	strong	yes
9	sunny	cool	normal	weak	yes
10	rainy	mild	normal	weak	yes
11	sunny	mild	normal	strong	yes
13	overcast	hot	normal	weak	yes

Theo công thức tính Entropy **(2)** ta có được các giá trị:

Tính **Entropy** cho child node **high** của thuộc tính **humidity** với **3** giá trị **yes, 4** giá trị **no** và $\boldsymbol{m_h}$ **= 7.**

$$\mathbf{H}(S_h) = -\left(\frac{3}{7}\log\frac{3}{7} + \frac{4}{7}\log\frac{4}{7}\right) \approx 0.683$$

Tính **Entropy** cho child node **normal** của thuộc tính **humidity** với **6** giá trị **yes, 1** giá trị **no** và $\boldsymbol{m_n}$ **= 7.**

$$\mathbf{H}(S_n) = -\left(\frac{6}{7}\log\frac{6}{7} + \frac{1}{7}\log\frac{1}{7}\right) \approx 0.410$$

Tính tổng trọng số Entropy của mỗi child node theo công thức **(3)** được.

$$\mathbf{H}(\mathbf{humidity}, \boldsymbol{S}) = \frac{7}{14}\,\mathrm{H}(S_h) + \frac{7}{14}\,\mathrm{H}(S_n)$$

$$= \frac{7}{14}\,0.683 + \frac{7}{14}\,0.410 \approx 0.547$$

- Xét thuộc tính **wind**. Thuộc tính này có thể nhận một trong ba giá trị **strong**, **weak**. Mỗi một giá trị sẽ tương ứng một child node. Gọi **tập hợp các điểm** trong **mỗi child node** này lần lượt là $\boldsymbol{S_s}$, $\boldsymbol{S_w}$ với tương ứng $\boldsymbol{m_s}$, $\boldsymbol{m_w}$.

Sắp xếp tập dữ liệu theo thuộc tính wind ta được

id	outlook	temperature	humidity	wind	play
2	sunny	hot	high	strong	no
6	rainy	cool	normal	strong	no
7	overcast	cool	normal	strong	yes
11	sunny	mild	normal	strong	yes
12	overcast	mild	high	strong	yes
14	rainy	mild	high	strong	no

id	outlook	temperature	humidity	wind	play
1	sunny	hot	high	weak	no
3	overcast	hot	high	weak	yes
4	rainy	mild	high	weak	yes
5	rainy	cool	normal	weak	yes
8	sunny	mild	high	weak	no
9	sunny	cool	normal	weak	yes
10	rainy	mild	normal	weak	yes
13	overcast	hot	normal	weak	yes

Theo công thức tính Entropy **(2)** ta có được các giá trị:

Tính **Entropy** cho child node **strong** của thuộc tính **wind** với **3** giá trị **yes, 3** giá trị **no** và $\boldsymbol{m_s}$ **= 6.**

$$\mathbf{H}(S_s) = -\left(\frac{3}{6}\log\frac{3}{6} + \frac{3}{6}\log\frac{3}{6}\right) \approx 0.693$$

Tính **Entropy** cho child node **weak** của thuộc tính **wind** với **6** giá trị **yes, 2** giá trị **no** và $\boldsymbol{m_w}$ **= 8.**

$$\mathbf{H}(S_w) = -\left(\frac{6}{8}\log\frac{6}{8} + \frac{2}{8}\log\frac{2}{8}\right) \approx 0.562$$

Tính tổng trọng số Entropy của mỗi child node theo công thức **(3)** được.

$$\mathbf{H}(\text{ wind }, \boldsymbol{S}) = \frac{6}{14} \text{H}(S_s) + \frac{8}{14} \text{H}(S_w)$$

$$= \frac{6}{14} 0.693 + \frac{8}{14} 0.562 \approx 0.618$$

Tổng trọng số Entropy của bốn thuộc tính sau các phép tính trên là:

- ✓ $\mathbf{H}(\text{ outlook}, \boldsymbol{S}) \approx 0.48$
- ✓ $\mathbf{H}(\text{ temperature}, \boldsymbol{S}) \approx 0.631$
- ✓ $\mathbf{H}(\text{ humidity}, \boldsymbol{S}) \approx 0.547$
- ✓ $\mathbf{H}(\text{ wind}, \boldsymbol{S}) \approx 0.618$

Information gain lớn nhất là khi chọn thuộc tính có trọng số Entropy nhỏ nhất. Vậy ở bước đầu tiên ta chọn thuộc tính **outlook** vì $\mathbf{H}(\text{ outlook}, \boldsymbol{S})$ đạt giá trị nhỏ nhất.

Những thuộc tính của outlook là overcast, sunny, rainy.

- ✓ Nếu thuộc tính là overcast thì sẽ dừng phân nhánh và cho kết quả là yes, vì entropy của thuộc tính overcast là bằng 0.

Nếu thuộc tính là sunny sẽ có entropy là:

$$\mathbf{H}(\mathbf{Sunny}) = -\left(\frac{2}{5}\log\frac{2}{5} + \frac{3}{5}\log\frac{3}{5}\right) \approx 0.673$$

Tập dữ liệu của thuộc tính **outlook** là **sunny** sắp xếp theo **temperature**

id	outlook	temperature	humidity	wind	play
9	sunny	cool	normal	weak	yes
1	sunny	hot	high	weak	no
2	sunny	hot	high	strong	no
8	sunny	mild	high	weak	no
11	sunny	mild	normal	strong	yes

➢ Tính entropy của thuộc tính **temperature** trong tập **outlook** là **sunny**:

$$\mathbf{H}(\mathbf{temperature}, \mathbf{sunny})$$

$$= \frac{1}{5}\mathbf{H}(\mathbf{cool}, \mathbf{temperature}) + \frac{2}{5}\mathbf{H}(\mathbf{hot}, \mathbf{temperature})$$

$$+ \frac{2}{5}\mathbf{H}(\mathbf{mild}, \mathbf{temperature})$$

$$\mathbf{H}(\mathbf{cool}, \mathbf{temperature}) = -\left(\frac{1}{1}\log\frac{1}{1}\right) = 0$$

$$\mathbf{H}(\mathbf{hot}, \mathbf{temperature}) = -\left(\frac{0}{2}\log\frac{0}{2} + \frac{2}{2}\log\frac{2}{2}\right) = 0$$

$$\mathbf{H}(\mathbf{mild}, \mathbf{temperature}) = -\left(\frac{1}{2}\log\frac{1}{2} + \frac{1}{2}\log\frac{1}{2}\right) \approx 0.693$$

- ✓ Vậy $\mathbf{H}(\mathbf{temperature}, \mathbf{sunny}) \approx \mathbf{0.2772}$

Tập dữ liệu của thuộc tính **outlook** là **sunny** sắp xếp theo **humidity.**

id	outlook	temperature	humidity	wind	play
1	sunny	hot	high	weak	no
2	sunny	hot	high	strong	no
8	sunny	mild	high	weak	no
9	sunny	cool	normal	weak	yes
11	sunny	mild	normal	strong	yes

- Tính entropy của thuộc tính **humidity** trong tập **outlook** là **sunny**:

$$\mathbf{H(humidity, sunny)} = \frac{3}{5}\mathbf{H(high, humidity)} + \frac{2}{5}\mathbf{H(normal, humidity)}$$

$$\mathbf{H(high, humidity)} = -\left(\frac{0}{3}\log\frac{0}{3} + \frac{3}{3}\log\frac{3}{3}\right) = 0$$

$$\mathbf{H(normal, humidity)} = -\left(\frac{2}{2}\log\frac{2}{2} + \frac{0}{2}\log\frac{0}{2}\right) = 0$$

✓ Vậy $\mathbf{H(humidity, sunny) = 0}$

Tập dữ liệu của thuộc tính **outlook** là **sunny** sắp xếp theo **wind.**

id	outlook	temperature	humidity	wind	play
2	sunny	hot	high	strong	no
11	sunny	mild	normal	strong	yes
1	sunny	hot	high	weak	no
8	sunny	mild	high	weak	no
9	sunny	cool	normal	weak	yes

- Tính entropy của thuộc tính **wind** trong tập **outlook** là **sunny**:

$$\mathbf{H(wind, sunny)} = \frac{2}{5}\mathbf{H(strong, wind)} + \frac{3}{5}\mathbf{H(weak, wind)}$$

$$\mathbf{H(strong, wind)} = -\left(\frac{1}{2}\log\frac{1}{2} + \frac{1}{2}\log\frac{1}{2}\right) \approx 0.693$$

$$\mathbf{H(weak, wind)} = -\left(\frac{1}{3}\log\frac{1}{3} + \frac{2}{3}\log\frac{2}{3}\right) \approx 0.637$$

✓ Vậy $\mathbf{H(wind, sunny) \approx 0.66}$

✓ Chọn thuộc tính **humidity** để tiếp tục phân nhánh vì tổng trọng số entropy bằng 0 với output là yes khi và chỉ khi humidity là **normal**.

Nếu thuộc tính là rainy sẽ có entropy là:

$$\mathbf{H(rainy)} = -\left(\frac{2}{5}\log\frac{2}{5} + \frac{3}{5}\log\frac{3}{5}\right) \approx 0.673$$

Tập dữ liệu của thuộc tính **outlook** là **rainy** sắp xếp theo **temperature**

id	outlook	temperature	humidity	wind	play
5	rainy	cool	normal	weak	yes
6	rainy	cool	normal	strong	no
4	rainy	mild	high	weak	yes
10	rainy	mild	normal	weak	yes
14	rainy	mild	high	strong	no

➢ Tính entropy của thuộc tính **temperature** trong tập **outlook** là **rainy**:

$$\mathbf{H(temperature, rainy)} = \frac{2}{5}\mathbf{H(cool, temperature)} + \frac{0}{5}\mathbf{H(hot, temperature)} + \frac{3}{5}\mathbf{H(mild, temperature)}$$

$$\mathbf{H(cool, temperature)} = -\left(\frac{1}{2}\log\frac{1}{2} + \frac{1}{2}\log\frac{1}{2}\right) \approx 0.693$$

$$\mathbf{H(hot, temperature)} = 0$$

$$\mathbf{H(mild, temperature)} = -\left(\frac{2}{3}\log\frac{2}{3} + \frac{1}{3}\log\frac{1}{3}\right) \approx 0.636$$

✓ Vậy $\mathbf{H(temperature, rainy)} \approx \mathbf{0.6588}$

Tập dữ liệu của thuộc tính **outlook** là **rainy** sắp xếp theo **humidity.**

id	outlook	temperature	humidity	wind	play
4	rainy	mild	high	weak	yes
14	rainy	mild	high	strong	no
5	rainy	cool	normal	weak	yes
6	rainy	cool	normal	strong	no
10	rainy	mild	normal	weak	yes

➢ Tính entropy của thuộc tính **humidity** trong tập **outlook** là **rainy**:

$$\mathbf{H}(\text{humidity, }\mathbf{rainy}) = \frac{2}{5}\mathbf{H}(\text{high, }\mathbf{humidity}) + \frac{3}{5}\mathbf{H}(\text{normal, }\mathbf{humidity})$$

$$\mathbf{H}(\text{high, }\mathbf{humidity}) = -\left(\frac{1}{2}\log\frac{1}{2} + \frac{1}{2}\log\frac{1}{2}\right) \approx 0.693$$

$$\mathbf{H}(\text{normal, }\mathbf{humidity}) = -\left(\frac{2}{3}\log\frac{2}{3} + \frac{1}{3}\log\frac{1}{3}\right) \approx 0.636$$

✓ Vậy $\mathrm{H}(\text{humidity, }\mathbf{rainy}) \approx 0.659$

Tập dữ liệu của thuộc tính **outlook** là **rainy** sắp xếp theo **wind.**

id	outlook	temperature	humidity	wind	play
14	rainy	mild	high	strong	no
6	rainy	cool	normal	strong	no
4	rainy	mild	high	weak	yes
5	rainy	cool	normal	weak	yes
10	rainy	mild	normal	weak	yes

- Tính entropy của thuộc tính **wind** trong tập **outlook** là **sunny**:

$$\mathbf{H(wind, sunny)} = \frac{2}{5}\mathbf{H(strong, wind)} + \frac{3}{5}\mathbf{H(weak, wind)}$$

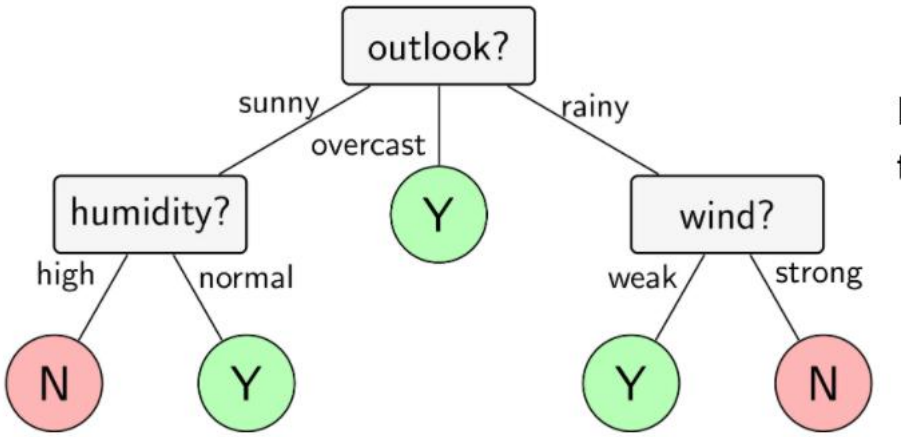
$$\mathbf{H(strong, wind)} = -\left(\frac{0}{2}\log\frac{0}{2} + \frac{2}{2}\log\frac{2}{2}\right) = 0$$

$$\mathbf{H(weak, wind)} = -\left(\frac{3}{3}\log\frac{3}{3} + \frac{0}{3}\log\frac{0}{3}\right) = 0$$

- ✓ Vậy $\mathbf{H(wind, rainy) = 0}$

- ✓ Chọn thuộc tính **wind** để tiếp tục phân nhánh vì tổng trọng số entropy bằng 0 với output là **yes** khi và chỉ khi rainy là **weak**.

Từ đó ta có cây quyết định được xây dựng như hình bên dưới



5.2 Random forest.

Random Forest là một thành viên trong họ thuật toán **decision tree** (cây quyết định).Ý tưởng phía sau *Random Forest* khá đơn giản.
Thuật toán này sinh một số cây quyết định (thường là vài trăm) và sử dụng chúng. Các câu hỏi của cây quyết định sẽ là câu hỏi về các thuộc tính.
Ví dụ: "Cánh hoa có dài hơn 1.7cm hay không?". Câu giá trị ở nút lá sẽ là các lớp (*class*). Sử dụng hàng trăm cây quyết định là bất khả thi với con người, nhưng máy tính có thể làm việc này tương đối dễ dàng.

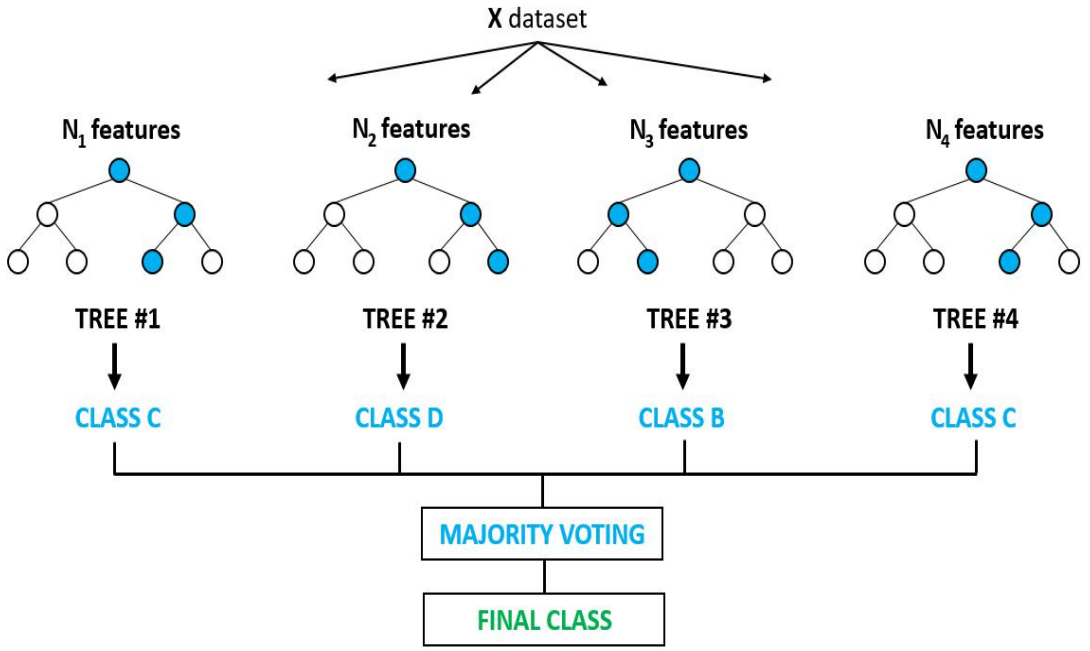
Có hai giải pháp. Cách thứ nhất là hỏi chuyên gia (ví dụ như một nhà nhân chủng học với bài toán phân biệt chủng loại của con vật). Nhưng không phải khi nào cũng có thể tiếp cận được với chuyên gia trong bài toán của mình. Hơn nữa, ngay cả những chuyên gia giỏi nhất cũng gặp khó khăn trong việc viết ra những kiến thức của mình và ngay cả khi tìm được một chuyên gia có khả năng đó thì chắc chắn sẽ có những thứ mà họ không biết tới. Ví dụ, nhà nhân chủng học của chúng ta có thể quên mất rằng con đà điểu có thể nhỏ hơn 50kg.

Thay vì sử dụng chuyên gia, các nhà nghiên cứu sử dụng phương án thứ hai: tạo ra một thuật toán tự sinh cây quyết định. Điều kiện duy nhất là phải có vài ví dụ

để máy tính có thể tham chiếu. Trong Iris dataset, những ví dụ này chính là những bông hoa mà chúng ta đã biết chủng loại.

Để tạo ra một cây quyết định, thuật toán *Random Forest* luôn bắt đầu bằng một cây rỗng. Một cây quyết định rỗng chỉ có một ô *Start* chỉ thẳng đến câu trả lời (ô xanh lá). Tiếp theo, thuật toán sẽ tìm câu hỏi đầu tiên và bắt đầu xây dựng cây quyết định. Mỗi lần thuật toán tìm được thêm một câu hỏi, nó tạo hai nhánh trên cây quyết định. Khi không còn câu hỏi nào nữa, thuật toán dừng lại và chúng ta có một cây quyết định hoàn chỉnh.

Làm thế nào để tìm ra những câu hỏi tốt nhất cho cây quyết định? Đây là một bước khá phức tạp nhưng ý tưởng đằng sau nó tương đối đơn giản: Ở thời điểm bắt đầu, thuật toán của chúng ta chưa biết phân biệt các chủng loại của các con vật. Nói cách khác, tất cả các con vật được cho chung vào một "cái túi". Để tìm ra câu hỏi tốt nhất, thuật toán thử đưa ra tất cả các câu hỏi có thể (có khi là hàng triệu câu hỏi). Ví dụ: "Nó có bao nhiêu chân?", "Nó có đuôi không?",... Sau đó, với mỗi câu hỏi, thuật toán sẽ đánh giá mức độ hiệu quả mà câu hỏi này giúp phân biệt các chủng loại, hay các *class*. Câu hỏi được chọn không cần thiết phải hoàn hảo, nhưng nó phải tốt hơn những câu hỏi khác. Để tính toán mức độ hiệu quả của câu hỏi, chúng ta sử dụng một độ đo có tên là **information gain**.
Có thể hiểu **information gain** như một cách để "cho điểm" các câu hỏi. Câu hỏi với *information gain* lớn nhất sẽ được chọn như là câu hỏi tốt nhất để xây dựng cây quyết định. Sau khi thuật toán xây dựng xong các cây quyết định, những cây này sẽ được sử dụng để trả lời câu hỏi (hay phân loại).



Random Forest coi mỗi cây quyết định như một cử tri bỏ phiếu độc lập (như một cuộc bầu cử thực sự). Ở cuối cuộc bầu cử, câu trả lời nhận được nhiều bầu chọn nhất từ các cây quyết định sẽ được lựa chọn.

Tuy nhiên, vẫn còn một vấn đề: Nếu như tất cả các cây được dựng theo cùng một cách, chúng sẽ cho những câu trả lời giống nhau. Như vậy chẳng khác gì chúng ta chỉ sử dụng một cây quyết định duy nhất cả. Ở đây, Random Forest có một cách làm rất hay: Để chắc chắn rằng không phải tất cả các cây quyết định cho cùng câu trả lời, thuật toán Random Forest chọn ngẫu nhiên các quan sát (observations). Chính xác hơn, Random Forest sẽ xoá một số quan sát và lặp lại một số khác một cách ngẫu nhiên. Xét toàn cục, những quan sát này vẫn rất gần với tập các quan sát ban đầu, nhưng những thay đổi nhỏ sẽ đảm bảo rằng mỗi cây quyết định sẽ có một chút khác biệt. Quá trình này gọi là **bootstrapping**

Thêm vào đó, để thực sự chắc chắn các cây quyết định là khác nhau, thuật toán Random Forest sẽ ngẫu nhiên bỏ qua một số câu hỏi khi xây dựng cây quyết định. Trong trường hợp này, nếu câu hỏi tốt nhất không được chọn, một câu hỏi kế tiếp sẽ được lựa chọn để dựng cây. Quá trình này được gọi là **attribute sampling**

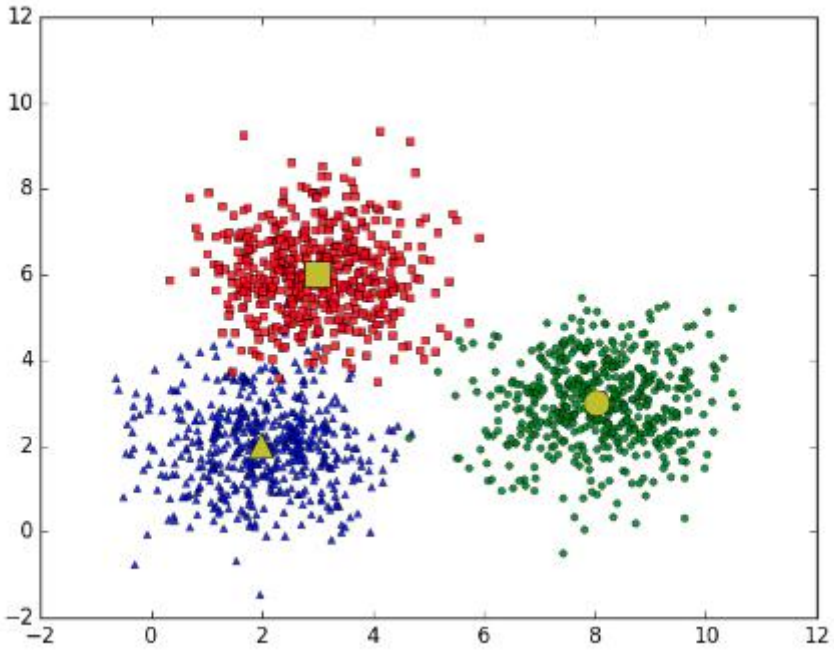
Tạo ra một thuật toán phức tạp như vậy, ngẫu nhiên thay đổi các quan sát và bỏ qua một số câu hỏi. Câu trả lời rất đơn giản: Có thể các mẫu thử mà chúng ta đang sử dụng chưa hoàn hảo. Ví dụ, có thể mẫu thử của chúng ta chỉ có những con mèo có lông đuôi. Trong trường hợp này những con mèo thuộc loài sphynx (mèo không lông) có thể được phân loại là con chuột. Tuy nhiên, nếu câu hỏi về đuôi không được hỏi (bởi vì sự thay đổi ngẫu nhiên), thuật toán có thể sử dụng câu hỏi các câu hỏi khác (ví dụ: Con vật đó có kích thước như thế nào?). Việc có nhiều câu hỏi đa dạng (có thể không hoàn hảo) là một ý tưởng không tồi: nó có thể là cứu tinh khi thuật toán tham chiếu đến một quan sát mà nó chưa từng thấy trước đây.

5.3. K-means clustering.

5.3.1. Khái niệm.

K-means clustering là một trong những thuật toán cơ bản nhất trong unsupervised learning.Trong thuật toán K-means clustering, chúng ta không biết nhãn (label) của từng điểm dữ liệu. Mục đích là làm thể nào để phân dữ liệu thành các cụm (cluster) khác nhau sao cho dữ liệu trong cùng một cụm có tính chất giống nhau.

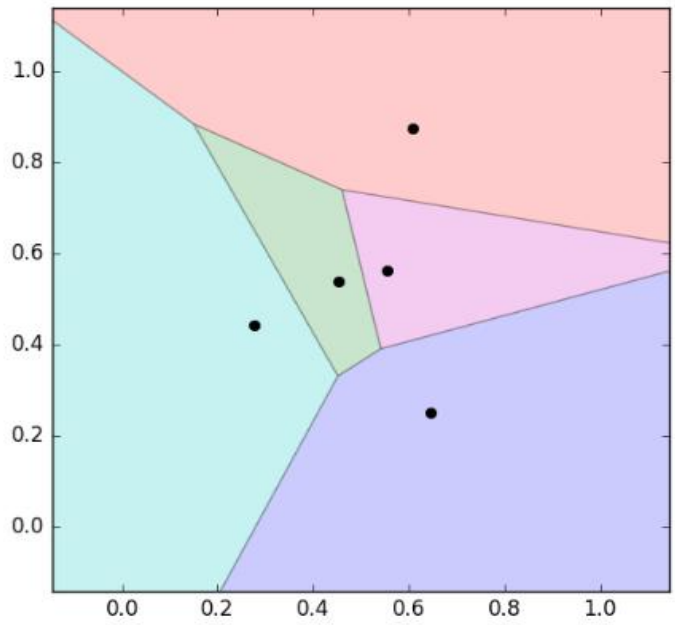
Ý tưởng đơn giản nhất về cluster (cụm) là tập hợp các điểm *ở gần nhau trong một không gian nào đó* (không gian này có thể có rất nhiều chiều trong trường hợp thông tin về một điểm dữ liệu là rất lớn). Hình bên dưới là một ví dụ về 3 cụm dữ liệu (từ giờ sẽ được viết gọn là *cluster*).



Bài toán với 3 cluster

Giả sử mỗi cluster có một điểm đại diện (*center*) màu vàng. Và những điểm xung quanh mỗi center thuộc vào cùng nhóm với center đó. Một cách đơn giản nhất, xét một điểm bất kỳ, ta xét xem điểm đó gần với center nào nhất thì nó thuộc về cùng nhóm với center đó. Tới đây, chúng ta có một bài toán thú vị: *Trên một vùng biển hình vuông lớn có ba đảo hình vuông, tam giác, và tròn màu vàng như hình trên. Một điểm trên biển được gọi là thuộc lãnh hải của một đảo nếu nó nằm gần đảo này hơn so với hai đảo kia . Hãy xác định ranh giới lãnh hải của các đảo.*

Hình dưới đây là một hình minh họa cho việc phân chia lãnh hải nếu có 5 đảo khác nhau được biểu diễn bằng các hình tròn màu đen:



Phân vùng lãnh hải của mỗi đảo. Các vùng khác nhau có màu sắc khác nhau.

Chúng ta thấy rằng đường phân định giữa các lãnh hải là các đường thẳng (chính xác hơn thì chúng là các đường trung trực của các cặp điểm gần nhau). Vì vậy, lãnh hải của một đảo sẽ là một hình đa giác.

Cách phân chia này trong toán học được gọi là Voronoi Diagram.

Trong không gian ba chiều, lấy ví dụ là các hành tinh, thì (tạm gọi là) lãnh không của mỗi hành tinh sẽ là một đa diện. Trong không gian nhiều chiều hơn, chúng ta sẽ có những thứ (mà tôi gọi là) siêu đa diện (hyperpolygon).

5.3.2. Tóm tắt thuật toán

Mục đích cuối cùng của thuật toán phân nhóm này là: từ dữ liệu đầu vào và số lượng nhóm chúng ta muốn tìm, hãy chỉ ra center của mỗi nhóm và phân các điểm dữ liệu vào các nhóm tương ứng.

Đầu vào: Dữ liệu **X** và số lượng cluster cần tìm **K**.

Đầu ra: Các center **M** và label vector cho từng điểm dữ liệu **Y**.

1. Chọn **K** điểm bất kỳ làm các center ban đầu.
2. Phân mỗi điểm dữ liệu vào cluster có center gần nó nhất.
3. Nếu việc gán dữ liệu vào từng cluster ở bước 2 không thay đổi so với vòng lặp trước nó thì ta dừng thuật toán.
4. Cập nhật center cho từng cluster bằng cách lấy trung bình cộng của tất

các các điểm dữ liệu đã được gán vào cluster đó sau bước 2.

5. Quay lại bước 2.

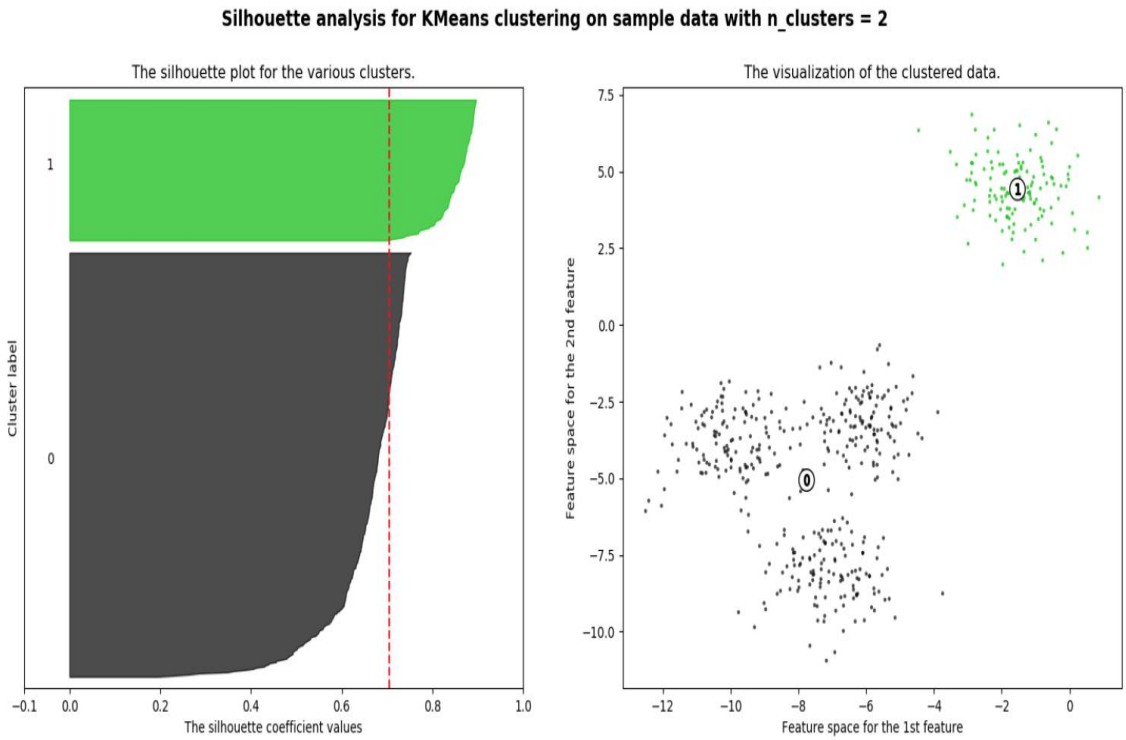
5.4. Silhouette coefficient.

Trong quá trình đi tìm cụm cho một tập dữ liệu thì quá trình đánh giá số lượng, chất lượng cụm là cần thiết.Hệ số Silhouette của mô hình cao hơn thì mô hình này cũng tốt hơn. Cách tính hệ số silhouette cho mỗi mẫu thử như sau:

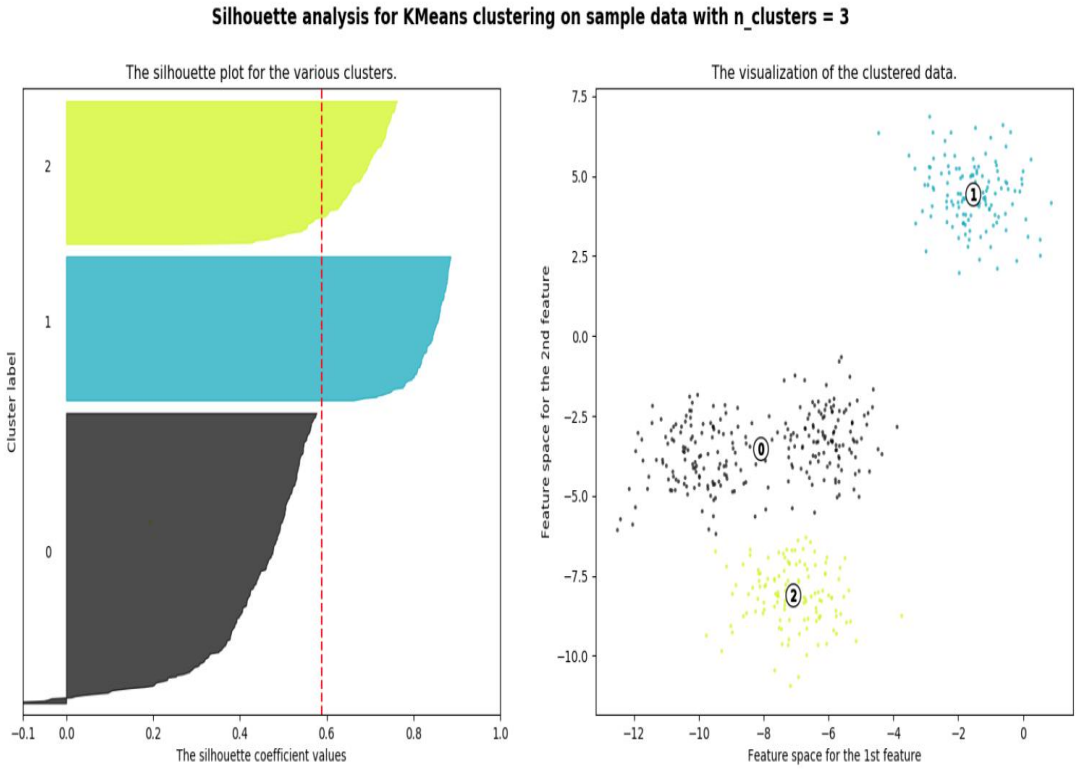
a : trung bình khoảng cách của các điểm trong cùng lớp của mẫu.
b : trung bình khoảng cách của các điểm khác lớp của mẫu.

$$s = \frac{b - a}{max(a, b)}$$

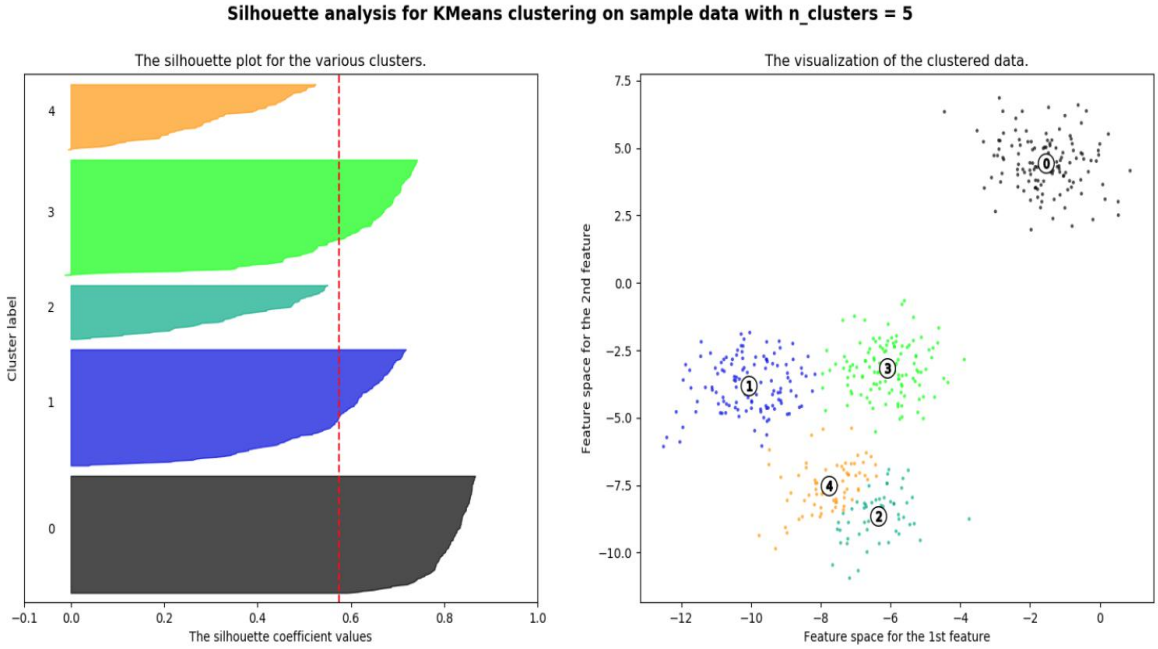
Hình sau được sử dụng từ nguồn của sklearn



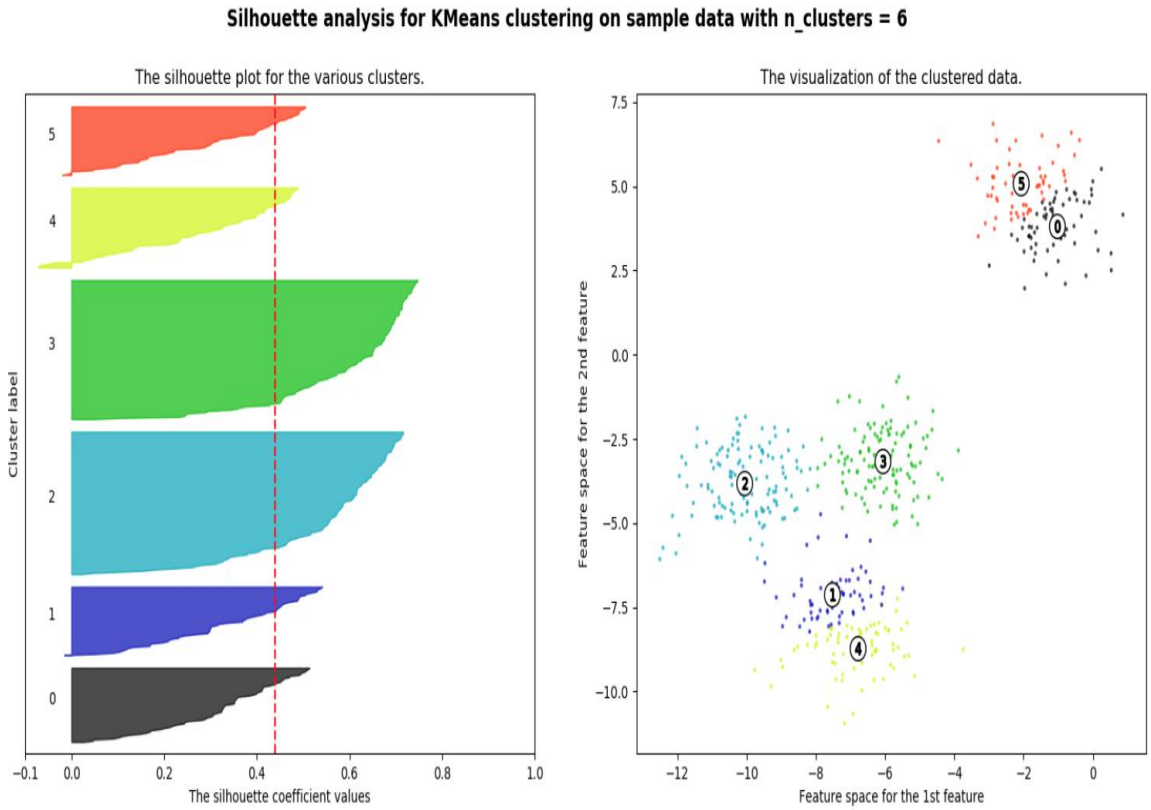
Số lượng cụm là 2



Số lượng cụm là 3



Số lượng cụm là 5



Số lượng cụm là 6

```
For n_clusters = 2 The average silhouette_score is : 0.7049787496083262
For n_clusters = 3 The average silhouette_score is : 0.5882004012129721
For n_clusters = 4 The average silhouette_score is : 0.6505186632729437
For n_clusters = 5 The average silhouette_score is : 0.5745566973301872
For n_clusters = 6 The average silhouette_score is : 0.43902711183132426
```

Kết quả khi chạy thuật toán

Theo kết quả thì với số lượng cụm là 2 có số điểm silhouette cao nhất tuy vậy khi nhìn sang các điểm đã được tô màu thì số lượng cụm vẫn còn có thể tăng lên được. Khi tăng số điểm lên 3,5,6 thì số điểm silhouette cao nhất là 5 và cũng cho đồ thị đẹp nhất, ở đây ta có đủ cơ sở để công nhận số lượng cụm bằng 5 là cách chọn hiệu quả cho mô hình.

6. Ứng dụng

6.1. Mục tiêu

Mục tiêu của phần báo cáo này là diễn giải cách hoạt động của ứng dụng từ tập dữ liệu ban đầu trải qua các bước phân tích dữ liệu để có được dữ liệu cuối cùng có thể dùng được trong việc train máy học, thử và so sánh trên nhiều loại thuật toán khác nhau như Decision Tree, Random Forest...

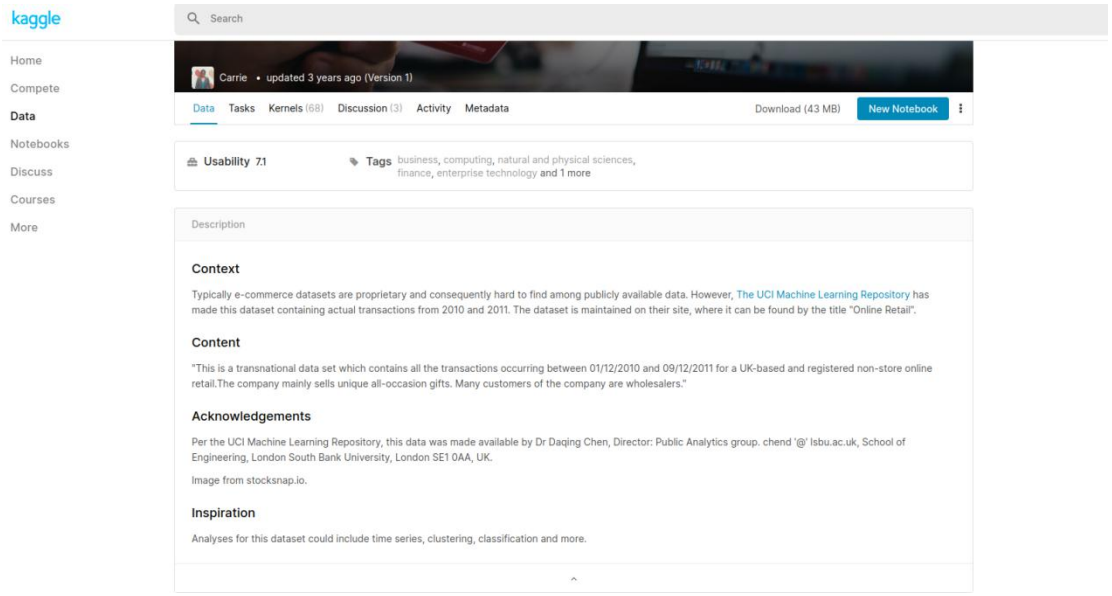
Quá trình giải thích về ứng dụng sẽ qua phần sau:

1. Chuẩn bị dữ liệu.
2. Tìm hiểu về nội dung của tập dữ liệu.
3. Tiếp cận phân loại sản phẩm.
4. Tiếp cận phân loại người dùng.
5. Thử phân loại người dùng trên nhiều thuật toán khác nhau.
6. Kết luận.

Từ những thông tin giao dịch cơ bản ban đầu từ của sàn thương mại điện tử, ta sẽ chiết xuất dữ liệu để có được cụm sản phẩm , sau đó tiếp tục phân tích và tìm ra được cụm người dùng. Có được hai cụm sản phẩm và người dùng thì việc định danh vùng người dùng cho từng người dùng là hoàn toàn có thể, có được tập dữ liệu sau khi định danh ta dùng để huấn luyện máy học nhận kết quả và so sánh.

Code và documentation của bài báo cáo tìm thấy tại link này.

6.2 Dữ liệu



Dữ liệu có thể được tìm thấy tại liên kết này , tập dữ liệu là những giao dịch đã diễn ra từ khoảng **01/12/2010** đến **09/12/2011** của một công ty có trụ sở đặt tại vương quốc Anh. Toàn bộ dữ liệu được ghi lại trong quá trình hoạt động của công ty, công ty chuyên buôn bán các mặt hàng quà tặng. Nhiều khách hàng của công ty là thương buôn hay nhà phân phối.

Tập dữ liệu sẽ gồm **541909** dòng và **8** cột.

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850	United Kingdom

Hình thể hiện 5 dòng đầu của tập dữ liệu.

Nội dung từng cột trong tập dữ liệu:

1. InvoiceNo: Mã đơn đặt hàng (gồm 6 chữ số). Bắt đầu bằng 'C' nghĩa là đơn hàng đã bị hủy.
2. StockCode: Mã sản phẩm, mỗi sản phẩm có một loại code riêng biệt.
3. Description: Mô tả sản phẩm.
4. Quantity: Số lượng sản phẩm trong đơn hàng.
5. InvoiceDate: Ngày đặt hàng.
6. UnitPrice: Đơn vị tiền tệ dùng mua hàng
7. CustomerID: Mã khách hàng.
8. Country: Tên quốc gia.

Người dùng trong dữ liệu đến từ **37** quốc gia, trong đó **Anh, Pháp, Đức** là 3 quốc gia sử dụng website nhiều nhất.

Hình ảnh plot ra từ tập dữ liệu.

6.3 Sơ lược về khách hàng và sản phẩm.

Số lượng sản phẩm là **3684**, số lần giao dịch là **22190** và số lượng khách hàng đã thanh toán trên dịch vụ của website là **4372**.

	products	transactions	customers
quantity	3684	22190	4372

Mỗi khách hàng có thể mua được nhiều sản phẩm trong một lần giao dịch và mã khách hàng **CustomerID**, mã giao dịch **InvoiceNo** là không bị trùng lặp.

	CustomerID	InvoiceNo	Number of products
0	12346	541431	1
1	12346	C541433	1
2	12347	537626	31
3	12347	542237	29
4	12347	549222	24
5	12347	556201	18
6	12347	562032	22
7	12347	573511	47
8	12347	581180	11
9	12348	539318	17

Các đơn hàng bị hủy không có đóng góp cho quá trình phân tích nên sẽ bị loại bỏ

	CustomerID	InvoiceNo	Number of products	order_canceled
0	12346	541431	1	0
1	12346	C541433	1	1
2	12347	537626	31	0
3	12347	542237	29	0
4	12347	549222	24	0

Tỉ lệ đơn bị hủy trên số đơn đặt hàng : 3654/22190 (16.47%)

Tỉ lệ đơn hàng bị hủy chiếm **16.47%** trên tổng số giao dịch, đây là một số khá lớn khi chiếm gần 1/5 trên tổng số giao dịch.

Một điều đáng chú ý nữa là các đơn bị hủy thì đứng trước trong cột **InvoiceNo** sẽ có ký tự **C** . Hàng thứ hai hình bên trên là một ví dụ.

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
61619	541431	23166	MEDIUM CERAMIC TOP STORAGE JAR	74215	2011-01-18 10:01:00	1.04	12346	United Kingdom
61624	C541433	23166	MEDIUM CERAMIC TOP STORAGE JAR	-74215	2011-01-18 10:17:00	1.04	12346	United Kingdom
286623	562032	22375	AIRLINE BAG VINTAGE JET SET BROWN	4	2011-08-02 08:48:00	4.25	12347	Iceland
72260	542237	84991	60 TEATIME FAIRY CAKE CASES	24	2011-01-26 14:30:00	0.55	12347	Iceland
14943	537626	22772	PINK DRAWER KNOB ACRYLIC EDWARDIAN	12	2010-12-07 14:57:00	1.25	12347	Iceland

Nhìn vào hàng **một** và hàng **hai** của hình trên ta thấy được trừ cột **InvoiceNo** và cột **Quantity**(cột quantity hàng hai giống như hàng một nhưng là số âm) khác nhau còn lại là hoàn toàn giống nhau, ta dự đoán đây là yếu tố chính trong việc tìm ra các đơn hàng đã đặt và đã bị hủy.

Dựa vào điều đó để **loại** đi các đơn hàng đã bị **hủy**.

Mã sản phẩm còn có ý nghĩa khác là khả năng cho biết đặc điểm của giao dịch, ví dụ khi tách chữ cái từ cột **StockCode** thì ta có dãy các chữ cái là
POST, D, C2, M, BANK CHARGES, PADS, DOT

Ý nghĩa của các ký tự được tách ra như sau:

- ✓ POST : POSTAGE (hàng gửi qua bưu chính)
- ✓ D : Discount (hàng giảm giá)
- ✓ C2 : CARRIAGE (hàng vận chuyển)
- ✓ M : Manual (hàng dễ vỡ).
- ✓ BANK CHARGES : phụ thu từ ngân hàng .
- ✓ PADS : PADS TO ALL CUSHIONS.
- ✓ DOT COM : DOTCOM POSTAGES.

Thêm cột **TotalPrice** cho bảng, công thức tính cột này là số lượng hàng (**Quantity**) nhân cho giá (**UnitPrice**).

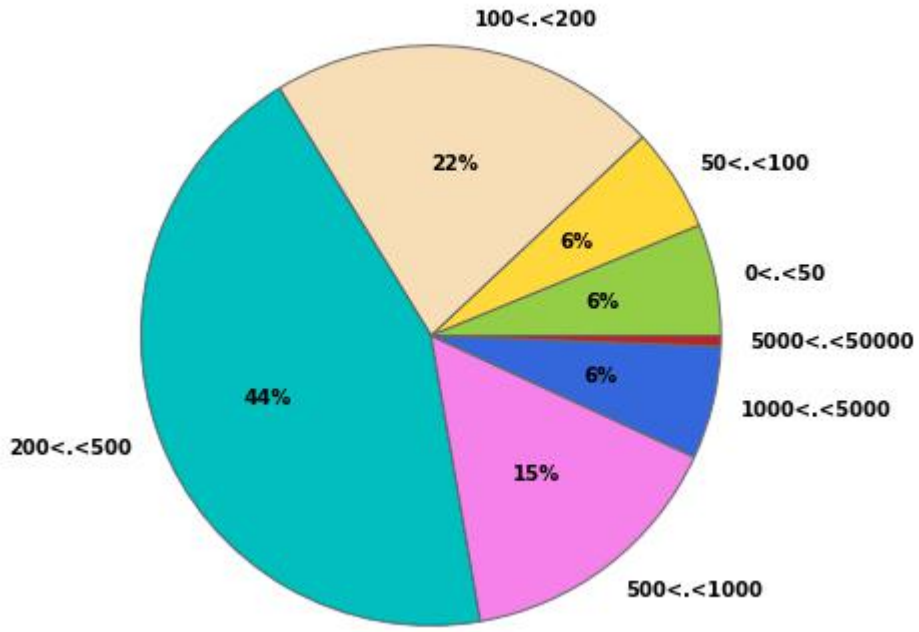
$$TotalPrice = Quantity \; x \; UnitPrice$$

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country	QuantityCanceled	TotalPrice
61619	541431	23166	MEDIUM CERAMIC TOP STORAGE JAR	74215	2011-01-18 10:01:00	1.04	12346	United Kingdom	74215	0.0
148288	549222	22375	AIRLINE BAG VINTAGE JET SET BROWN	4	2011-04-07 10:43:00	4.25	12347	Iceland	0	17.0
428971	573511	22698	PINK REGENCY TEACUP AND SAUCER	12	2011-10-31 12:25:00	2.95	12347	Iceland	0	35.4
428970	573511	47559B	TEA TIME OVEN GLOVE	10	2011-10-31 12:25:00	1.25	12347	Iceland	0	12.5
428969	573511	47567B	TEA TIME KITCHEN APRON	6	2011-10-31 12:25:00	5.95	12347	Iceland	0	35.7

Các món hàng cùng giỏ thì có cùng **InvoiceNo**, ta gom tất cả vào cùng một giỏ và tính tổng các **TotalPrice** lại với nhau gọi là **BasketPrice**.

	CustomerID	InvoiceNo	Basket Price	InvoiceDate
1	12347	537626	711.79	2010-12-07 14:57:00.000001024
2	12347	542237	475.39	2011-01-26 14:29:59.999999744
3	12347	549222	636.25	2011-04-07 10:42:59.999999232
4	12347	556201	382.52	2011-06-09 13:01:00.000000256
5	12347	562032	584.91	2011-08-02 08:48:00.000000000
6	12347	573511	1294.32	2011-10-31 12:25:00.000001280

Hình bên dưới là phân phối giá trị đơn đặt hàng



Nhận xét số lượng giỏ hàng có giá từ 200£ đến 500£ chiếm phần lớn gần 50% trên tổng giao dịch, mức giao dịch có khoảng bé hơn 50£ chỉ chiếm 6%. Bổ sung thêm các khoảng giao dịch trong tầm giá từ 5000£ đến 50000£ là không nhiều và chiếm rất ít trong toàn bộ giao dịch.

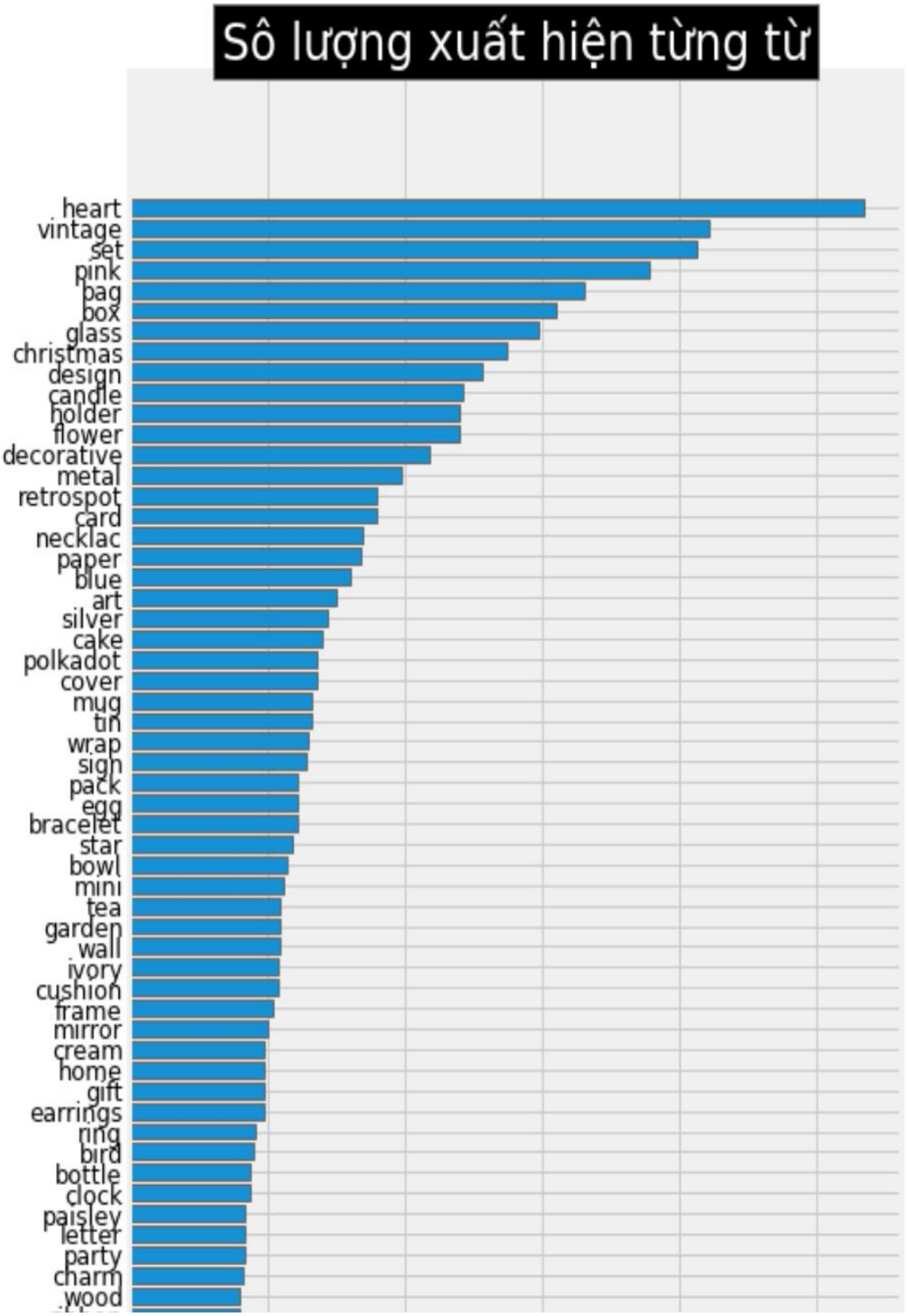
6.4 Phân loại sản phẩm.

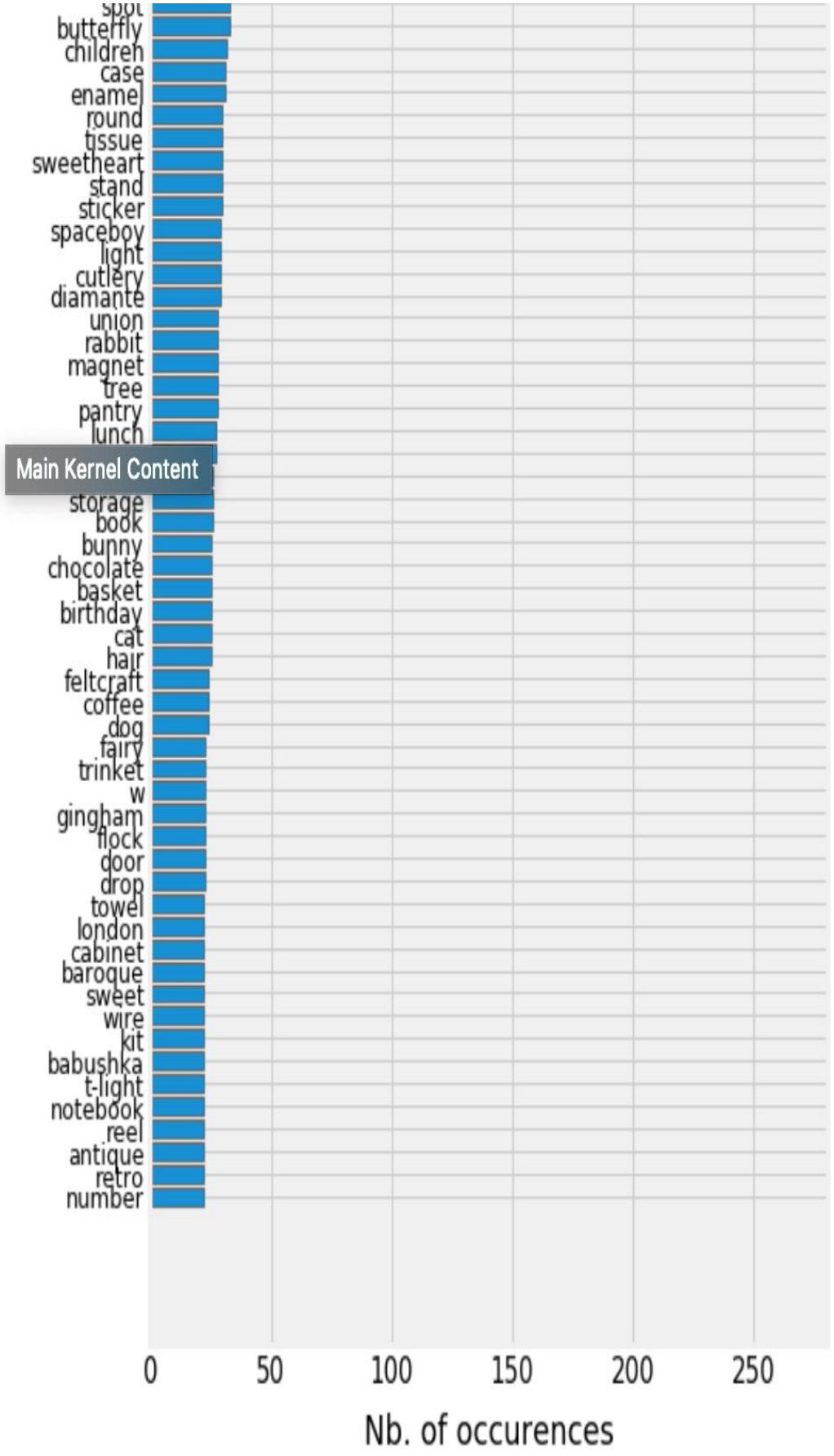
6.4.1. Chiết xuất từ loại.

Thông tin trong cột description trong mỗi giao dịch đóng vai trò rất quan trọng trong việc phân loại sản phẩm. VĐiều đầu tiên cần làm là xác định các từ có nghĩa trong tập dữ liệu.

Số lượng từ ngữ có nghĩa là **1483** từ.

Hai hình bên dưới cho thấy các từ ngữ được dùng cùng với số lần xuất hiện của từ đó trong dữ liệu.


Sô lượng xuất hiện từng từ
heart
vintage
set
pink
bag
box
glass
christmas
design
candle
holder
flower
decorative
metal
retrospot
card
necklac
paper
blue
art
silver
cake
polkadot
cover
mug
tin
wrap
sign
pack
egg
bracelet
star
bowl
mini
tea
garden
wall
ivory
cushion
frame
mirror
cream
home
gift
earrings
ring
bird
bottle
clock
paisley
letter
party
charm
wood


spot
butterfly
children
case
enamel
round
tissue
sweetheart
stand
sticker
spaceboy
light
cutlery
diamante
union
rabbit
magnet
tree
pantry
lunch
Main Kernel Content
storage
book
bunny
chocolate
basket
birthday
cat
hair
feltcraft
coffee
dog
fairy
trinket
w
gingham
flock
door
drop
towel
london
cabinet
baroque
sweet
wire
kit
babushka
t-light
notebook
reel
antique
retro
number
0
50
100
150
200
250
Nb. of occurences

Số lượng từ trong tập dữ liệu có thể dùng được là hơn 1400 từ và một số từ phổ biến trong hơn 200 sản phẩm. Nếu lược qua hai hình trên thì ta có thể thấy là một số từ không có nghĩa hoặc là sẽ không có đóng góp gì cho phân loại sản phẩm ví dụ như là các từ ngữ chỉ màu sắc hoặc là các từ có tần số xuất hiện ít mà cụ thể là vào khoảng dưới 20 lần xuất hiện. Loại bỏ các từ đó đi thì số lượng các từ chỉ còn khoảng 193.

6.4.2 Mã hoá Dữ liệu

Tạo một ma trận với **cột** là các từ vựng trong nhóm trên và **hàng** chính là các sản phẩm trong tập dữ liệu. Gọi ma trận là ***X***, $a_{i,j} = 1$ nếu tồn tại từ loại **i** trong sảng phẩm **j**.

	mot 1	...	mot j	...	mot N
produit 1	$a_{1,1}$				$a_{1,N}$
...			...		
produit i	...		$a_{i,j}$		...
...			...		
produit M	$a_{M,1}$				$a_{M,N}$

Đây là ma trận sau nhị phân khi hiện thức bước trên.

	heart	vintage	set	bag	box	glass	christmas	design	candle	holder	flower	decorative	metal	retrospot	card	necklac	paper	art	silver	cake
0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
3873	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3874	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3875	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3876	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0
3877	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

3878 rows × 193 columns

Ngoài ra thì các thuộc tính về vùng giá của sản phẩm cũng đóng một vai trò quan trọng trong việc phân vùng sản phẩm nên ta thêm 6 cột về vùng giá vào ma trận X.

Phạm vi	Số lượng sản phẩm
0<.<1	964
1<.<2	1009
2<.<3	673
3<.<5	606
5<.<10	470
.>10	156

Hình trên thể hiện **6** vùng giá của sản phẩm .

Các sản phẩm có giá từ 1£ đến 2£ có số lượng nhiều nhất hơn 1000 sản phẩm, các sản phẩm có giá hơn 10£ chiếm số lượng ít nhất chỉ có 156 sản phẩm.

6.4.3. Phân cụm sản phẩm.

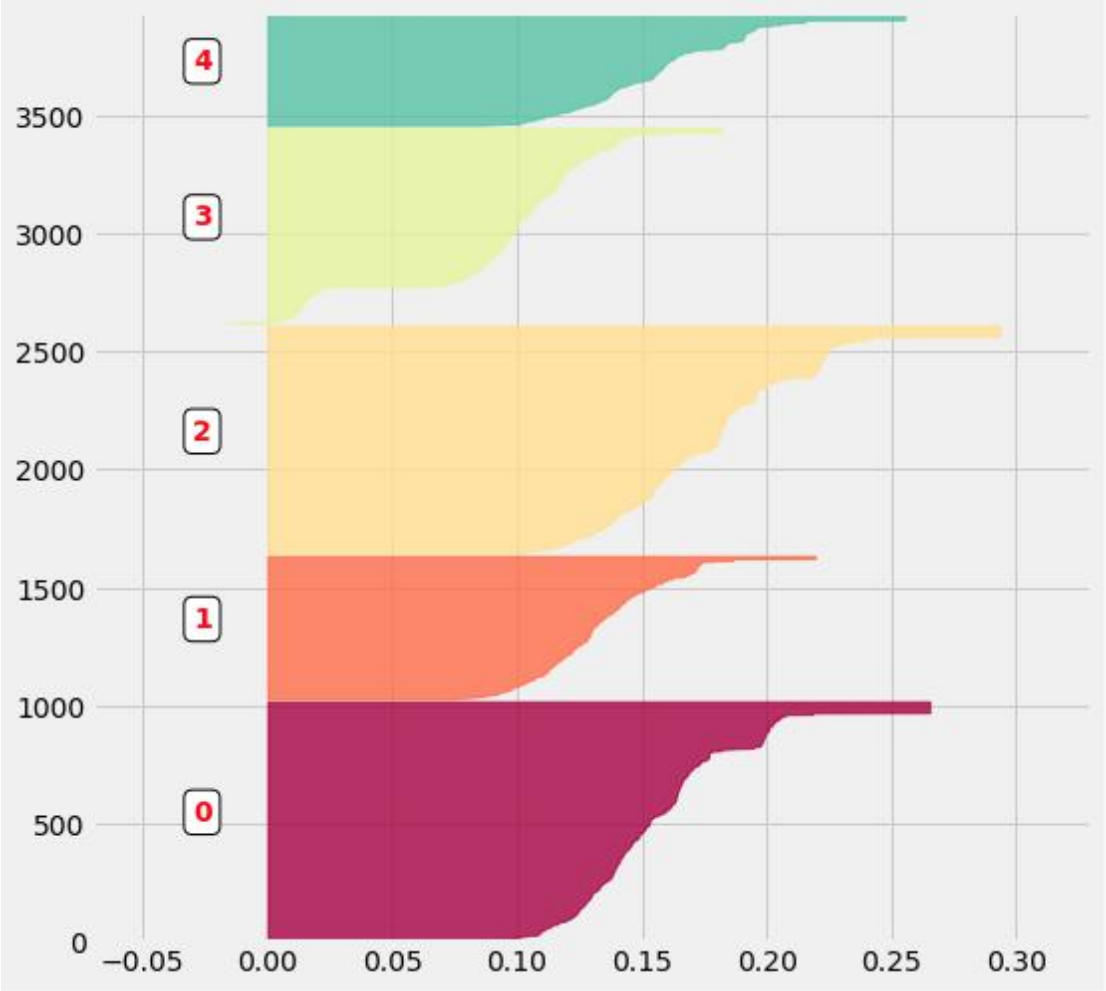
Sử dụng Kmeans từ sklearn để phân cụm sản phẩm thành các lớp khác nhau, ma trận X là một **ma trận nhị phân** nên việc chạy thuật toán kmeans là dễ dàng làm được. Bên cạnh Kmeans còn một thuật toán để phân vùng khác tốt hơn cho ma trận nhị phân là **Hamming**, nhưng trong phần báo cáo này thuật toán **kmeans** đã đáp ứng được những nhu cầu cơ bản trong việc phân vùng sản phẩm và vì kmeans cũng sử dụng thuật toán tính khoảng cách Euclidean nên có thể tính được trên ma trận nhị phân.

Chạy thử thuật toán **kmeans** với số lượng cụm tăng dần từ **3** đến **10**.

```
For n_clusters = 3 The average silhouette_score is : 0.10071681758064248
For n_clusters = 4 The average silhouette_score is : 0.12609893747265383
For n_clusters = 5 The average silhouette_score is : 0.1466257603527048
For n_clusters = 6 The average silhouette_score is : 0.14683518501631235
For n_clusters = 7 The average silhouette_score is : 0.13753408821830407
For n_clusters = 8 The average silhouette_score is : 0.1391956553108947
For n_clusters = 9 The average silhouette_score is : 0.13802163935265813
```

Điểm **silhouette** của các cụm trên là gần như bằng nhau vì không chênh lệch nhau quá nhiều, riêng cụm **5** và **6** có số điểm cao nhất. Quyết định chọn cụm 5 vì nếu chênh nhau không quá nhiều thì cụm số 6 sẽ có rất ít sản phẩm dẫn đến dư thừa.

Hình bên dưới hiển thị số điểm silhouette của từng cụm từ 0 đến 4.



Dễ dàng thấy được là không có cụm nào có số điểm **âm**, dấu hiệu cho thấy việc phân cụm đã cho kết quả tốt khi không có sự dư thừa khi phân bổ trong từng cụm là đều nhau.

Thống kê lại những từ sử dụng trong description của từng cụm thì ta có được word cloud của cả 5 cụm như sau.

Có một số từ xuất hiện ở trong nhiều cụm khác nhau, nhưng về cơ bản là ở mỗi cụm dễ thấy được sự khác nhau riêng biệt với các cụm khác. Cụm 2 nghiêng về các mặt hàng giáng sinh, cụm 4 nghiêng về các mặt hàng xa xỉ, ...

6.5. Phân cụm người dùng.

6.5.1 Phân tích người dùng.

Cần thêm 5 cột với tên mỗi cột là categ_**N** với **N** ∈ [0:4] chứa số tiền đã tiêu trong mỗi loại sản phẩm.

	InvoiceNo	Description	categ_product	categ_0	categ_1	categ_2	categ_3	categ_4
0	536365	WHITE HANGING HEART T-LIGHT HOLDER	3	0.0	0.00	0.0	15.3	0.0
1	536365	WHITE METAL LANTERN	1	0.0	20.34	0.0	0.0	0.0
2	536365	CREAM CUPID HEARTS COAT HANGER	1	0.0	22.00	0.0	0.0	0.0
3	536365	KNITTED UNION FLAG HOT WATER BOTTLE	1	0.0	20.34	0.0	0.0	0.0
4	536365	RED WOOLLY HOTTIE WHITE HEART.	1	0.0	20.34	0.0	0.0	0.0

Nhóm dữ liệu lại theo **CustomerID** và **InvoiceNo** để có được giá tiền của từng loại sản phẩm trong mỗi giỏ hàng. **InvoiceDate** là ngày giỏ hàng được thanh toán.

	CustomerID	InvoiceNo	Basket Price	categ_0	categ_1	categ_2	categ_3	categ_4	InvoiceDate
1	12347	537626	711.79	187.2	293.35	23.40	83.40	124.44	2010-12-07 14:57:00.000001024
2	12347	542237	475.39	130.5	169.20	84.34	91.35	0.00	2011-01-26 14:29:59.999999744
3	12347	549222	636.25	330.9	115.00	81.00	109.35	0.00	2011-04-07 10:42:59.999999232
4	12347	556201	382.52	74.4	168.76	41.40	78.06	19.90	2011-06-09 13:01:00.000000256
5	12347	562032	584.91	109.7	158.16	61.30	157.95	97.80	2011-08-02 08:48:00.000000000

Thêm các thuộc tính quan trọng trong mỗi người dùng như sô lần người dùng thanh toán trên website, số tiền ít nhất, nhiều nhất, tổng số tiền mà người dùng dã tiều trên trang web.

	CustomerID	count	min	max	mean	sum	categ_0	categ_1	categ_2	categ_3	cate
0	12347	5	382.52	711.79	558.172000	2790.86	29.836681	32.408290	10.442659	18.636191	8.67
1	12348	4	227.44	892.80	449.310000	1797.24	41.953217	0.000000	38.016069	20.030714	0.0(
2	12350	1	334.40	334.40	334.400000	334.40	48.444976	0.000000	11.692584	39.862440	0.0(
3	12352	6	144.35	840.30	345.663333	2073.98	12.892120	15.711338	0.491808	56.603728	14.3
4	12353	1	89.00	89.00	89.000000	89.00	13.033708	0.000000	0.000000	64.606742	22.3

Cuối cùng thì thêm hai cột **LastPurchase** và **FirstPurchase** đại diện cho số ngày từ ngày cuối cùng người dùng mua sản phẩm đến ngày trang web dừng hoạt động và số ngày từ ngày đầu tiên người dùng mua sản phẩm đến ngày trang web dừng hoạt động.

Hình bên dưới thể hiện khoảng thời gian tồn tại của trang web trong tập dữ liệu.

```
2010-12-01 08:26:00 -> 2011-12-09 12:50:00
```

Một trong những mục đích quan trọng nhất của việc phân loại người dùng là tăng doanh thu nhờ việc quảng cáo các mặt hàng cụ thể mà người dùng quan tâm nhất và số lượng người dùng chỉ mua một loại sản phẩm duy nhất chiếm **38.74%,** nghĩa là chiếm hơn 1/3 trên tổng số người dùng.

6.5.2 Mã hóa dữ liệu.

Tạo ma trận sử dụng được cho thuật toán **kmeans** chỉ cần những cột :

- ✓ Count
- ✓ Min
- ✓ Max
- ✓ Mean
- ✓ Categ_0
- ✓ Categ_1
- ✓ Categ_2
- ✓ Categ_3
- ✓ Categ_4

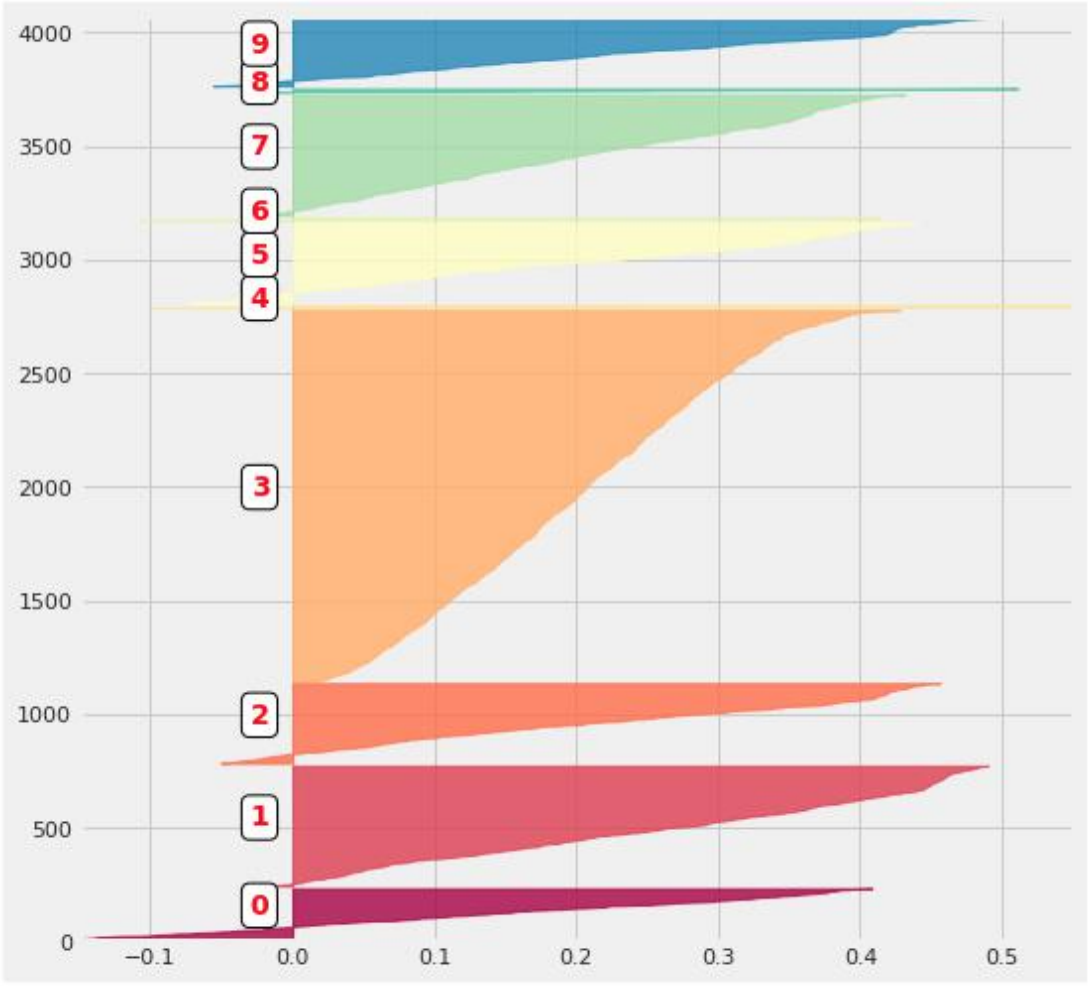
6.5.3 Phân cụm người dùng.

Dùng ma trận đã tạo để chạy trên thuật toán kmeans với n_clusters ∈ [3,19]

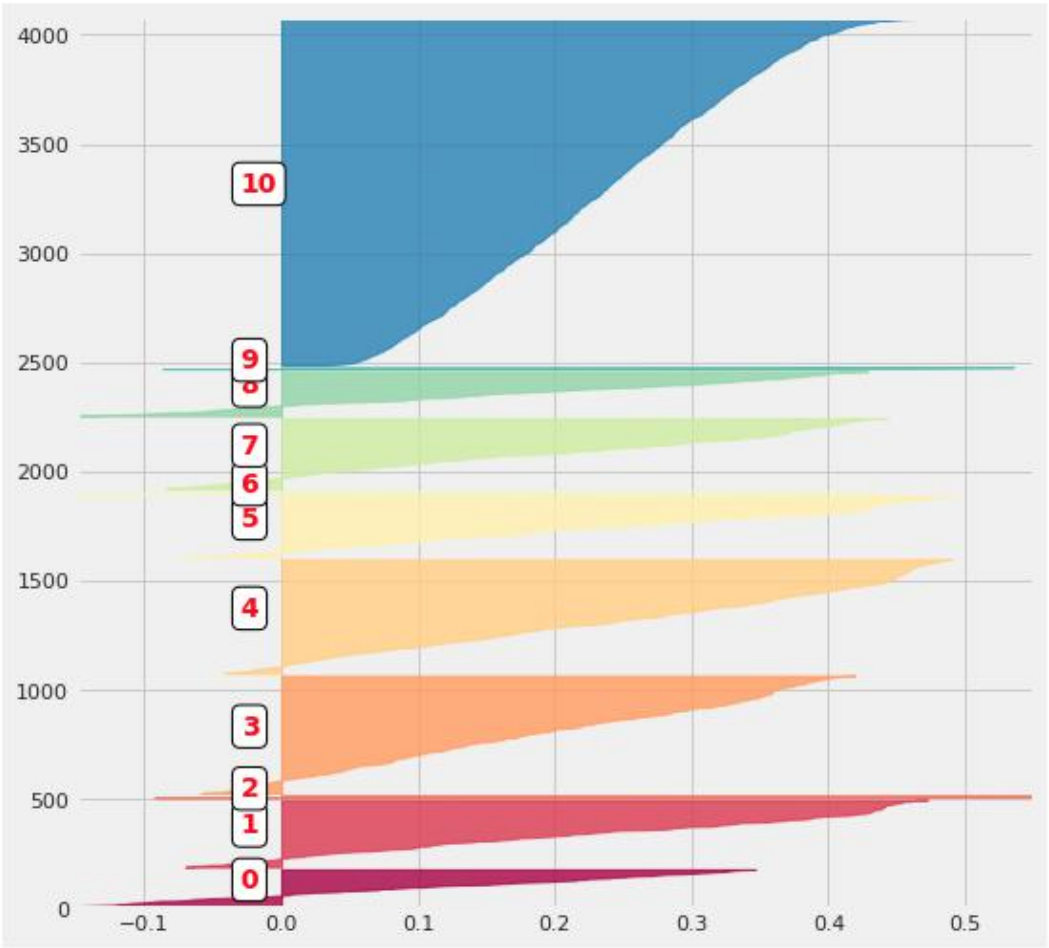
```
For n_clusters = 3 The average silhouette_score is : 0.15634333089613026
For n_clusters = 4 The average silhouette_score is : 0.1503040890411225
For n_clusters = 5 The average silhouette_score is : 0.16382769182569418
For n_clusters = 6 The average silhouette_score is : 0.17478503799867445
For n_clusters = 7 The average silhouette_score is : 0.18892619010777317
For n_clusters = 8 The average silhouette_score is : 0.19414193015286807
For n_clusters = 9 The average silhouette_score is : 0.204643507541903
For n_clusters = 10 The average silhouette_score is : 0.21044757351057622
For n_clusters = 11 The average silhouette_score is : 0.21636577085047481
For n_clusters = 12 The average silhouette_score is : 0.2191366509975317
For n_clusters = 13 The average silhouette_score is : 0.19980159882504012
For n_clusters = 14 The average silhouette_score is : 0.19086359498394836
For n_clusters = 15 The average silhouette_score is : 0.18379624470977843
For n_clusters = 16 The average silhouette_score is : 0.18647185207281852
For n_clusters = 17 The average silhouette_score is : 0.17962742925110692
For n_clusters = 18 The average silhouette_score is : 0.17855075766629927
For n_clusters = 19 The average silhouette_score is : 0.17056193980143972
```

Nhận thấy với số lượng cụm bằng 10, 11, 12 có số điểm thuộc vào nhóm cao nhất.

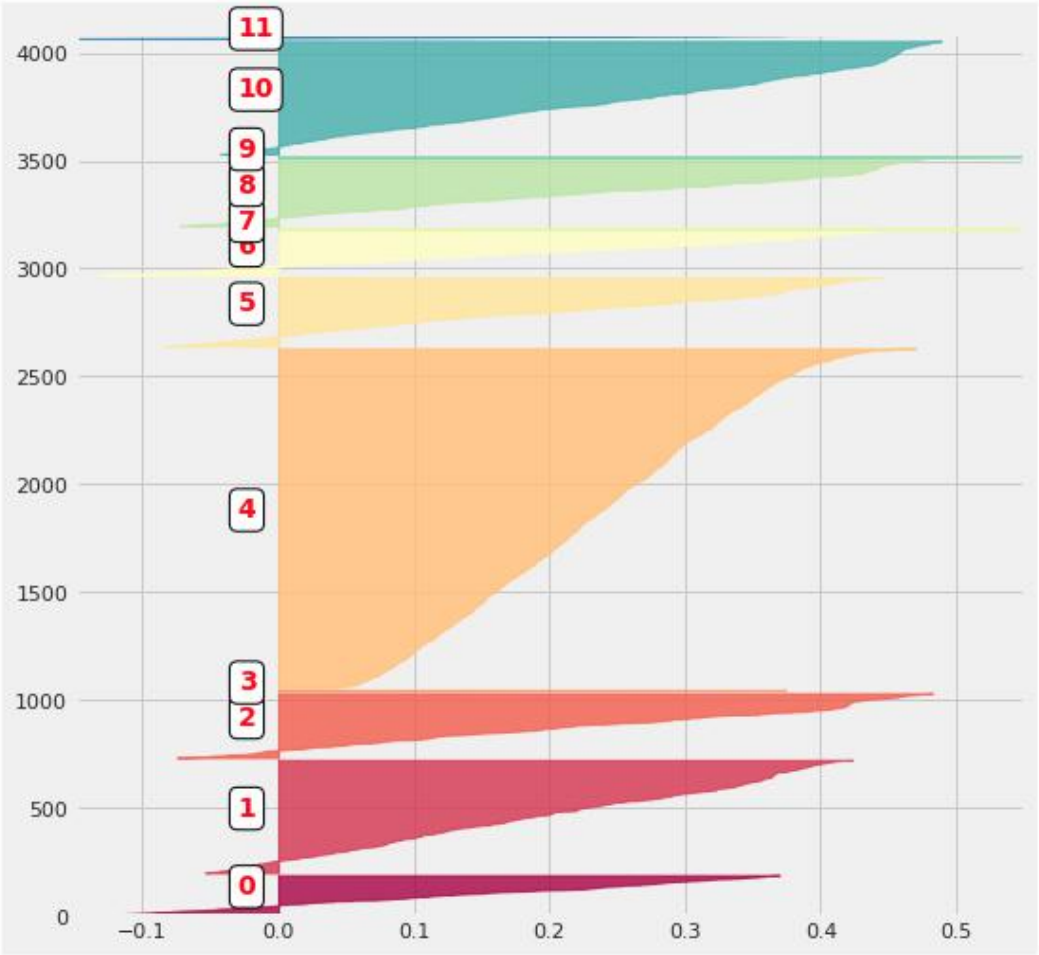
Mô phỏng điểm silhouette với n_cluster = 10



Mô phỏng điểm silhouette với n_cluster = 11



Mô phỏng điểm silhouette với n_cluster = 12



Với cụm 11 và 12 thấy có kết quả tốt hơn so với cụm 10 dù không rõ ràng , riêng khi so sánh giữa cụm 11 và 12 thì một điều lưu ý là số lượng cụm dư thừa ở cụm 11 ít hơn 12 nên ta quyết định chọn số cụm người dùng là 11.

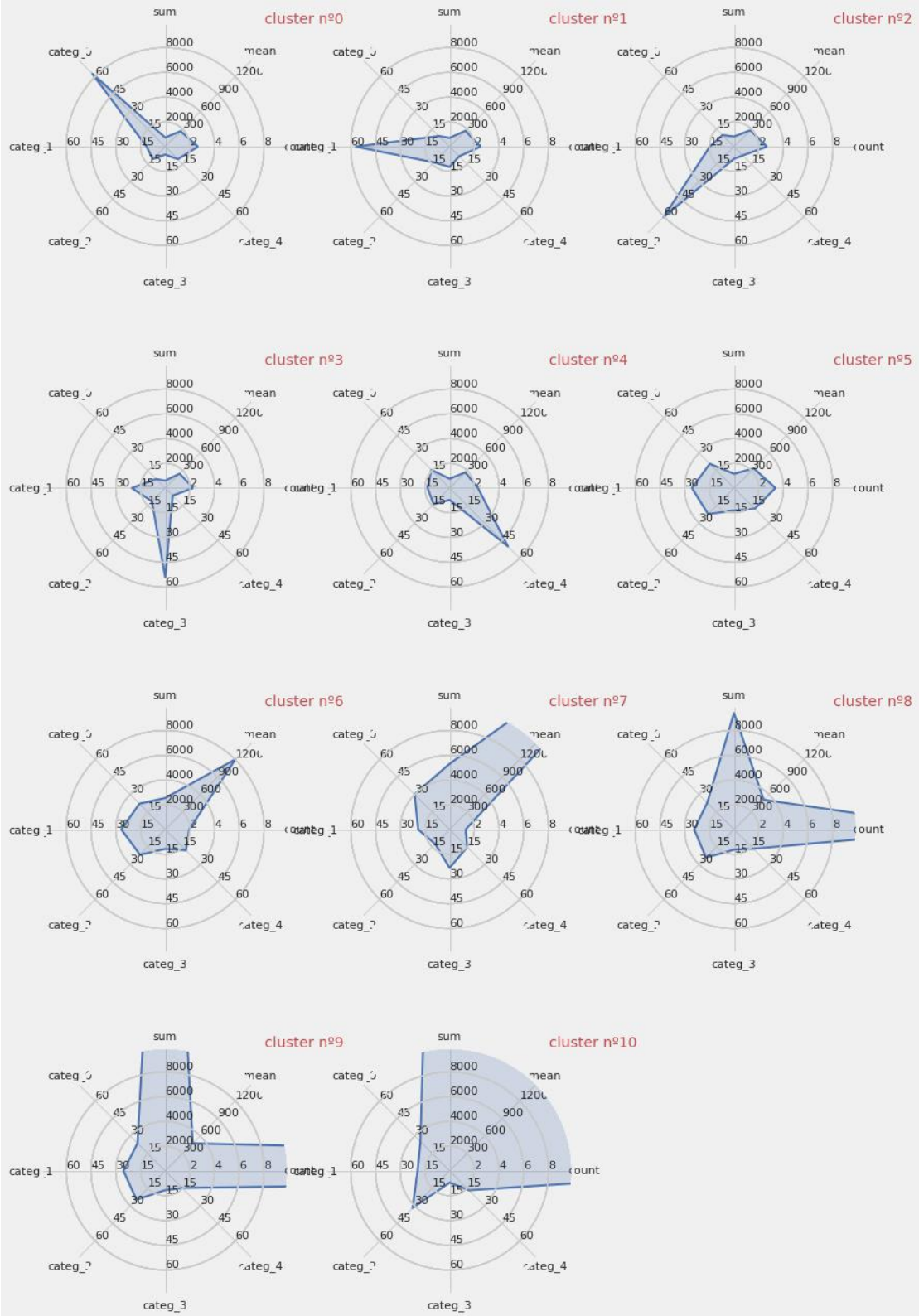
Thống kê số người dùng trong mỗi cụm.

	5	0	7	3	1	4	8	6	9	2	10
Số lượng người dùng	1551	535	514	352	323	284	204	171	13	10	8

Dữ liệu tổng quát của **11** cụm người dùng.

	cluster	count	min	max	mean	sum	categ_0	categ_1	categ_2	categ_3	categ_4	size
0	7.0	2.646272	202.797593	346.153537	263.880052	732.302277	62.499746	11.270564	10.566737	4.806266	10.867477	523
1	2.0	2.527002	210.200130	342.742590	272.886213	710.030393	9.085103	55.521425	13.583174	13.258063	8.556013	537
2	9.0	2.618590	205.541859	369.945962	284.023902	814.504199	9.449740	14.726292	61.164174	7.366741	7.293052	312
3	1.0	2.228571	192.036607	319.094536	247.121099	585.904929	7.829066	19.236901	11.340135	55.252072	6.341826	280
4	6.0	2.282609	200.270373	339.101677	262.119409	706.260155	14.871049	13.366774	13.464596	7.155058	51.187103	322
5	3.0	3.375396	216.184021	483.511256	337.587822	1157.861960	21.087399	25.042885	22.443394	13.447794	17.983209	1577
6	0.0	1.882629	987.419812	1407.160005	1179.439423	2363.990521	22.121846	26.741260	21.279472	11.886074	17.971665	213
7	4.0	1.454545	3743.075455	3916.051818	3827.643939	5533.385455	28.554160	20.474943	13.254647	22.971435	14.744815	11
8	8.0	19.211765	75.195765	1482.027471	528.513152	9977.270176	23.027945	23.871857	24.184970	12.129779	16.800344	170
9	10.0	103.571429	10.985714	1858.250000	370.712472	39586.580000	24.651399	25.798489	22.316637	13.341834	13.915817	7
10	5.0	25.461538	357.272308	16403.534615	4214.171528	87402.536923	25.052191	19.439446	31.768919	7.129122	16.610322	13

Những radar chart bên dưới thể hiện hành vi người dùng



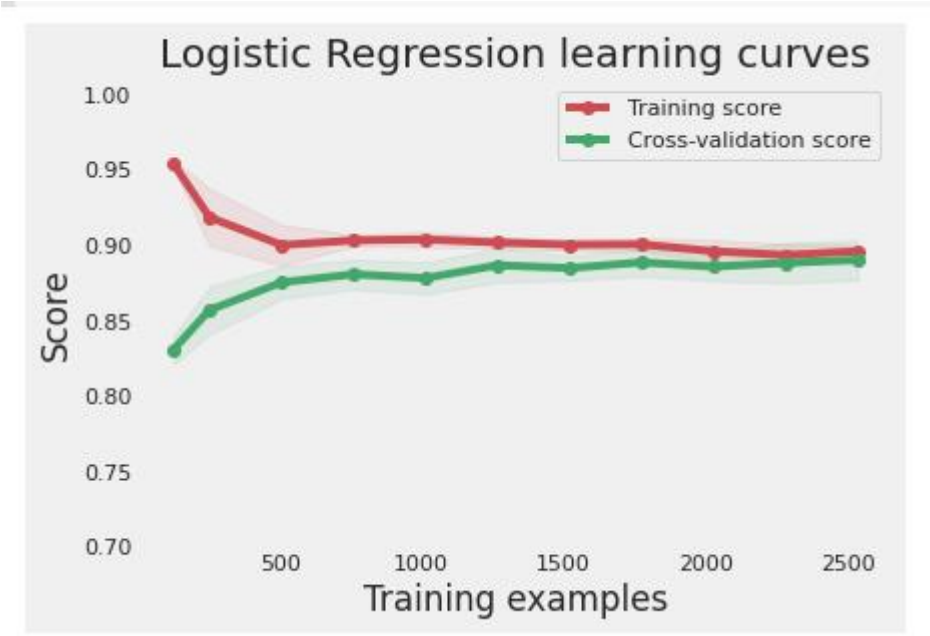
- ✓ Thấy được 5 cụm người dùng đầu tiên là những người dùng có thiên hướng chỉ mua một sản phẩm cụ thể trong suốt quá trình sử dụng website.
- ✓ Cụm người dùng có thói quen chi tiêu đa dạng nghiêng về **mean.**
- ✓ Cụm người dùng có số lần mua nhiều trên nền tảng thì sẽ nghiêng về **count**.
- ✓ Cụm chi tiêu nghiên về **sum** là những người chi tiêu nhiều khi sử dụng nền tảng.

6.6 Kiểm thử mô hình.

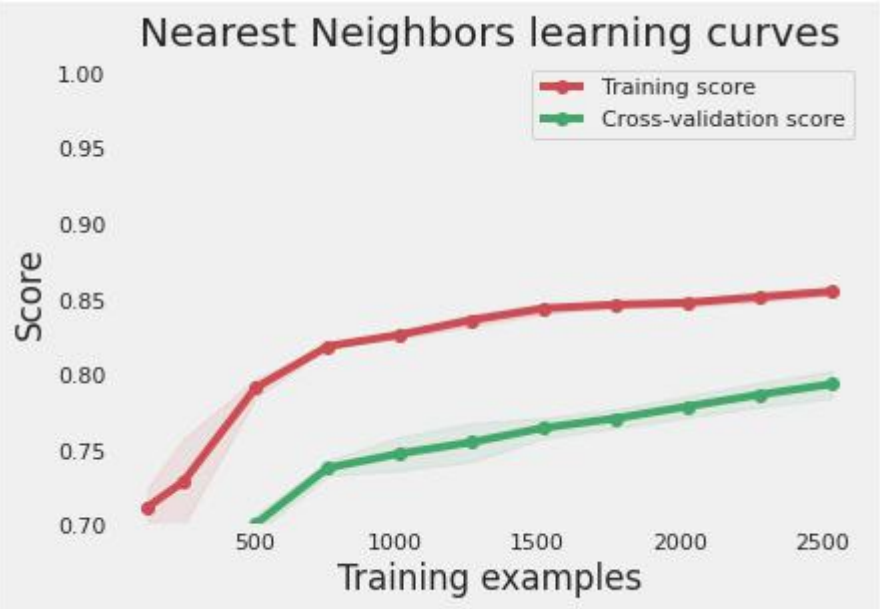
Sử dụng learning curve thường được sử dụng để đánh giá mô hình, dự đoán về khả năng mô hình bị overfiting, underfiting.

- ✓ Nếu đường đi của training score và cross-validation score đều thấp nghĩa là mô hình đã bị **underfit**.
- ✓ Nếu đường đi của training score cao và cross-validation score thấp nghĩa là mô hình đã bị **overfit**.

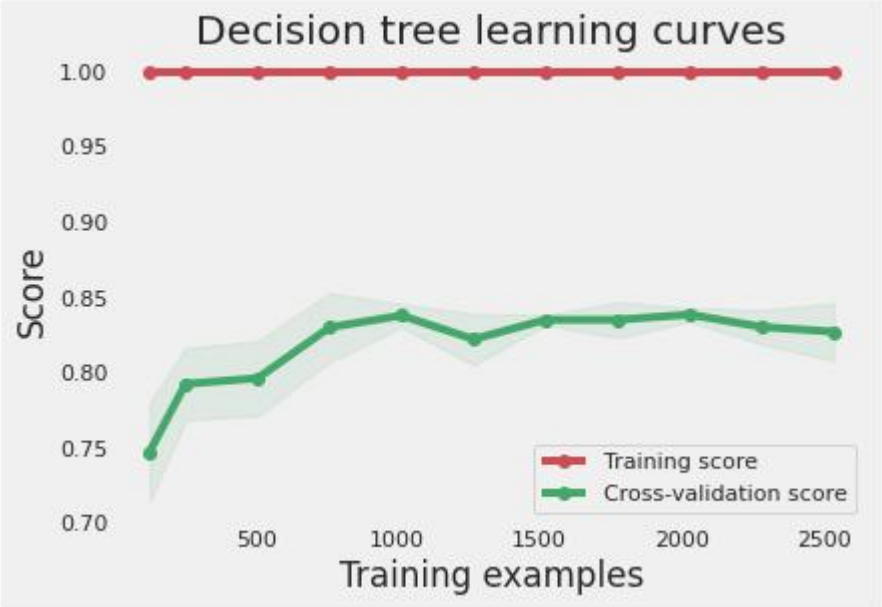
Thử nghiệm trên thuật toán **logistic regression** cho độ chính xác 86.29%.



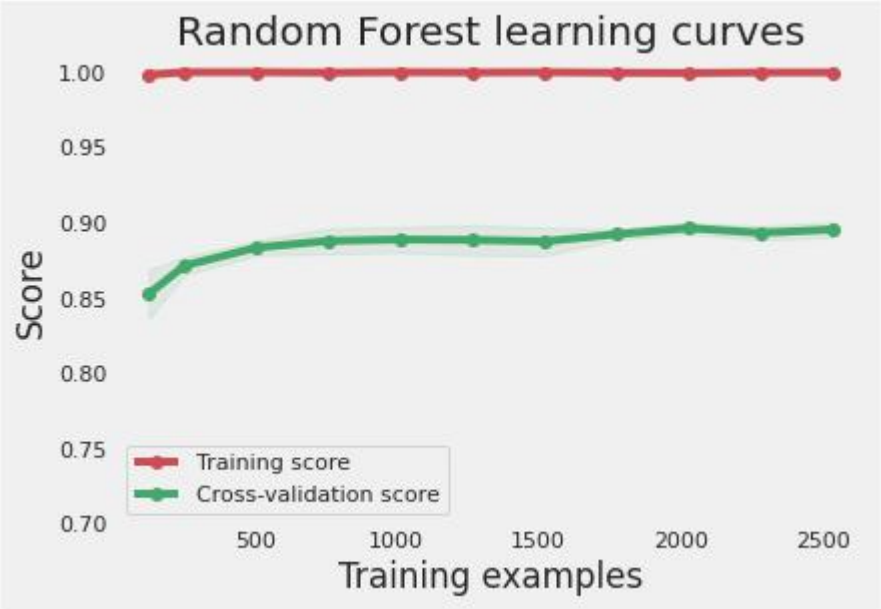
Thử nghiệm trên thuật toán **KNN** cho độ chính xác 79.78%.



Thử nghiệm trên thuật toán **decision tree** cho độ chính xác 86.13%.



Thử nghiệm trên thuật toán **random forest** cho độ chính xác 92.56%



Thử nghiệm trên thuật toán **Adaboost** cho độ chính xác 51.20%