

BAN CƠ YẾU CHÍNH PHỦ
HỌC VIỆN KỸ THUẬT MẬT MÃ



ĐỒ ÁN TỐT NGHIỆP

**NGHIÊN CỨU KỸ THUẬT PHÂN CỤM K-MEANS VÀ ỨNG
DỤNG CHO BÀI TOÁN PHÂN CỤM DỮ LIỆU TỰ ĐỘNG**

Ngành: An toàn thông tin

Mã số: 7.48.02.02

Sinh viên thực hiện:

Đoàn Đức Mạnh

Lớp: AT13CU

Người hướng dẫn:

ThS. Thái Thị Thanh Vân

Đơn vị: Học viện Kỹ thuật mật mã

Hà Nội, 2021

BAN CƠ YẾU CHÍNH PHỦ
HỌC VIỆN KỸ THUẬT MẬT MÃ



ĐỒ ÁN TỐT NGHIỆP

**NGHIÊN CỨU KỸ THUẬT PHÂN CỤM K-MEANS VÀ ỨNG
DỤNG CHO BÀI TOÁN PHÂN CỤM DỮ LIỆU TỰ ĐỘNG**

Ngành: An toàn thông tin

Mã số: 7.48.02.02

Sinh viên thực hiện:

Đoàn Đức Mạnh

Lớp: AT13CU

Người hướng dẫn:

ThS. Thái Thị Thanh Vân

Đơn vị: Học viện Kỹ thuật mật mã

Hà Nội, 2021

MỤC LỤC

Lời nói đầu	iv
Danh mục kí hiệu và viết tắt	vii
Danh mục hình vẽ	viii
CHƯƠNG 1: TỔNG QUAN VỀ KHAI PHÁ DỮ LIỆU VÀ PHÂN CỤM DỮ LIỆU	1
1.1. Khai phá dữ liệu.....	1
1.1.1. Khái niệm:	1
1.1.2. Phát hiện tri thức trong cơ sở dữ liệu (KDD):	2
1.1.3. Các nhiệm vụ chính trong khai phá dữ liệu	3
1.1.4. Ứng dụng khai phá dữ liệu trong thực tế	4
1.2. Bài toán phân cụm dữ liệu	4
1.2.1. Phân cụm dữ liệu	4
1.2.2. Các loại phân cụm dữ liệu.....	9
1.3. Ứng dụng của phân cụm dữ liệu trong thực tế	15
CHƯƠNG 2 : THUẬT TOÁN K-MEANS VÀ BÀI TOÁN PHÂN CỤM DỮ LIỆU TỰ ĐỘNG.....	16
2.1. Thuật toán phân cụm K-means	16
2.1.1. Khái quát thuật toán	16
2.1.2. Mô tả thuật toán.....	19
2.1.3. Phương pháp chọn số cụm	21
2.2. Ứng dụng của thuật toán K-means trong bài toán phân cụm dữ liệu tự động.	24
2.2.1. K-means và phân cụm dữ liệu tự động	24
2.2.2. Phân cụm văn bản	25
2.2.3. Một số ứng dụng khác	33
CHƯƠNG 3: ỨNG DỤNG THUẬT TOÁN K-MEANS CHO BÀI TOÁN PHÂN CỤM LỊCH SỬ TÌM KIẾM CA NHÂN	34
3.1 Mô tả bài toán	34

3.1.1.	Ứng dụng của bài toán:	34
3.1.2.	Mô tả bài toán.....	35
3.2	Thu thập dữ liệu lịch sử tìm kiếm cá nhân	35
3.2.1.	Lịch sử truy cập web cá nhân.....	35
3.2.2.	Ứng dụng thu thập lịch sử tìm kiếm cá nhân	37
3.2.3.	Thu thập thông tin về từ khóa	38
3.3	. Xử lý dữ liệu văn bản với thuật toán Tf-idf và bộ thư viện NLTK.....	40
3.3.1.	Khái quát về thuật toán Tf-idf.....	40
3.3.2.	Xử lý dữ liệu với NLTK (Natural Language Toolkit):	40
3.3.3.	Áp dụng thuật toán Tf-idf.....	41
3.4	. Áp dụng thuật toán K-means	42
	Tài liệu tham khảo.....	45

LỜI CẢM ƠN

Xin chân thành cảm ơn Cô Ths. Thái Thị Thanh Vân – Giảng viên khoa công nghệ thông tin – Học viện Kỹ thuật Mật mã. Cảm ơn cô đã quan tâm, theo sát và hướng dẫn giúp em hoàn thành đồ án tốt nghiệp này.

Em xin cảm ơn các thầy cô trong trường trong quá trình giảng dạy đã truyền đạt cho em nhiều kiến thức và các góp ý trong quá trình học tập và trong cả cuộc sống.

Trong quá trình làm đồ án còn nhận được sự giúp đỡ của các bạn sinh viên khác trong trường. Xin cảm ơn các bạn đã giúp đỡ đưa ra những lời khuyên để phục vụ cho đồ án cũng như công việc sau này.

Em xin chân thành cảm ơn!

SINH VIÊN THỰC HIỆN ĐỒ ÁN

Đoàn Đức Mạnh

LỜI NÓI ĐẦU

Trong thời kỳ công nghệ ngày càng phát triển trên toàn thế giới và cụ thể là cách mạng công nghiệp 4.0, khi mà dữ liệu ngày càng quan trọng và tồn tại trong mọi lĩnh vực công việc khác nhau đặc biệt là thương mại và sản xuất. Việc áp dụng khoa học công nghệ vào quá trình xử lý và phân tích dữ liệu là vô cùng quan trọng và cần thiết, mang tính thách thức cao. Chính vì thế khai phá dữ liệu xuất hiện thu hút các nhà nghiên cứu phát triển, mang tới một xu hướng mới tiếp cận nhằm khai thác các giá trị từ dữ liệu. Việc ứng dụng khai phá dữ liệu giúp các ngân hàng có thể xây dựng mô hình rủi ro tín dụng, giúp các nhà sách phân loại độc giả dự đoán các xu hướng đọc sách và trong rất nhiều các lĩnh vực khác.

Dữ liệu là một phần tất yếu có trong tất cả các công ty. Dữ liệu thường lớn, phong phú đa dạng và liên quan đến nhiều đối tượng khác nhau. Điều này gây khó khăn cho các công ty để có thể lấy được các thông tin cần thiết. Phương án đề ra là việc phân chia và gom các dữ liệu có tính chất tương tự nhau, giống với việc sắp xếp các cuốn sách có cùng chủ đề trong cùng một giá vậy, điều này sẽ giúp cho việc tìm kiếm sách hoặc có một cái nhìn tổng thể về các loại sách. Các nhà nghiên cứu đã cho ra rất nhiều các thuật toán với các mục đích phân cụm khác nhau có thể kể đến như: DBSCAN thuật toán phân cụm dựa trên mật độ, hay STING thuật toán phân cụm dựa trên các ô lưới. Và không thể không nhắc tới K-means, một thuật toán kinh điển trong việc phân cụm dữ liệu thành các phần riêng biệt và là thuật toán đầu tiên tìm được trong các bài hoặc sách hướng dẫn về machine learning.

Dữ liệu không chỉ quan trọng với công ty, nó còn là một phần không thể thiếu trong cuộc sống của mỗi cá nhân. Những dữ liệu đó có thể là tài liệu học tập, là các bản ghi chép cá nhân hay nhạy cảm hơn là lịch sử truy cập web. Trong cuộc sống gắn liền với Internet của mỗi người thì không ngoa nếu nói rằng biết được lịch sử web của một người sẽ hiểu được tính cách cũng như sở thích của người đó. Khi muốn điều tra một người, việc nắm được các thông tin cơ bản của người đó là bước đầu tiên, khi điều tra sâu hơn về công việc, tính

cách và sở thích thì công việc sẽ vô cùng khó. Nhưng nếu có được lịch sử web của người đó thì hoàn toàn có thể đoán và đánh giá được. Tuy nhiên, bởi lịch sử web gắn liền với công việc và sở thích của một người nên nó chứa nhiều thông tin và thường là một dữ liệu phức tạp. Do vậy cần có một ứng dụng có thể tóm gọn, phân tích và đưa ra một cái nhìn khái quát về dữ liệu trên, và việc tóm tắt và tổng hợp cũng là một chức năng của khai phá dữ liệu nói chung và phân cụm dữ liệu nói riêng.

Tầm quan trọng của dữ liệu là rất lớn cả trong công việc và trong cả cuộc sống thường ngày. Tuy nhiên, việc nhận thức về an toàn dữ liệu của mỗi người là chưa tốt. Bên cạnh đó, ứng dụng của việc khai phá dữ liệu cụ thể là phân nhóm dữ liệu hiện nay tại Việt Nam chưa phổ biến. Đó là lý do em chọn đề án này **“Nghiên cứu kỹ thuật phân cụm K-means và ứng dụng cho bài toán phân cụm dữ liệu tự động”**. Mục đích của đề án là nghiên cứu về bài toán phân cụm dữ liệu, thuật toán K-means và ứng dụng để xây dựng một ứng dụng phân cụm dữ liệu lịch sử tìm kiếm trên Google sử dụng thuật toán K-means.

Ngoài phần mở đầu, tài liệu tham khảo và kết luận, nội dung đề án bao gồm 3 chương:

Chương 1: Tổng quan về khai phá dữ liệu và phân cụm dữ liệu:

Chương này trình bày những nội dung cơ bản, cung cấp một cái nhìn khái quát về khai phá dữ liệu, các chức năng và ứng dụng của khai phá dữ liệu. Đồng thời phần này sẽ mô tả bài toán phân cụm trong khai phá dữ liệu

Chương 2: Thuật toán K-means và bài toán phân cụm dữ liệu tự động: Chương này mô tả về thuật toán K-means, cách thức hoạt động và ứng dụng của thuật toán. Chương này cũng đưa ra các thuật toán, phương pháp nhằm cải tiến thuật toán K-means. Bên cạnh đó, chương này mô tả chi tiết bài toán phân cụm dữ liệu và ứng dụng của nó trong việc phân tích, tìm ra tri thức trong dữ liệu. Từ đó đưa ra các yêu cầu với thuật toán K-means trong bài toán phân cụm dữ liệu tự động

Chương 3: Ứng dụng thuật toán K-means cho bài toán phân cụm lịch sử tìm kiếm cá nhân: Đưa ra chương trình phân cụm dữ liệu dựa trên mô hình

phân cụm tài liệu (Document clustering). Dữ liệu cụ thể là lịch sử tìm kiếm Google của một cá nhân. Từ đó thấy được tầm quan trọng của dữ liệu và việc khai phá dữ liệu đem lại nhiều lợi ích không chỉ trong công nghiệp và thương mại mà còn trong nhiều lĩnh vực khác trong cuộc sống.

DANH MỤC KÍ HIỆU VÀ VIẾT TẮT

NLTK	The Natural Language Toolkit
KDD	Knowledge Discovery in Database
HC	Hierarchical Clustering
PAM	Partitioning Around Medoid
STING	Statistical information grid-based
OPTIGRID	Optimal grid
DBSCAN	Density-based Spatial Clustering of Applications with Noise
SOM	Self-organizing map
SOFM	hay self-organizing feature map
ANN	Neural Networks
AI	Artificial Intelligence
CBOW	Common Bag Of Words
BERT	Bidirectional Encoder Representations from Transformers
IDF	Inverse Document Frequency
TF	Term Frequency
NSP	Next Sentence Prediction
MLM	Masked Language Model
NLP	Natural Language Processing

DANH MỤC HÌNH VẼ

Hình 1: Các bước phát hiện tri thức trong cơ sở dữ liệu (KDD)	3
Hình 2: Phân cụm dữ liệu 2 chiều	8
Hình 3: Phân cụm tổng hợp.....	9
Hình 4: Phân cụm phân cấp	10
Hình 5: Phân cụm dựa trên lưới (Grid based Clustering)	12
Hình 6: DBSCAN.....	14
Hình 7: Phân cụm K-means	17
Hình 8: Regression trong Supervised learning	19
Hình 9: Cách thức thuật toán K-means thực hiện phân cụm	20
Hình 11: Tọa độ các điểm ví dụ	21
Hình 13: Biểu đồ thể hiện số K và tổng khoảng cách các điểm đến trung tâm cụm.....	22
Hình 15: Biểu đồ phân tích Silhouette score.....	24
Hình 17: Các bước trong thuật toán Tf-idf	27
Hình 16: Ứng dụng K-means trong việc nén ảnh	33
Hình 18. Sơ đồ các bước thực hiện bài toán phân cụm dữ liệu lịch sử tìm kiếm ..	35
Hình 19. Phương pháp lọc, trích xuất từ khóa từ url tìm kiếm Google	38
Hình 20: Phương pháp tạo list địa chỉ tìm kiếm Google mới	39
Hình 21. File des.txt chứa mô tả của hơn 400 từ khóa trong file key.txt	39
Hình 22: Phương pháp xử lý dữ liệu với thư viện nltk	41
Hình 23: Hàm TfidfVectorizer	41
Hình 24: Sử dụng phương pháp Elbow chọn K.....	42
Hình 25: Biểu đồ thể hiện số K và tổng khoảng cách giữa các điểm tới trung tâm trong cụm.....	42
Hình 26. Kết quả sau khi phân cụm	43

CHƯƠNG 1: TỔNG QUAN VỀ KHAI PHÁ DỮ LIỆU VÀ PHÂN CỤM DỮ LIỆU

1.1. Khai phá dữ liệu

1.1.1. Khái niệm:

Trong thời kỳ công nghệ số hiện nay, mọi hoạt động đều ít nhiều có sự can thiệp của công nghệ. Nó giải quyết được các vấn đề trong cuộc sống, trong công việc đặc biệt là lĩnh vực công nghiệp và thương mại. Khi đó dữ liệu số ngày trở nên quan trọng. Công nghệ phát triển lớn cùng với sự bùng nổ của thời đại Internet dẫn đến việc các dữ liệu ngày càng khổng lồ và phức tạp. Dữ liệu là một yếu tố quan trọng trong kinh doanh, công nghiệp vì nó giúp doanh nghiệp cải thiện kết quả kinh doanh, đưa ra các chiến lược tốt với thị trường. Bên cạnh đó, máy học xuất hiện mang theo những phương pháp giúp các doanh nghiệp, công ty đưa ra các quyết định, dự đoán một cách nhanh và chính xác dựa vào dữ liệu.

Lĩnh vực khai thác dữ liệu đã có những bước tiến nhanh chóng trong hai thập kỷ qua, đặc biệt là từ quan điểm của cộng đồng khoa học máy tính. Quá trình phân tích dữ liệu đã được nghiên cứu rộng rãi trong lĩnh vực xác suất và thống kê. Thông thường, khai phá dữ liệu là một thuật ngữ được đặt ra bởi cộng đồng thiên về khoa học máy tính. Đối với các nhà khoa học máy tính, các vấn đề như khả năng mở rộng, khả năng sử dụng và triển khai tính toán là cực kỳ quan trọng.

Lượng dữ liệu khổng lồ là kết quả trực tiếp của những tiến bộ trong công nghệ và việc tin học hóa mọi khía cạnh của cuộc sống hiện đại. Do đó, việc cần thiết là kiểm tra xem liệu người ta có thể trích xuất những hiểu biết ngắn gọn và xử lý từ dữ liệu có sẵn cho các mục tiêu dành riêng cho ứng dụng hay không. Ví dụ: dữ liệu được thu thập có thể được lấy từ các nguồn không đồng nhất ở các định dạng khác nhau và bằng cách nào đó cần được xử lý bằng chương trình máy tính tự động để có được thông tin chi tiết. Để giải quyết vấn đề này, các nhà phân tích khai thác dữ liệu sử dụng một quy trình xử lý, nơi dữ liệu thô được thu thập, làm sạch và chuyển đổi thành một định dạng chuẩn hóa. Dữ liệu có thể được lưu trữ trong một hệ thống cơ sở dữ liệu thương mại và cuối cùng được xử

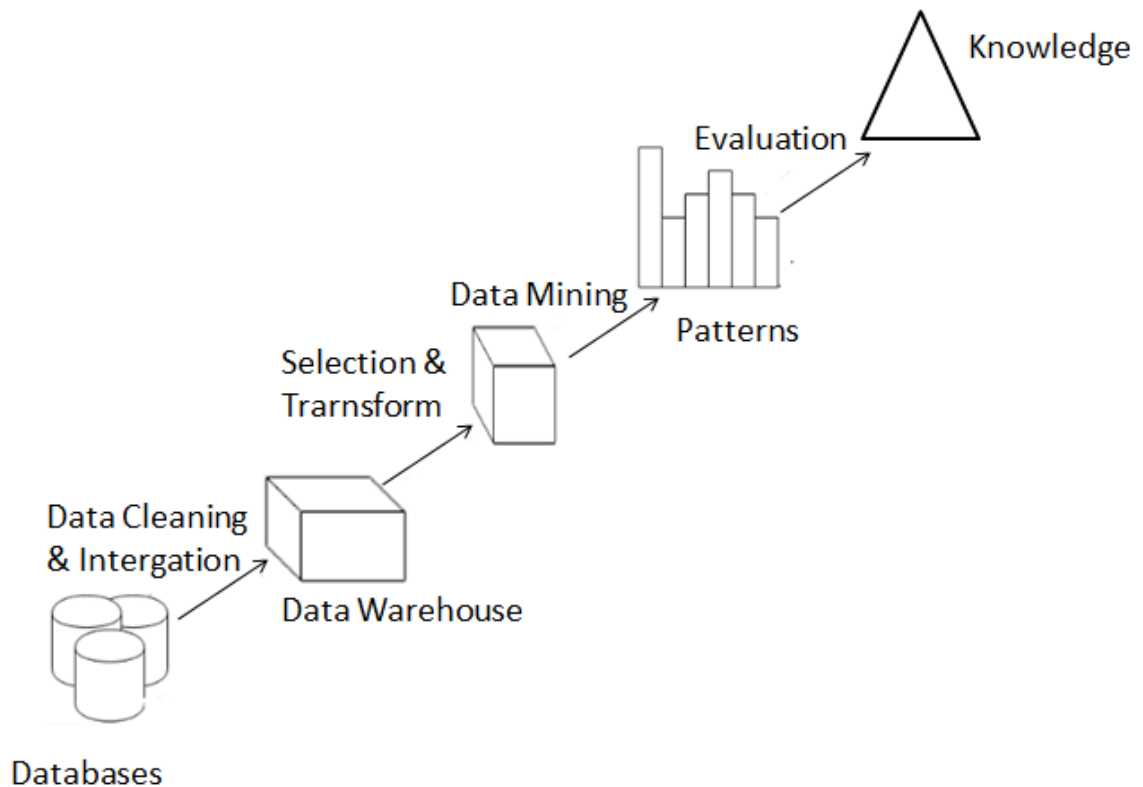
lý để hiểu rõ hơn với việc sử dụng các phương pháp phân tích. Quy trình chế biến này về mặt khái niệm tương tự như quy trình khai thác thực tế từ quặng khoáng sản đến sản phẩm cuối cùng được tinh chế. Thuật ngữ "khai thác" bắt nguồn từ sự tương tự này.

Khai phá dữ liệu là một tập hợp các kỹ thuật được sử dụng để tự động khai thác và tìm ra các mối quan hệ lẫn nhau giữa các dữ liệu trong một tập hợp dữ liệu lớn và phức tạp, đồng thời cũng tìm ra các mâu thuẫn tiềm ẩn trong tập dữ liệu đó. Thuật ngữ Data Mining thể hiện việc tìm kiếm dữ liệu chung có giá trị giữa một dữ liệu lớn và phức tạp.

1.1.2. Phát hiện tri thức trong cơ sở dữ liệu (KDD):

Ngoài thuật ngữ Data Mining, khai phá dữ liệu còn biết tới với một số tên gọi như Knowledge Mining (Khai phá tri thức), Data Analysis (Phân tích dữ liệu),... Các nhà nghiên cứu thì gọi nó là một trong các bước phát hiện tri thức trong cơ sở dữ liệu KDD (Knowledge Discovery in Database KDD). KDD được xem như một quá trình với 7 bước lần lượt nhau như sau:

- Làm sạch dữ liệu (Data cleaning): Các dữ liệu nhiễu, không nhất quán và không cần thiết sẽ được bỏ đi.
- Tích hợp dữ liệu (Data intergation): Quá trình hợp nhất dữ liệu từ nhiều nguồn khác nhau tạo thành kho dữ liệu (Data warehouse) sau khi đã làm sạch ở bước trên.
- Lựa chọn dữ liệu (Data selection): Trích rút các dữ liệu phù hợp với nhiệm vụ phân tích từ kho dữ liệu.
- Chuyển đổi dữ liệu (Data tranform): Dữ liệu được chọn sẽ chuyển đổi sang dạng phù hợp với quá trình khai phá.
- Khai phá dữ liệu (Data mining): Là quá trình quan trọng nhất trong đó sử dụng các phương pháp thông minh để trích rút ra các mẫu dữ liệu
- Đánh giá mẫu (Knowledge evaluation): Đánh giá các kết quả tìm được dựa trên một số độ đo nào đó
- Biểu diễn tri thức (knowledge presentation): Trực quan hóa tri thức tới người dùng bằng các kỹ thuật biểu diễn và thể hiện



Hình 1: Các bước phát hiện tri thức trong cơ sở dữ liệu (KDD)

1.1.3. Các nhiệm vụ chính trong khai phá dữ liệu

Data Mining được chia nhỏ với các bài toán như sau:

- Mô tả khái niệm (concept description): Tóm tắt, mô tả và tổng hợp.
- Các quy tắc kết hợp (Association rules): Cung cấp các quy tắc tương quan tới sự hiện diện của một nhóm dữ liệu này với dữ liệu khác. Ví dụ Giày – Tất, Máy tính – Chuột, kem đánh răng – bàn chải. Ví dụ một người tới cửa hàng mua máy tính thì thông thường sẽ mua thêm chuột, vì thế nên cửa hàng nắm bắt được thông tin và bán thêm chuột tăng doanh số.
- Phân lớp (Classification): Xếp các đối tượng vào lớp đã biết trước hoặc dự đoán một số đối tượng tiếp theo. Phương pháp này được gọi là học có giám sát (Supervised Learning) trong học máy

- Phân cụm (Clustering): Chia các đối tượng ra thành các cụm không biết trước số cụm và tên cụm. Trong học máy phương pháp này được gọi là học không giám sát (Unsupervised learning).
- Khai phá chuỗi (Sequential patterns): Tương tự Association rules nhưng có thêm tính thứ tự và tính thời gian.

1.1.4. Ứng dụng khai phá dữ liệu trong thực tế

Khai phá dữ liệu là một phương pháp mới dựa trên dữ liệu có thể kết hợp với học máy, xác suất thống kê, trí tuệ nhân tạo, cơ sở dữ liệu, trực quan hóa. Thu hút được nhiều các nhà nghiên cứu nhằm đưa ra các mô tả hoặc dự đoán nhờ vào những ứng dụng thực tiễn của nó. Dưới đây là một số ứng dụng điển hình có thể kể đến:

- Phân tích thị trường (Market analysis)
- Tài chính và đầu tư (Finance and finance investments)
- Khai thác văn bản, web (Text mining, web mining)
- Quản lý rủi ro (Risk analysis and management)
- Viễn thông (Telecommunication)
- Bio-informatics
- Quảng cáo (Advertisement)

1.2. Bài toán phân cụm dữ liệu

1.2.1. Phân cụm dữ liệu

a. Các loại dữ liệu

Sự bùng nổ thông tin, cả dạng có cấu trúc và không có cấu trúc đều tạo ra các tập dữ liệu khác nhau và cả số lượng lớn dữ liệu. Trong dữ liệu phi cấu trúc, thông tin không tuân theo một định dạng cụ thể hoặc được tổ chức theo một cách xác định trước và cũng không có một mô hình dữ liệu được xác định trước. Ví dụ, nó thường chứa nhiều văn bản nhưng cũng có thể chứa video, số, âm thanh, hình ảnh, ... Mặt khác, dữ liệu có cấu trúc thường nằm trong cơ sở dữ liệu quan hệ, có các mối quan hệ ngữ nghĩa trong mỗi đối tượng, các trường lưu trữ được phân định theo độ dài dữ liệu số điện thoại, mỗi chuỗi văn bản có độ

dài thay đổi được chứa trong các bản ghi giúp tìm kiếm dễ dàng hơn không giống như dữ liệu không có cấu trúc.

- **Categorical Data (Dữ liệu rõ ràng):** Đây là loại dữ liệu được thu thập trong các nhóm và số lượng sự kiện trong mỗi nhóm được tính bằng số. Dữ liệu này bao gồm số lượng thuộc tính hữu hạn. Ví dụ: chủng tộc, giới tính, nhóm tuổi và trình độ học vấn.
- **Text Data (Dữ liệu văn bản):** Đây là thứ tự các ký tự có thể đọc được của con người được mã hóa thành các định dạng máy tính có thể đọc được như EBCDIC, ASCII, ...
- **Multimedia Data (Dữ liệu đa phương tiện):** Loại dữ liệu bao gồm nhiều loại phương tiện khác nhau như văn bản, âm thanh, video và hình ảnh động. Chúng phụ thuộc vào thời gian và quá trình xử lý của chúng bị giới hạn về thời gian. Chúng là đại diện rời rạc của thực tế hoặc hình ảnh. Tuy nhiên, chúng xuất hiện liên tiếp với quan sát vật lý khi được trình bày thường xuyên và định kỳ ở tần số đủ cao
- **Stream Data (Dữ liệu luồng):** Đây là sự sắp xếp các gói dữ liệu được mã hóa kỹ thuật số được sử dụng để giao tiếp hoặc nhận thông tin trong quá trình được truyền đạt. Dữ liệu luồng có thể được coi như một tập hợp con của dữ liệu đa phương tiện, chẳng hạn như video, âm thanh và hình ảnh động.
- **Uncertain Data (Dữ liệu không chắc chắn):** Đây là loại dữ liệu có liên quan đến nhiều cho phép nó đi chệch khỏi các giá trị đã định, chính xác hoặc ban đầu. Tính không chắc chắn hoặc tính xác thực của dữ liệu là một trong những đặc điểm quan trọng của dữ liệu trong thời đại dữ liệu lớn
- **Time Series Data (Dữ liệu chuỗi thời gian):** Đây là một chuỗi các điểm dữ liệu được lập chỉ mục theo một thứ tự thời gian cụ thể hoặc các khoảng thời gian hoặc khoảng thời gian. Ví dụ: để xác định tỷ lệ thất nghiệp hàng tháng sẽ liên quan đến một chuỗi thời gian. Tỷ lệ thất

nghiệp được xác định rõ ràng và được đánh giá thường xuyên vào các khoảng thời gian xác định hoặc khoảng thời gian cách đều nhau

- Big Data (Dữ liệu lớn): Đây là thuật ngữ được sử dụng để mô tả bất kỳ khối lượng nào của cả dữ liệu có cấu trúc, bán cấu trúc và không có cấu trúc có khả năng được khai thác để có thông tin hữu ích. Nó mô tả các tập dữ liệu không chỉ phức tạp mà còn khổng lồ đến mức mà phần mềm ứng dụng xử lý dữ liệu truyền thống không có khả năng xử lý chúng. Bảo mật thông tin, cập nhật, truy vấn, trực quan hóa truyền, chia sẻ, tìm kiếm, phân tích dữ liệu, lưu trữ dữ liệu và thu thập dữ liệu là những thách thức lớn của dữ liệu lớn

b. Khái niệm phân cụm dữ liệu

Dữ liệu là một phần không thể thiếu trong hoạt động cuộc sống của con người. Từ những dữ liệu lớn như thông tin khách hàng, sản phẩm hay chỉ đơn giản là dữ liệu cá nhân như tài liệu học tập. Dữ liệu càng trở nên quan trọng thì việc sắp xếp nó thành các nhóm sẽ giúp công việc gắn với dữ liệu sẽ trở nên nhanh và hiệu quả hơn.

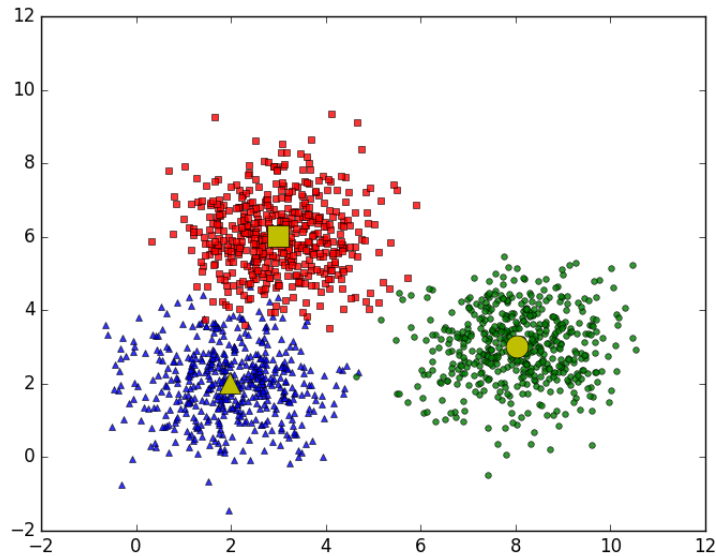
Dữ liệu sẽ vô dụng nếu thông tin hữu ích có thể được sử dụng không thể được suy ra từ nó. Quá trình phân cụm sẽ dựa trên một số tiêu chí, chia sẻ dữ liệu thành các danh mục (cụm) quan trọng, thiết thực hoặc cả hai dựa trên các đặc điểm chung được chia sẻ. Trong nghiên cứu, phân cụm và phân loại đã được sử dụng để phân tích dữ liệu trong lĩnh vực máy học, tin sinh học, thống kê, nhận dạng mẫu. Những thách thức và vấn đề trong phân cụm nảy sinh từ các bộ dữ liệu lớn, như hiểu sai kết quả và hiệu quả của các thuật toán phân cụm, việc hiểu được những thách thức này là cần thiết cho việc lựa chọn các thuật toán phân cụm. Phân cụm dữ liệu có hệ thống dựa trên các đặc điểm của các kỹ thuật phân nhóm khác nhau làm cho chúng phù hợp hơn hoặc ít sai lệch hơn khi áp dụng cho một số loại dữ liệu, chẳng hạn như dữ liệu không chắc chắn, dữ liệu đa phương tiện, dữ liệu đồ thị, dữ liệu sinh học, dữ liệu luồng, dữ liệu văn bản, dữ liệu chuỗi thời gian, dữ liệu phân loại và dữ liệu lớn. Sự phù hợp của các thuật toán phân cụm có sẵn đối với các lĩnh vực ứng dụng khác nhau, một số phương

pháp hợp lệ được sử dụng để đánh giá mức độ phù hợp của các cụm được tạo ra bởi các thuật toán phân cụm.

Bài toán phân cụm dữ liệu (Clustering) từ lâu đã trở thành một vấn đề cần được quan tâm, là một kỹ thuật quan trọng trong khai phá dữ liệu. Trong học máy, Clustering là một phương pháp học không giám sát (Unsupervised learning). Khác với học có giám sát (Supervised learning)_ở đây dữ liệu ban đầu đã có đầu vào và đầu ra cho trước được gọi là dữ liệu huấn luyện, các dữ liệu này sẽ xây dựng lên một hàm huấn luyện và những dữ liệu mới sẽ được dự đoán thông qua hàm huấn luyện trên. Học không giám sát không cho trước đầu ra của dữ liệu ban đầu, trong đó, một tập dữ liệu ban đầu được thu thập sẽ được thuật toán phân tích cấu trúc để thực hiện một công việc nào đó chẳng hạn như việc phân nhóm dữ liệu hoặc giảm số chiều của dữ liệu. Bài toán phân cụm dữ liệu là bài toán tiêu biểu nhất cho phương pháp học máy này.

Phân cụm dữ liệu, còn được gọi là phân loại không giám sát, được định nghĩa là kỹ thuật tạo nhóm đối tượng, sao cho các đối tượng trong một cụm rất giống nhau và các đối tượng trong các cụm khác nhau tương đối khác nhau. Mục tiêu chính của phân nhóm là khám phá tập hợp các mẫu, điểm hoặc đối tượng từ các nhóm tự nhiên. Sự tiến bộ nhanh chóng trong các ứng dụng phân cụm như hình ảnh kỹ thuật số, tìm kiếm trên Internet, giám sát video và những tiến bộ trong công nghệ lưu trữ đã mang lại nhiều tập dữ liệu có dung lượng cao một chiều cao.

Phân cụm là quá trình tìm cách nhóm, chia các dữ liệu đã cho thành các cụm (Clusters) sao cho các đối tượng trong cùng một nhóm có sự tương đồng theo một tiêu chí nào đó. Mục đích của phân cụm là tìm ra bản chất bên trong của dữ liệu. Các bài toán phân cụm đều sinh ra các cụm, tuy nhiên không có phương pháp, tiêu chí nào là tối ưu cho việc đánh giá kết quả của việc phân cụm. Điều này phụ thuộc vào mục đích của bài toán phân cụm như: data reduction, natural clusters, outlier detection...



Hình 2: Phân cụm dữ liệu 2 chiều

Có ba loại phân cụm dữ liệu được phân loại dựa trên mục đích:

- Natural classification (Phân cụm tự nhiên): Xác định mối quan hệ, mức độ giống nhau giữa các sinh vật. Ví dụ: xác định mối quan hệ giữa các loài, sự phát sinh các giống loài.
- Structurally underlying (Dựa trên cấu trúc): Phải dựa trên sự cơ bản về mặt cấu trúc từ đó có thể tạo giả thuyết, phát hiện thị trường và cũng như xác định các đặc điểm nổi bật
- Compression (Nén): Cũng là một kiểu kỹ thuật quan trọng giúp việc tổ chức, và tóm tắt dữ liệu thông qua các cụm.

Trong khai thác dữ liệu và học máy, vấn đề của kỹ thuật phân cụm dữ liệu đã được nghiên cứu rộng rãi trong các lĩnh vực này vì tính hữu ích của nó trong các ứng dụng khác nhau để phân đoạn, tóm tắt, tiếp thị mục tiêu, tin sinh học,... Các vấn đề phân cụm như tính hợp lệ của cụm, tính mạnh mẽ của kỹ thuật phân nhóm, bản chất của dữ liệu và các vấn đề khác đã được giải quyết rộng rãi trong một số ứng dụng như phân đoạn, khai thác văn bản, xử lý hình ảnh, ... Sự hiểu biết về các loại dữ liệu được sử dụng là rất quan trọng nó là yếu tố quyết định chính của phương pháp phân tích dữ liệu sẽ được sử dụng.

Không giống như học có giám sát, phân cụm được coi là một phương pháp học không có giám sát vì nó không có kết quả ban đầu để so sánh kết quả đầu ra

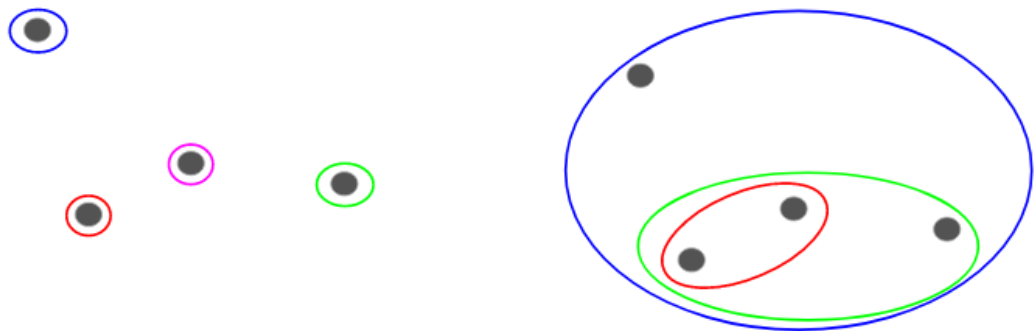
của thuật toán phân cụm với các nhãn thực để đánh giá hiệu suất của nó. Thuật toán chỉ muốn phân tích, tìm ra cấu trúc của dữ liệu bằng cách nhóm các điểm dữ liệu thành các nhóm con riêng biệt.

1.2.2. Các loại phân cụm dữ liệu

a. Hierarchical Clustering (Phân cụm theo phân cấp)

Dạng phân cụm này dựa trên mức độ gần nhau của các điểm dữ liệu của chúng. Hierarchical Clustering (HC) cho phép các cụm con trong một cụm dẫn đến một cụm lồng nhau được tổ chức theo cấu trúc giống cây. Phương pháp này có hai cách tiếp cận chính, chúng bao gồm: Agglomerative (tổng hợp) và Divisive (phân chia thứ bậc)

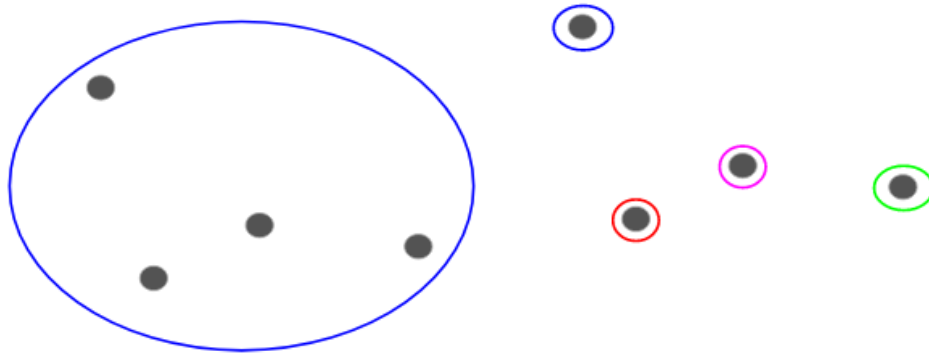
- Agglomerative: phương pháp này được thực hiện bằng cách: ban đầu, đặt mỗi điểm vào một cụm của chính nó, sau đó tìm và kết hợp hai điểm gần nó nhất, một điểm trong trường hợp này đề cập đến một đối tượng riêng lẻ hoặc một cụm đối tượng. Phương pháp này chỉ định mỗi điểm cho một cụm riêng lẻ. Giả sử có 4 điểm dữ liệu, gán mỗi điểm cho một cụm và do đó sẽ có 4 cụm ở đầu. Sau đó, ở mỗi lần lặp lại, hợp nhất cặp cụm gần nhất và lặp lại bước này cho đến khi chỉ còn lại một cụm duy nhất



Hình 3: Phân cụm tổng hợp

- Divisive: Phân cụm phân cấp hoạt động theo cách ngược lại. Thay vì bắt đầu với n cụm, nó bắt đầu với một cụm duy nhất và gán tất cả các điểm cho cụm đó. Vì vậy, không thành vấn đề nếu có 10 hay 1000 điểm dữ liệu. Tất cả những điểm này sẽ thuộc cùng một cụm ở phần đầu. Tiếp theo đó, ở mỗi lần lặp, chia điểm xa nhất trong cụm và lặp lại quá trình

này cho đến khi mỗi cụm chỉ chứa một điểm duy nhất. Tách (hoặc phân chia) các cụm ở mỗi bước, do đó có tên là phân nhóm phân cấp.



Hình 4: Phân cụm phân cấp

b. Partitional Clustering (Phân cụm theo phần)

Partitional Clustering phân tách một tập dữ liệu thành một tập hợp các cụm rời rạc. Tập dữ liệu có N điểm sẽ được phân vùng dữ liệu thành K vùng ($N \geq K$), với mỗi phân vùng đại diện cho một cụm. Phương pháp phân cụm này phân loại dữ liệu thành K nhóm thỏa mã điều kiện: mỗi nhóm chứa ít nhất một điểm và mỗi điểm thuộc đúng một nhóm.

Trong phương pháp này, số lượng cụm k sẽ được cung cấp bởi người dùng, sau đó được cải thiện theo cách lặp lại. Tuy nhiên, thuật toán phân cụm theo phân cấp (hierarchical clustering) được định nghĩa là phương pháp phân cụm bằng cách tạo cấu trúc dữ liệu dựa trên cây nhị phân (binary-tree) được gọi là dendrogram. Mặt khác, phương pháp hierarchical clustering là một cụm lồng nhau tổ chức thông tin về một cây và không yêu cầu giá trị cụ thể của k , không giống như phân cụm không phân cấp. Thuật toán phân cụm phân cấp phát triển phân cụm ở dạng giống cây từ các điểm dữ liệu riêng lẻ trong một cụm duy nhất. Phân cụm theo từng phần vẫn là một trong những kỹ thuật phổ biến và được áp dụng nhiều nhất vì tính đơn giản, hiệu quả, rất dễ thực hiện và thành công trên thực nghiệm của nó.

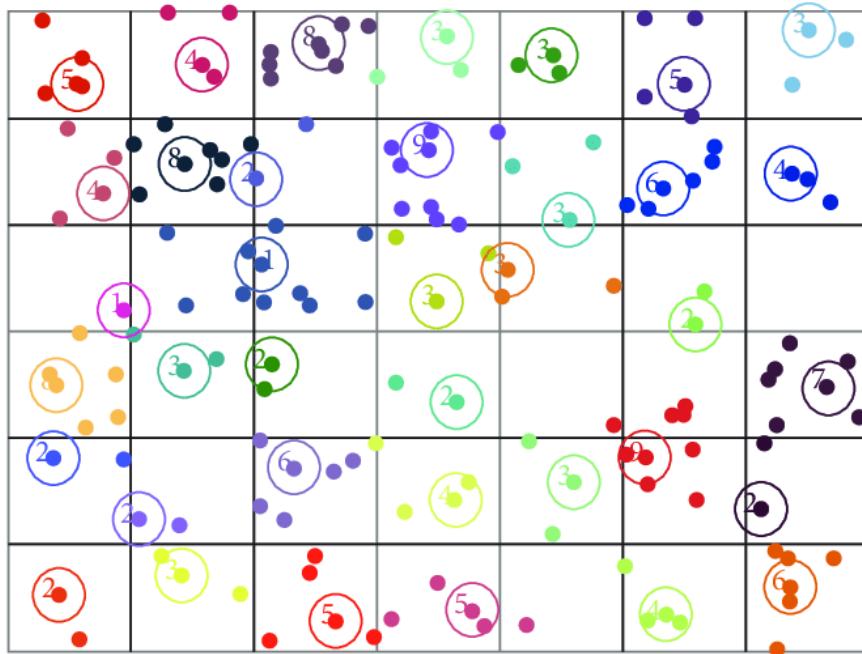
K-means là một thuật toán thuộc phương pháp phân cụm theo phần này. Thuật toán phân chia n điểm dữ liệu thành k cụm, sao cho khoảng cách trung bình giữa các điểm trong cùng một cụm nhỏ nhất. Thuật toán K-means yêu cầu

giá trị k ban đầu xác định số lượng cụm và gán các đối tượng cho các nhóm để giảm thiểu sai số bình phương. Có một số phần mở rộng cho K-means để nâng cao hiệu suất của nó, chẳng hạn như Intelligent Kernel K-means

Ngoài thuật toán K-means, thuật toán PAM (Partitioning Around Medoid) cũng là một thuật toán nổi tiếng trong phương pháp phân cụm theo phần. Thuật toán PAM phân cụm các đối tượng trên m biến tỷ lệ khoảng thời gian nhất định, và cũng có thể được áp dụng khi ma trận dữ liệu đầu vào không giống nhau. Cả thuật toán PAM và K-means đều cố gắng giảm thiểu khoảng cách giữa các điểm được gán nhãn trong một cụm cụ thể và một điểm được chọn làm trung tâm của cụm cụ thể đó. Nhưng PAM mạnh hơn K-means vì tổng các điểm khác biệt mà nó giảm thiểu so với tổng các khoảng cách Euclidean bình phương trong trường hợp K-means. Nó dễ bị ảnh hưởng bởi vấn đề đầu vào ban đầu và cũng không thể tính toán các tập dữ liệu lớn, các cụm được kết nối cao và bộ dữ liệu chiều cao làm cho nó ít yêu cầu hơn để phân nhóm một nhóm dữ liệu.

c. Grid based Clustering (Phân cụm dựa trên lưới, các ô đường kẻ)

Grid based Clustering hiệu quả trong việc khai thác các tập dữ liệu đa chiều lớn. Các thuật toán này phân vùng không gian dữ liệu thành một số ô hữu hạn để tạo thành cấu trúc lưới và sau đó tạo thành các cụm từ các ô trong cấu trúc lưới. Các cụm tương ứng với các vùng có mật độ điểm dữ liệu cao hơn các vùng xung quanh chúng. Phân cụm dựa trên lưới nổi tiếng với việc trích xuất các cụm trong một không gian đa chiều khổng lồ được lượng tử hóa thành một nhóm ô tạo thành cấu trúc lưới mà trên đó tất cả các hoạt động phân cụm được thực hiện.



Clusters = 37

Hình 5: Phân cụm dựa trên lưới (Grid based Clustering)

Phương pháp phân cụm dựa trên lưới này có thể kể đến thuật toán STING (Statistical information grid-based). Thuật toán sử dụng thông tin thống kê để ước tính kết quả truy vấn được mong đợi. Thuật toán STING yêu cầu khả năng tính toán rất thấp nhưng lại có khả năng xử lý tập dữ liệu không gian lớn. Biểu diễn đồ họa của cụm thu được từ cấu trúc phân cấp của các ô lưới và thông tin thống kê liên quan. Một nhược điểm lớn của kỹ thuật này là người dùng được yêu cầu cung cấp tham số mật độ xác định chất lượng phân cụm.

Ngoài STING, thuật toán OPTIGRID (Optimal grid) cũng là một thuật toán phân cụm dựa trên lưới. Trong đó việc phân cụm dữ liệu được thực hiện sao cho tập dữ liệu được phân vùng trong một vùng có mật độ thấp và mặt phẳng cắt phải được phân biệt các cụm càng nhiều càng tốt.

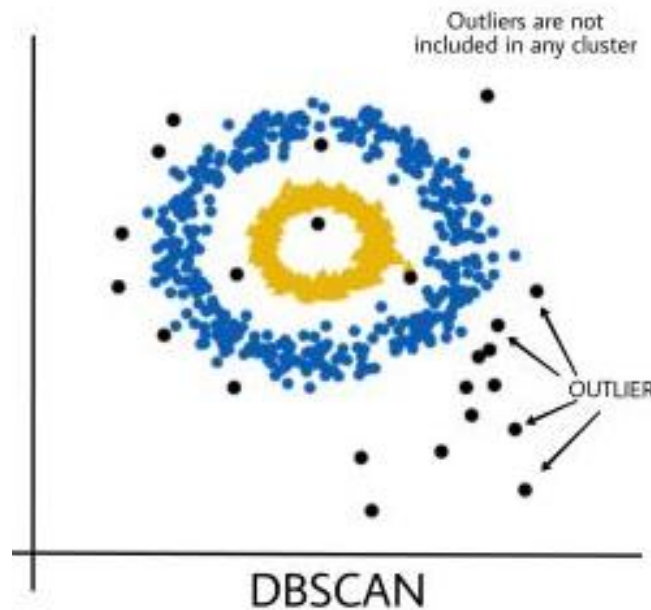
d. Density based Clustering (Phân cụm dựa trên mật độ)

Density based Clustering là phương pháp xác định các nhóm/cụm khác biệt trong dữ liệu dựa trên mức độ “dày đặc” của các điểm dữ liệu, cho phép nó tìm hiểu các cụm có hình dạng tùy ý và xác định các điểm ngoại lai trong dữ liệu.

Thuật toán phân cụm dựa trên mật độ là một thuật toán đóng vai trò quan trọng trong việc khám phá cấu trúc hình dạng phi tuyến tính dựa trên mật độ và

coi các cụm là vùng dày đặc của các đối tượng trong không gian dữ liệu và các cụm được chia theo vùng có mật độ thấp. Giá trị mật độ được liên kết với mỗi đối tượng được đánh giá là số đối tượng lân cận của nó trong một bán kính nhất định. Chất lượng của các kỹ thuật này không bị ảnh hưởng bởi các yếu tố ngoại lai và hình dạng của cụm.

Thuật toán DBSCAN (Density-based Spatial Clustering of Applications with Noise) là thuật toán phổ biến nhất sử dụng phương pháp phân cụm dựa trên mật độ này. Các cụm là các vùng dày đặc trong không gian dữ liệu, được phân tách bởi các vùng có mật độ điểm thấp hơn. Thuật toán DBSCAN được dựa trên khái niệm trực quan của “clusters” (cụm) và “noise” (nhiều). Ý tưởng chính là đối với mỗi điểm của một cụm, vùng lân cận của một bán kính nhất định phải chứa ít nhất một số điểm tối thiểu. Các phương pháp phân vùng (K-means, PAM clustering) và phân cụm phân cấp hoạt động để tìm các cụm hình cầu hoặc cụm lồi. Nói cách khác, chúng chỉ phù hợp với các cụm nhỏ gọn và được phân tách rõ ràng. Hơn nữa, chúng cũng bị ảnh hưởng nghiêm trọng bởi sự hiện diện của “nhiều” và các yếu tố ngoại lai trong dữ liệu. Ưu điểm chính của DBSCAN là không yêu cầu đặc điểm kỹ thuật apriori của số lượng cụm, nó có thể quản lý các ngoại lệ trong khi phân nhóm tập dữ liệu. Thuật toán rất mạnh mẽ trong việc tách các cụm có mật độ cao so với các cụm có mật độ thấp và có thể dễ dàng tìm thấy các cụm và kích thước tùy ý. Tuy nhiên, hiệu suất xử lý tập dữ liệu chiều cao của nó rất kém hoặc yếu.



Hình 6: DBSCAN

e. Soft Clustering

Soft Clustering là một dạng phân cụm trong đó một điểm dữ liệu có thể nằm trong nhiều hơn một cụm.

Fuzzy C-means là một kỹ thuật phân cụm mềm trong đó các điểm dữ liệu thuộc về nhiều hơn một cụm và được phân biệt bằng hàm liên thuộc mờ của nó. Trong phương pháp này, ma trận thành viên của tập dữ liệu đầu vào được giữ nguyên và điều này được cập nhật trong mỗi lần lặp. Phương pháp này có hai ưu điểm chính: Nó có khả năng phân cụm các điểm dữ liệu chồng chéo và tỷ lệ hội tụ của nó rất cao. Tuy nhiên, nhược điểm chính của nó là về tính hợp lệ của cụm do yêu cầu tiên nghiệm của giá trị C cần thiết cho chất lượng của các kết quả phân nhóm.

f. Model based Clustering (phân cụm dựa trên mô hình)

Model based Clustering là dạng phân cụm dữ liệu dựa theo các phương pháp thống kê. Dữ liệu ban đầu được giả định là đã được tạo ra từ một hỗn hợp hữu hạn của các mô hình thành phần. Mỗi mô hình thành phần là một phân phối xác suất, điển hình là một phân phối đa biến tham số.

Self-organizing map (SOM) hay self-organizing feature map (SOFM) là một dạng mạng thần kinh nhân tạo (ANN) được huấn luyện sử dụng học không có giám sát để tạo ra một bản đồ trực quan của tập dữ liệu trong không gian hai

chiều (2D) hoặc ba chiều (3D) và các cụm tương tự được đặt gần nhau do kết quả của mạng nơ-ron một lớp và điều này áp dụng cho các tập dữ liệu lớn

g. Ensemble Clustering

Ensemble Clustering liên quan đến việc áp dụng sự kết hợp của một số phương pháp phân nhóm trên một tập dữ liệu nhất định thành một phân cụm đồng thuận tốt hơn và mạnh mẽ hơn. Sau đó, hàm đồng thuận được sử dụng để tổng hợp các kết quả từ các kỹ thuật phân nhóm khác nhau để tạo ra một kết quả phân nhóm duy nhất. Phương pháp này tránh được hạn chế của việc nhập trước số lượng cụm bằng cách sử dụng các chỉ số xác nhận cụm để lựa chọn số cụm tối ưu cho mỗi tập dữ liệu. Do đó, phân vùng dựa trên đồ thị được áp dụng để thu được kết quả cuối cùng của việc phân cụm, cho phép loại bỏ các cạnh không nhất quán (các giá trị ngoại lai). Loại dữ liệu sử dụng các phương pháp phân nhóm này là dữ liệu Luồng

1.3. Ứng dụng của phân cụm dữ liệu trong thực tế

Kỹ thuật phân cụm có thể áp dụng cho rất nhiều lĩnh vực khác nhau như kinh tế, y tế, khoa học xã hội. Clustering có thể giúp một công ty tăng sự hiệu quả của hoạt động marketing và bán hàng đa mục tiêu, giúp công ty trong lĩnh vực kinh doanh tăng doanh số tăng lợi nhuận, và còn tăng trải nghiệm của khách hàng, thỏa mãn nhu cầu cá nhân. Nó cho phép việc phân nhóm khách hàng, xác định các nhóm khách hàng tiềm năng trong việc truy cập tìm kiếm, mua sắm cùng với thông tin về đặc điểm các khách hàng từ đó tính chỉnh những giải pháp bán hàng cho từng nhóm khách hàng phù hợp.

Ngoài ra có thể kể đến một số ứng dụng như:

- Biology: Phân nhóm động vật và thực vật dựa trên đặc điểm hay thuộc tính của chúng.
- Libraries: Theo dõi đọc giả, sách, dự đoán nhu cầu của người đọc
- Finance: Phân nhóm các đối tượng sử dụng các dịch vụ tài chính, dự đoán xu hướng của khách hàng.
- Phân loại tài liệu, phân loại người dùng web...

CHƯƠNG 2 : THUẬT TOÁN K-MEANS VÀ BÀI TOÁN PHÂN CỤM DỮ LIỆU TỰ ĐỘNG

2.1. Thuật toán phân cụm K-means

2.1.1. Khái quát thuật toán

Ngày nay, việc phát triển vượt bậc của công nghệ thông tin cùng Internet có sự đóng góp không hề nhỏ của AI (Artificial Intelligence) – Trí tuệ nhân tạo và cụ thể ở đây đó là Machine Learning. Machine learning là một phần nhỏ trong AI, nó có khả năng tự học hỏi dựa trên dữ liệu mà không cần lập trình cụ thể. Trong nhiều trường hợp yêu cầu sử dụng kiến thức lớn thì việc tìm kiếm con người sẽ trở nên khó khăn và khan hiếm, hoặc một số việc yêu cầu đưa ra các quyết định nhanh chóng dựa trên lượng dữ liệu khổng lồ và không ổn định thì chắc chắn chỉ con người thôi là chưa đủ. Ngoài ra việc dữ liệu chủ yếu được sinh ra ngày nay là chỉ phù hợp cho máy đọc (Computer readable). Máy học được sinh ra để giải quyết các vấn đề trên. Nghiên cứu các phương pháp, các thuật toán để mô hình hóa bài toán cho phép máy tính tự động hiểu, xử lý và đọc dữ liệu để thực thi nhiệm vụ được giao cũng như cách đánh giá giúp tăng tính hiệu quả.

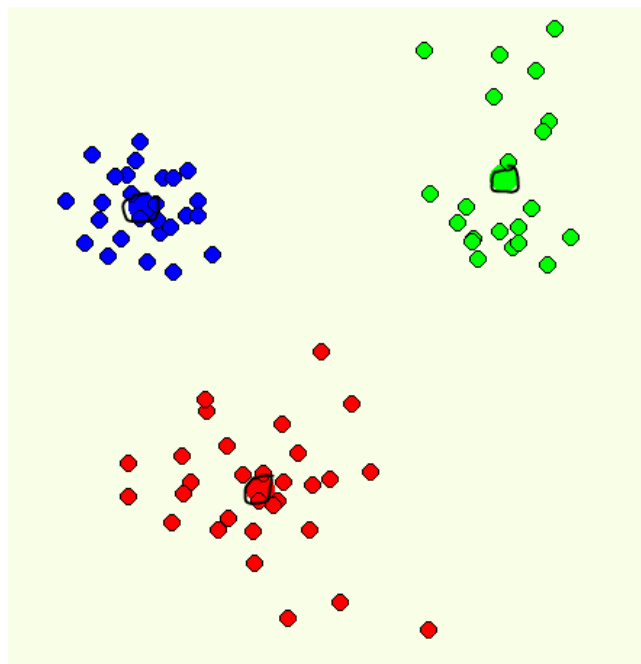
Thuật toán phân cụm K-means là một thuật toán cơ bản nhất để giải quyết bài toán phân cụm. Phương pháp này tạo và cập nhật liên tục các điểm trung tâm với mục đích phân nhóm các điểm dữ liệu cho trước vào các nhóm khác nhau. Đầu tiên tạo các điểm trung tâm ngẫu nhiên và gán các điểm cho trước với điểm trung tâm gần đó nhất. Tiếp theo nó sẽ cập nhật lại các điểm trung tâm của các cụm và thực hiện gán các điểm lại. Sau đó sẽ liên tục lặp lại cho đến khi điểm trung tâm không thay đổi trong vòng lặp tiếp theo. Tuy nhiên thông thường thì với các số liệu lớn và phức tạp nên người ta sẽ dùng thuật toán ở một kết quả gần đúng và có thể chấp nhận được

Thuật toán k-means sử dụng phương pháp tạo và cập nhật trung tâm để phân nhóm các điểm dữ liệu cho trước vào các nhóm khác nhau. Thuật ngữ K-means được James MacQueen sử dụng lần đầu tiên vào năm 1967, mặc dù ý tưởng này

quay trở lại Hugo Steinhaus vào năm 1956. Thuật toán tiêu chuẩn được đề xuất lần đầu tiên bởi Stuart Lloyd của Bell Labs vào năm 1957

Mục đích thuật toán đưa ra ban đầu là phân cụm với dữ liệu ban đầu không có nhãn, và nhiệm vụ của thuật toán là làm thế nào để phân dữ liệu thành các cụm (cluster) khác nhau nhưng các dữ liệu trong cùng một cụm phải có tính chất giống nhau hoặc gần tương đương.

Ví dụ trong thực tế: Khi mà một người đi làm bình thường một ngày nhận được khoảng 20 mail (bao gồm cả email không quan trọng, email rác, quảng cáo) và anh ta sẽ kiểm tra mail đó thường xuyên. Tuy nhiên một hôm anh ta đột nhiên bị bệnh và sau đó phải nghỉ làm khoảng 1 tuần. Thì khi quay lại làm việc số lượng email phải kiểm tra của anh ta là $20 \times 7 = 140$ cái mail. Vậy làm cách nào để người đó có thể phân cụm ra các email khác nhau, email quảng cáo và các email rác ra một nhóm và các email liên quan đến công việc ra một nhóm. Để từ đó xác định các email quan trọng cần kiểm tra. Khi gặp bài toán như trên thì đối với một người biết về Machine Learning thì thuật toán đầu tiên nghĩ ra sẽ là K-means. Vì nó là thuật toán cơ bản nhất mà tất cả các cuốn sách hay các khóa học về Machine Learning đều nhắc tới, và thậm chí là nhắc tới đầu tiên.

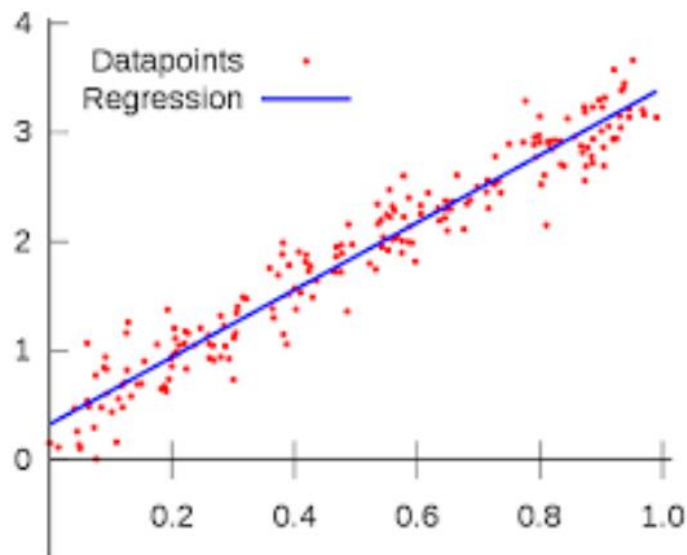


Hình 7: Phân cụm K-means

Ví dụ về các điểm dữ liệu đã được phân cụm trong không gian 2 chiều. Về cơ bản các cụm (cluster) là các điểm gần nhau trong một khoảng không gian tạo nên một tập điểm. Như trên *Hình 7*, ba cụm với các trung điểm (center) được khoanh viền đen và to hơn so với các điểm khác với các nhãn (label) là đỏ, xanh lá và xanh dương.

Thuật toán K-means là một thuật toán lặp lại nhằm phân vùng tập dữ liệu thành các nhóm (cụm) con không trùng lặp được xác định bởi số cụm K cho trước, trong đó mỗi điểm dữ liệu chỉ thuộc về một nhóm. Nó làm cho các điểm dữ liệu trong cụm càng giống nhau càng tốt đồng thời giữ cho các cụm càng khác nhau (càng xa) càng tốt. Thuật toán chỉ định các điểm dữ liệu cho một cụm sao cho tổng khoảng cách bình phương giữa các điểm dữ liệu và trung tâm của cụm (trung bình cộng của tất cả các điểm dữ liệu thuộc về cụm đó) là nhỏ nhất. Khi càng có ít biến thể trong các cụm, thì các điểm dữ liệu trong cùng một cụm càng đồng nhất (tương tự).

Khác với Supervised learning khi mà các thuật toán đề có mục đích là dự đoán đầu ra nó xây dựng một hàm từ những dữ liệu cho trước, những dữ liệu này đã có đầu vào và đầu ra. Hàm trên gọi là hàm huấn luyện, dữ liệu trên được gọi là dữ liệu huấn luyện. Từ hàm trên sẽ dự đoán đầu ra của các dữ liệu mới. Đầu ra có thể là một giá trị liên tục (hồi quy) hay có thể là một nhãn do các đối tượng đầu vào (phân loại), tương đương với 2 dạng của kỹ thuật là Regression và Classification. Ví dụ về thuật toán Facebook gợi ý tag bạn bè trong ảnh. Khi đăng một ảnh Facebook sẽ nhận diện khuôn mặt để dễ dàng cho việc gắn thẻ, đó là một thuật toán Supervised learning. Facebook sử dụng rất nhiều ảnh (hàng ngàn, hoặc thậm chí hàng triệu tỉ....) để huấn luyện (training data), trong dữ liệu đó có đầu vào là các bức ảnh, đầu ra là các bức ảnh với kết quả khoanh các gương mặt người. Thuật toán trên sẽ học và các bức ảnh sau khi đăng lên sẽ được thuật toán dự đoán ra các vùng là khuôn mặt người.



Hình 8: Regression trong Supervised learning

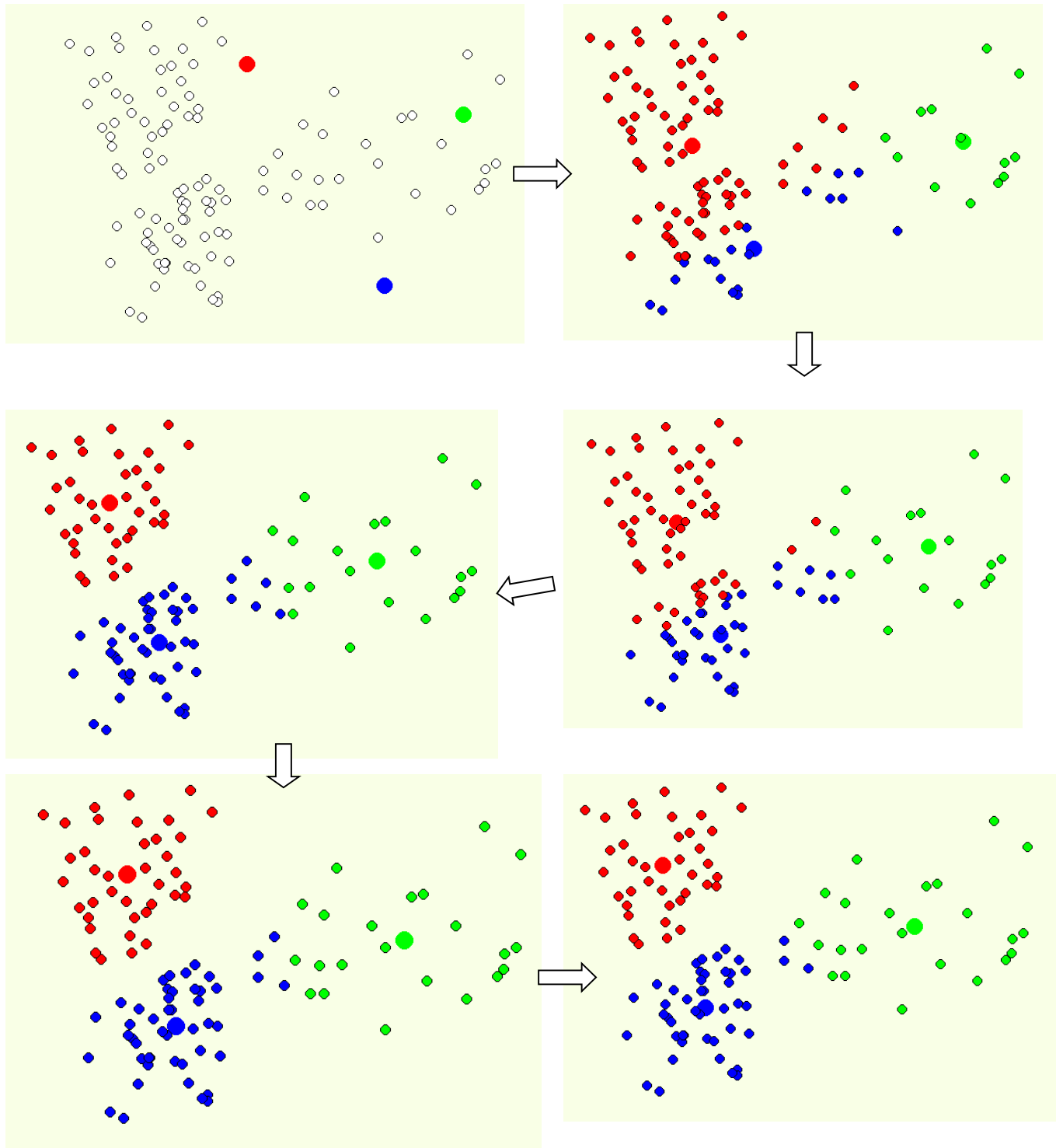
Thuật toán K-means không sử dụng để dự đoán đầu ra cũng giống như các thuật toán Unsupervised learning khác, dữ liệu đầu vào không có kết quả trước. Thay vào đó, một tập dữ liệu ban đầu được thu thập sẽ được thuật toán phân tích cấu trúc để thực hiện một công việc nào đó chẳng hạn như việc phân nhóm dữ liệu hoặc giảm số chiều của dữ liệu. Ví dụ về việc phân nhóm dữ liệu để nén ảnh: Một bức ảnh được vào bởi các điểm ảnh (Pixel) mỗi pixel sẽ có một màu, càng nhiều màu khác nhau trên mỗi pixel thì bức ảnh càng nặng. Nhưng khi nhóm các màu tương đồng với nhau lại và cho nó về cùng một giá trị, độ lớn bức ảnh sẽ nhỏ lại.

2.1.2. Mô tả thuật toán

Dữ liệu đầu vào sẽ là n điểm với số lượng cụm (cluster) là k

- Bước đầu tiên cần chọn ra k điểm bất kỳ làm các center ban đầu và gán các điểm khác với center gần nó nhất. Khi đó ta sẽ có k cụm
- Bước tiếp theo tính trung điểm mới của các cụm bằng việc lấy trung bình cộng của tất cả các điểm trong cụm. Ví dụ như trong không gian 2 chiều, tọa độ trung điểm mới của cụm là (x, y) trong đó x, y lần lượt là trung bình cộng các tọa độ trục tương ứng của các điểm trong cụm.
- Cập nhật tọa độ trung điểm mới và tiếp tục gán các điểm với các trung điểm mới đã cập nhật.

d. Kiểm tra nếu sau bước *c* mà nhãn của các điểm không thay đổi thì dừng thuật toán. Nếu thay đổi thì lặp lại bước *b*



Hình 9: Cách thức thuật toán K-means thực hiện phân cụm

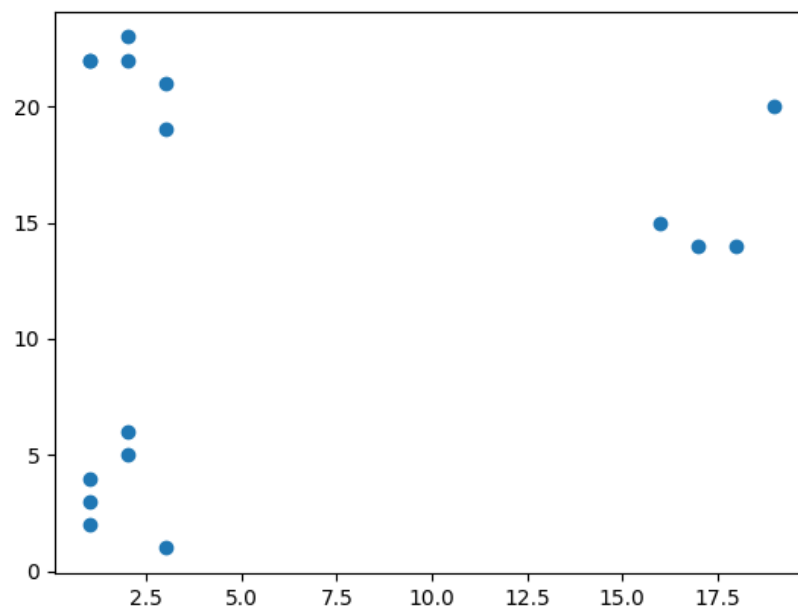
2.1.3. Phương pháp chọn số cụm

Các phương pháp chọn số cụm nhằm cải tiến K-means hầu hết đều có công thức là lặp đi lặp lại thuật toán, với các số K cụm khác nhau nhằm tìm ra số K phù hợp nhất.

a. Phương pháp Khuỷu tay (Thuật toán Elbow)

Vấn đề quan trọng nhất trong các thuật toán phân cụm đó là xác định được số cụm k. Chỉ nói đến việc lựa chọn số cụm đã có thể trở thành một bài toán riêng, bởi không có một số k nào cụ thể để có thể trở thành số cụm cho tất cả các bài toán. Ta có thể tự hình dung dữ liệu bài toán của mình có bao nhiêu cụm và gán cho nó. Tuy nhiên thì điều đó là không thể trong các bài toán phức tạp. Có một cách làm khác đó là thử từng giá trị k (1, 2, 3, 4, ...) và đánh giá kết quả phân cụm theo một logic nào đó. Cụ thể ở đây đối với phương pháp khuỷu tay sẽ sử dụng tổng khoảng cách giữa các điểm với trung điểm trong một cụm. Càng nhiều cụm thì tổng khoảng cách giữa điểm với trung điểm sẽ càng ít. Tuy nhiên sẽ có một số khoảng khi tăng k lên khoảng cách giảm rất nhiều và một số khoảng khi tăng k thì tổng khoảng cách giảm không đáng kể. Và điểm k nối giữa 2 khoảng chính là điểm k phù hợp. Đó là logic của phương pháp này

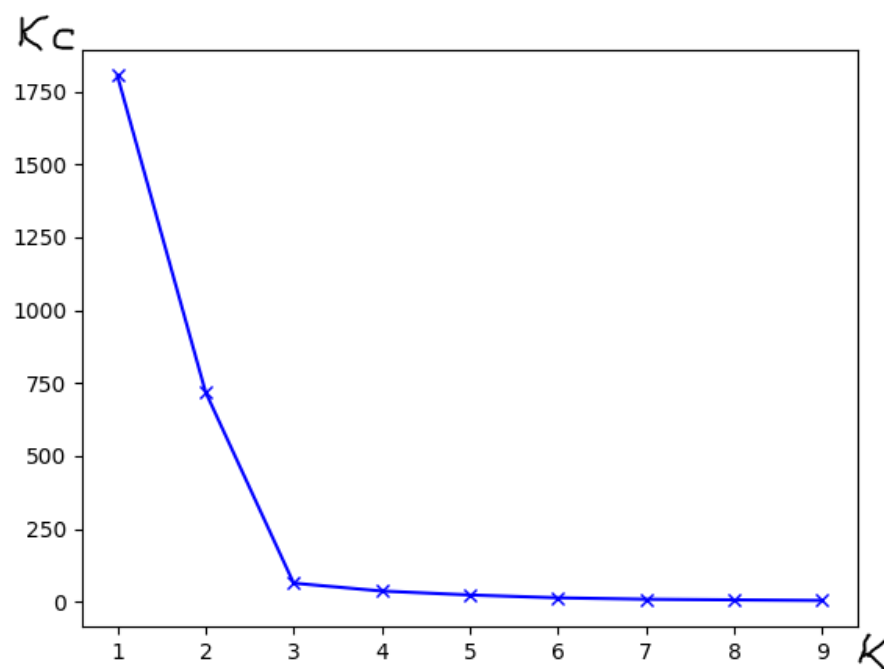
Giả sử có một số điểm cho trước như hình vẽ:



Hình 10: Tọa độ các điểm ví dụ

Nhìn qua hình 11 có thể thấy được là 3 cụm sẽ phù hợp để phân cụm các điểm phía trên.

Biểu đồ thể hiện giữa số k và tổng khoảng cách giữa trung điểm với các điểm trong cụm k tương ứng. Ở đây, khi tăng k lên từ 1 đến 2 và từ 2 đến 3 khoảng cách giảm rất nhiều còn từ 3 lên 4 ... khoảng cách giảm không đáng kể. Từ đó có thể kết luận $k=3$ là số cụm phù hợp để có thể phân cụm các điểm phía trên. Nhìn qua đồ thị có thể thấy nó giống một cái tay, và tại điểm mà $k=3$ nó sẽ là vị trí khuỷu tay _ lý do tại sao người ta gọi đó phương pháp khuỷu tay



Hình 11: Biểu đồ thể hiện số K và tổng khoảng cách các điểm đến trung tâm cụm

Phương pháp Elbow là một phương pháp thực nghiệm để tìm số lượng cụm tối ưu cho một tập dữ liệu. Đây có lẽ là phương pháp nổi tiếng nhất để xác định số lượng cụm tối ưu. Thật không may, không phải lúc nào dữ liệu cũng được phân cụm rõ ràng như vậy, phương pháp Elbow là heuristic và không phải lúc nào cũng có thể hoạt động cho tất cả các tập dữ liệu.

b. Silhouette score:

Để có thể chọn số cụm cho thuật toán K-means. Ngoài phương pháp khuỷu tay thì phân tích silhouette score cũng là một cách hữu ích và thường được xem xét trong các bài toán phân cụm có sử dụng K-means. Tương tự như Elbow,

phân tích Silhouette cũng sử dụng tư duy ý tưởng là lấy nhiều số K và sau đó chọn một số K thích hợp. Tuy nhiên thay vì chọn dựa trên biểu đồ, phương pháp này dựa vào đánh giá Silhouette score dựa vào các giá trị:

- Khoảng cách trung bình giữa quan sát và tất cả các điểm dữ liệu khác trong cùng một cụm. Khoảng cách này cũng có thể được gọi là khoảng cách trung bình trong cụm. Khoảng cách trung bình được ký hiệu bằng **a**
- Khoảng cách trung bình giữa quan sát và tất cả các điểm dữ liệu khác của cụm gần nhất tiếp theo. Khoảng cách này cũng có thể được gọi là khoảng cách cụm gần nhất trung bình. Khoảng cách trung bình được ký hiệu là **b**

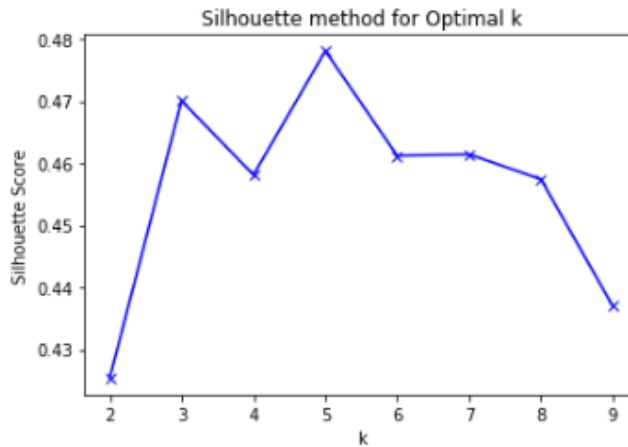
Phân tích Silhouette có thể được sử dụng để nghiên cứu khoảng cách tách biệt giữa các cụm kết quả, nó cho ra một tập hợp các điểm dữ liệu mẫu được sử dụng để đo lường mức độ dày đặc và được phân tách rõ ràng của các cụm. Biểu đồ Silhouette hiển thị thước đo mức độ gần của mỗi điểm trong một cụm với các điểm trong các cụm lân cận và do đó cung cấp một cách để đánh giá các thông số như số lượng cụm một cách trực quan. Số đo này có phạm vi là $[-1, +1]$.

$$\frac{b^i - a^i}{\max(a^i, b^i)}$$

Trong đó:

- a^i là khoảng cách trung bình từ tất cả các điểm dữ liệu trong cùng một cụm
- b^i là khoảng cách trung bình từ tất cả các điểm dữ liệu trong cụm gần nhất

Hệ số Silhouette gần +1 cho biết rằng mẫu ở xa các cụm lân cận. Giá trị 0 cho biết mẫu nằm trên hoặc rất gần ranh giới quyết định giữa hai cụm lân cận và giá trị âm cho biết những mẫu đó có thể đã được gán cho cụm sai.



Hình 12: Biểu đồ phân tích Silhouette score

Với hình thì theo phương pháp có thể thấy, $k=5$ là số cụm phù hợp. Phương pháp phân tích Silhouette có thể tốt hơn Elbow vì nó làm cho quyết định về số lượng cụm tối ưu có ý nghĩa và rõ ràng hơn. Nhưng số liệu này là tính toán tốn kém vì hệ số được tính cho mọi trường hợp. Do đó, quyết định liên quan đến số liệu tối ưu được chọn cho số lượng quyết định cụm phải được thực hiện theo nhu cầu và yêu cầu từ dữ liệu.

2.2. Ứng dụng của thuật toán K-means trong bài toán phân cụm dữ liệu tự động.

2.2.1. K-means và phân cụm dữ liệu tự động

Như đã trình bày ở chương 1, dữ liệu luôn là vấn đề quan trọng và việc phân cụm dữ liệu mang lại rất nhiều lợi ích cho các công ty. Tuy nhiên vấn đề đặt ra đó là càng theo thời gian, dữ liệu càng lớn và phức tạp khiến việc phân nhóm dữ liệu bằng con người dần thành không thể. Chính vì thế sự xuất hiện của máy học mang lại rất nhiều những phương pháp có thể giúp ích con người trong việc phân cụm dữ liệu, và thậm chí là đảm nhiệm luôn công việc này. Bài toán phân cụm dữ liệu tự động là một bài toán lớn, phức tạp thu hút sự chú ý của các nhà nghiên cứu và thuật toán K-means là một trong những phát minh đầu tiên trong việc phân cụm dữ liệu.

K-means là một thuật toán đi đầu và là thuật toán kinh điển trong vấn đề phân cụm dữ liệu. Tuy nhiên là thuật toán đơn giản nhưng K-means yêu cầu

phải chỉ ra số lượng cụm (clusters) và tất nhiên đối với những dữ liệu lớn phức tạp và đa chiều thì việc có thể chọn ra được số cụm bởi mắt thường là không thể. Chính vì thế những cải tiến về sau của các nhà nghiên cứu đã cho việc chọn số cụm trở nên khoa học và “tự động”. Có thể kể đến như phương pháp “khủy tay” (Elbow method) hoặc có thể sử dụng silhouette score – một thông số hay là một thước đo về cách một thuật toán phân cụm đã hoạt động. Giá trị silhouette score sẽ nằm trong khoảng -1 đến 1, càng gần đến 1 có nghĩa là số lượng cụm dùng để phân cụm càng chính xác. Công thức silhouette score = $(b-a)/\max(a,b)$, trong đó, a là khoảng cách trung bình giữa các điểm trong một cụm, b là khoảng cách trung bình giữa tất cả các cụm.

Các thuật toán cải tiến K-means nhằm giúp thuật toán này “tự động” hơn loại bỏ nhược điểm của nó là phải chỉ định số cụm ban đầu. Đa phần đều có một ý tưởng đó là cho số cụm K chạy trong khoảng thích hợp và chọn số K hoàn hảo nhất. Điều này sẽ làm tăng thời gian xử lý và độ phức tạp của thuật toán (không quá đáng kể) tuy nhiên sẽ giảm phụ thuộc vào việc khởi tạo số cụm ban đầu.

2.2.2. Phân cụm văn bản

Ứng dụng của phân cụm dữ liệu mạnh mẽ nhất trong việc phân chia các dữ liệu lớn thành các nhóm nhỏ nhằm tóm tắt, tổng hợp sắp xếp dữ liệu đặc biệt là phân cụm văn bản (Document Clustering)

Các dữ liệu hiện tại có rất nhiều dạng, tuy nhiên K-means chỉ có thể áp dụng với dữ liệu dạng số cụ thể là các vector. Với dữ liệu dạng ảnh, có thể sử dụng các vector để xác định các vị trí điểm ảnh. Tuy nhiên với dữ liệu dạng chữ (Text) trong việc phân cụm văn bản, việc chuyển đổi từ dữ liệu dạng này ra vector cũng sẽ là một bài toán cần quan tâm nếu có sử dụng K-means với mục đích phân cụm dữ liệu dạng Text.

Nhắc tới chuyển đổi dữ liệu văn bản thành vector chắc chắn không thể không nhắc tới thuật toán Tf-idf.

a. Tf-idf

TF-IDF là sự kết hợp của hai từ khác nhau: Term Frequency (**TF**) và Inverse Document Frequency (**IDF**)

TF là tần suất xuất hiện của từ trong văn bản, tuy nhiên các văn bản thường sẽ có độ dài khác nhau nên một số từ sẽ có tần suất xuất hiện lớn hơn trong một văn bản dài so với là một văn bản ngắn. Chính vì thế nên TF thường sẽ được chia cho độ dài văn bản

$$tf(t, d) = \frac{f(t, d)}{\max\{f(w, d): w \in d\}}$$

Trong đó:

- $tf(t, d)$: Tần suất xuất hiện của từ “t” trong văn bản “d”.
Giá trị thuộc khoảng [0, 1]
- $f(t, d)$: là số lần xuất hiện của một từ “t” trong văn bản “d”
- $\max\{f(w, d): w \in d\}$: Số lần xuất hiện nhiều nhất của một từ nào đó trong văn bản “d”

TF được sử dụng để tính toán số lần từ ngữ hiện diện trong một tài liệu. Giả sử, có tài liệu “T1” chứa 5000 từ và từ “Alpha” xuất hiện trong tài liệu đúng 10 lần. Trong thực tế, tổng độ dài của các tài liệu có thể thay đổi từ rất nhỏ đến lớn, vì vậy có khả năng bất kỳ từ nào có thể xảy ra thường xuyên hơn trong các tài liệu lớn so với các tài liệu nhỏ. Vì vậy để khắc phục thì tần suất xuất hiện sẽ được tính bằng số lần xuất hiện chia cho tổng số từ của tài liệu. Trong trường hợp trên, tần suất xuất hiện của từ “Alpha” trong tài liệu trên là: $TF = 10/5000 = 0.002$

IDF là tần suất nghịch đảo văn bản giúp đánh giá tầm quan trọng của một từ. Bởi trong các văn bản thường có các từ không quá quan trọng mà xuất hiện rất nhiều, VÍ DỤ trong tiếng anh: “is”, “I”, “are”, “or”, “and”,... Như vậy cần giảm tầm quan trọng của những từ này xuống

$$idf(t, D) = \log \frac{|D|}{|\{d \in D : t \in d\}|}$$

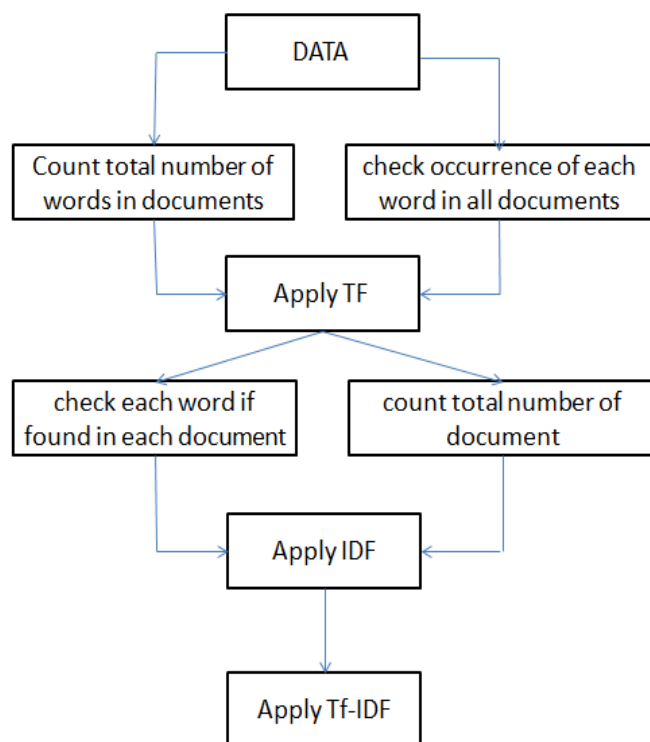
Trong đó:

- $idf(t, D)$: Giá trị của từ trong văn bản
- $|D|$: tổng số văn bản
- $|\{d \in D : t \in d\}|$: Số văn bản chứa từ nhất định

Cơ số logarit trong công thức tính idf không thay đổi giá trị của một từ mà chỉ thu hẹp khoảng giá trị của từ đó. Vì thay đổi cơ số sẽ dẫn đến việc giá trị của các từ thay đổi bởi một số nhất định và tỷ lệ giữa các trọng lượng với nhau sẽ không thay đổi. (nói cách khác, thay đổi cơ số sẽ không ảnh hưởng đến tỷ lệ giữa các giá trị IDF). Tuy nhiên việc thay đổi khoảng giá trị sẽ giúp tỷ lệ giữa IDF và TF tương đồng để dùng cho công thức TF-IDF như bên dưới.

$$tfidf(t, d, D) = tf(t, d) \times idf(t, D)$$

Những từ có giá trị TF-IDF cao là những từ xuất hiện nhiều trong văn bản này, và xuất hiện ít trong các văn bản khác. Việc này giúp lọc ra những từ phổ biến và giữ lại những từ có giá trị cao (từ khoá của văn bản đó).



Hình 13: Các bước trong thuật toán Tf-idf

Có một số hạn chế của thuật toán TF-IDF cần được giải quyết. Hạn chế chính của TF-IDF là, thuật toán không thể xác định các từ ngay cả với một chút thay đổi về thì của nó. Ví dụ: thuật toán sẽ coi “go” và “goes” là 2 từ khác nhau, “play” và “playing” là 2 từ khác nhau, tương tự “years” và “year”, “go” và “went” ... Do hạn chế này, khi áp dụng thuật toán TF-IDF, đôi khi nó cho một số kết quả không như mong muốn. Một hạn chế khác của TF-IDF là, nó không thể kiểm tra ngữ nghĩa của văn bản trong tài liệu và do thực tế, nó chỉ hữu ích cho đến mức từ vựng. Nó cũng không thể kiểm tra sự đồng xuất hiện của các từ. Có nhiều kỹ thuật có thể được sử dụng để cải thiện hiệu suất và độ chính xác. TF-IDF cũng không hiệu quả nếu văn bản cần phân loại không đồng nhất, có thể kết hợp TF-IDF với Naïve Bayes để phân loại thích hợp trong khi xem xét mối quan hệ giữa các lớp

Theo thời gian, các thuật toán mới đang ra đời để giải quyết một số hạn chế của các thuật toán cũ. Ví dụ, stemming process có thể được sử dụng để khắc phục các vấn đề của TF-IDF không thể xác định rằng “play” và “plays” về cơ bản là các từ giống nhau. Stemming process về cơ bản được sử dụng để kết hợp các dạng khác nhau của bất kỳ từ cụ thể nào như “play” và “plays” hoặc “played” thành một biểu diễn duy nhất và chung chung hơn như “play”. Các từ dừng (stop words) có thể được thêm vào càng nhiều càng tốt để các từ không có giá trị như “the” hoặc “a” được lọc và loại bỏ trước khi xử lý dữ liệu. Điều này sẽ đảm bảo ở một mức độ nào đó rằng các từ có ích hay hữu ích trong văn bản sẽ được đưa vào đầu ra.

Thuật toán TF-IDF rất dễ thực hiện và rất mạnh mẽ nhưng không thể bỏ qua những hạn chế của nó. Trong thế giới dữ liệu lớn ngày nay, thế giới đòi hỏi một số kỹ thuật mới để xử lý dữ liệu, trước khi thực hiện phân tích. Nhiều nhà nghiên cứu đã đề xuất một dạng cải tiến của thuật toán TF-IDF được gọi là Adaptive TF-IDF. Thuật toán được đề xuất kết hợp leo đồi để tăng hiệu suất. Một biến thể của TF-IDF cũng đã được quan sát thấy có thể được áp dụng trong nhiều ngôn ngữ bằng cách sử dụng dịch thống kê. Những gã khổng lồ về công cụ tìm kiếm như Google đã điều chỉnh các thuật toán mới nhất như xếp hạng

trang (PageRank) để mang lại kết quả phù hợp nhất khi người dùng đặt một truy vấn. Trong nghiên cứu tương lai, thế giới sẽ chứng kiến một số kỹ thuật mới có thể khắc phục những hạn chế của TF-IDF, để việc truy xuất truy vấn có thể chính xác hơn. TF-IDF có thể được kết hợp với các kỹ thuật khác như Naïve Bayes để có được kết quả tốt hơn nữa.

Ngoài Tf-idf còn có một số các thuật toán khác có thể chuyển đổi dữ liệu dạng Text sang Vector khác như Word2vec hay một phương pháp của Google là BERT (Bidirectional Encoder Representations from Transformers). Các phương pháp này đều tìm ra đại diện của các từ thông qua một bộ dữ liệu rất lớn, nắm bắt nắm bắt ngữ cảnh của một từ trong tài liệu, sự tương đồng về ngữ nghĩa và cú pháp, mối quan hệ với các từ khác

Tóm lại, trong bài toán phân cụm dữ liệu văn bản thì Tf-idf chắc chắn là một thuật toán không thể bỏ qua. Thuật toán này chuyển hóa các văn bản thành các vector dựa trên tần suất xuất hiện của từ trong văn bản (TF) và mức độ quan trọng của từ đó trong văn bản (IDF). Những từ có giá trị TF-IDF cao là những từ xuất hiện nhiều trong văn bản này và xuất hiện ít trong các văn bản khác, nó là các từ quan trọng giúp phân biệt các văn bản. Trong thuật toán, việc tính toán số lần xuất hiện của từ mang tới kết quả tốt nhưng đây cũng chính là hạn chế của thuật toán khi mà thuật toán không thể xác định các từ giống nhau về nghĩa nhưng khác nhau về hình thức.

b. Xử lý dữ liệu với NLTK

Tf-idf là thuật toán mang tới nhiều giá trị và là thuật toán nổi tiếng nhất trong vấn đề phân cụm dữ liệu văn bản (Document Clustering). Tuy nhiên nhược điểm lớn nhất của thuật toán đó là việc không thể nhận thấy sự đồng nghĩa của các từ giống nhau về nghĩa nhưng khác nhau về hình thức. Chẳng hạn như ‘play’, ‘plays’ hay ‘played’, nó là 3 từ có nghĩa giống nhau nhưng đối với thuật toán Tf-idf sẽ coi 3 từ là những từ khác nhau.

Một vấn đề nữa xảy ra khi phân cụm dữ liệu văn bản đó là việc trong văn bản chứa rất nhiều các từ ngữ mà không phải từ nào cũng quan trọng và mang tính đặc trưng cho văn bản. Cùng với đó ngoài các từ thì trong văn bản còn chứa

rất nhiều các ký tự đặc biệt và các chữ số (số chỉ năm, tháng hay số lượng...) cũng không giúp ích cho việc nhận biết đặc trưng của văn bản. Chính vì thế trước khi đưa dữ liệu vào thuật toán Tf-idf trong bài toán phân cụm dữ liệu thì cần phải xử lý dữ liệu trước. Quá trình xử lý dữ liệu sẽ bao gồm các việc: loại bỏ các ký tự không thuộc bảng chữ cái, loại bỏ các từ ngữ có thể xuất hiện nhiều lần trong các văn bản nhưng không mang lại nhiều ý nghĩa (VÍ DỤ: I, you, and, have, that...), đưa các từ về dạng gốc của nó nhằm hạn chế nhược điểm của thuật toán Tf-idf.

The Natural Language Toolkit (NLTK) bộ ngôn ngữ học tự nhiên được phát triển cùng với khóa học ngôn ngữ học tính toán tại Đại học Pennsylvaniain 2001 (Loper and Bird, 2002). Nó được thiết kế với ba ứng dụng: assignments, demonstrations, projects

Assignments: NLTK hỗ trợ các nhiệm vụ có độ khó và phạm vi khác nhau. Trong các bài tập đơn giản nhất, sinh viên thử nghiệm với các công cụ tổng hợp hiện có để thực hiện nhiều nhiệm vụ NLP khác nhau. Khi sinh viên trở nên quen thuộc hơn với bộ công cụ, họ có thể được yêu cầu sửa đổi các thành phần hiện có hoặc tạo hệ thống hoàn chỉnh từ các components hiện có

Demonstrations: Các phép toán đồ họa tương tác của NLTK đã được chứng minh là rất hữu ích cho sinh viên học các khái niệm NLP. Các cuộc trình diễn đưa ra từng bước thực hiện các thuật toán quan trọng, hiển thị trạng thái hiện tại của các cấu trúc dữ liệu quan trọng

Projects: NLTK cung cấp cho sinh viên một framework cho các dự án nâng cao. Các dự án điển hình có thể liên quan đến việc triển khai một thuật toán mới, phát triển một thành phần mới hoặc triển khai một nhiệm vụ mới

Xử Lý Ngôn Ngữ Tự Nhiên có vai trò hết sức quan trọng trong ngành Khoa Học Máy Tính. Nó có vô vàn ứng dụng hữu ích trong cuộc sống cũng như nghiên cứu: Nhận dạng chữ viết, tiếng nói, tổng hợp tiếng nói, dịch tự động, tìm kiếm thông tin, tóm tắt văn bản và cả về khai phá dữ liệu.

NLTK là một thư viện được viết bằng Python. Là một ngôn ngữ hướng đối tượng, Python cho phép dữ liệu và phương thức được đóng gói và sử dụng lại một cách dễ dàng. Python đi kèm với một thư viện tiêu chuẩn mở rộng, bao gồm các công cụ để xử lý số và lập biểu đồ đồ họa. Cú pháp trình tạo mới được bổ sung gần đây giúp dễ dàng tạo các triển khai liên tích cực của các thuật toán (Loper, 2004; Rossum, 2003a; Rossum, 2003b)

NLTK được thực hiện như một tập hợp lớn các mô-đun phụ thuộc lẫn nhau. Một tập hợp các mô-đun cốt lõi bao gồm các kiểu dữ liệu cơ bản được sử dụng trong toàn bộ bộ công cụ. Các mô-đun còn lại là các mô-đun nhiệm vụ, mỗi mô-đun dành cho một nhiệm vụ xử lý ngôn ngữ tự nhiên riêng lẻ. Ví dụ: `nltk.tokenizer` được dành cho nhiệm vụ mã hóa hoặc chia văn bản thành các phân cấu trúc của nó.

Các chức năng, khả năng của NLTK có thể kể đến:

- Tokenization (`nltk.tokenize`): Tokenizers chia chuỗi thành danh sách các chuỗi con. Ví dụ: tokenizers có thể được sử dụng để tìm các từ và dấu câu trong một chuỗi “Good muffins cost \$3.88 in New York. Please buy me two of them.Thanks.” -> ['Good', 'muffins', 'cost', '\$', '3.88', 'in', 'New', 'York', '.', 'Please', 'buy', 'me', 'two', 'of', 'them', '.', 'Thanks', '.']
- Stemming (`nltk.stem`): là quá trình tạo ra các biến thể của một từ gốc. Trong phương pháp này, các từ có cùng nghĩa nhưng có một số biến thể tùy theo ngữ cảnh hoặc câu được chuẩn hóa. Ví dụ: “programing”, “programer” chuyển thành “program”
- Tagging (`nltk.tag.api`): Giao diện gán thẻ mỗi mã thông báo trong một câu với thông tin bổ sung, chẳng hạn như phân lời nói của nó. Giao diện xử lý để gán thẻ cho mỗi mã thông báo trong danh sách. Thẻ là các chuỗi phân biệt chữ hoa chữ thường xác định một số thuộc tính của mỗi mã thông báo, chẳng hạn như phân lời nói hoặc ý nghĩa của nó.

- Parsing (nltk.parse.api): Một lớp xử lý để lấy ra các cây đại diện cho các cấu trúc có thể có cho một chuỗi các mã thông báo. Những cấu trúc cây này được gọi là " parses ". Thông thường, bộ phân tích cú pháp được sử dụng để lấy cây cú pháp cho các câu. Nhưng các trình phân tích cú pháp cũng có thể được sử dụng để lấy ra các loại cấu trúc cây khác, chẳng hạn như cây hình thái và cấu trúc diễn ngôn.
- stopwords list (nltk.corpus): Danh sách các từ xuất hiện nhiều trong các văn bản tuy nhiên không mang nhiều giá trị có thể đánh giá nội dung văn bản

Mỗi thuật toán xử lý ngôn ngữ được thực hiện như một lớp. Ví dụ: ChartParser và RecursiveDescentParser phân loại mỗi thuật toán đơn lẻ để phân tích cú pháp một văn bản. Các thuật toán xử lý ngôn ngữ được triển khai bằng cách sử dụng các lớp thay cho các hàm vì ba lý do. Đầu tiên, các tùy chọn algorithm-specific có thể được chuyển cho trình cấu trúc, cho phép một giao diện nhất quán để áp dụng các thuật toán. Thứ hai, một số thuật toán cần phải khởi tạo trạng thái của chúng trước khi chúng có thể được sử dụng. Ví dụ: lớp NthOrderTagger phải được khởi tạo bằng cách đào tạo trên một kho ngữ liệu được gán thẻ trước khi nó có thể được sử dụng. Thứ ba, phân lớp con có thể được sử dụng để tạo ra các phiên bản chuyên biệt của một thuật toán nhất định. Mỗi mô-đun xử lý tạo ra một giao diện cho nhiệm vụ của nó. Các lớp giao diện được phân biệt bằng cách đặt tên chúng bằng chữ hoa ở sau "I", chẳng hạn như "ParserI". Mỗi giao diện là một phương thức hành động duy nhất thực hiện nhiệm vụ được thực hiện bởi giao diện. Ví dụ, giao diện ParserI xác định phương thức phân tích cú pháp và giao diện Tokenizer xác định phương thức mã hóa. Khi thích hợp, một giao diện sẽ xác định các phương thức hành động mở rộng, cung cấp các biến thể về phương thức phản ứng cơ bản.

NLTK là một nền tảng hàng đầu để xây dựng các chương trình Python để làm việc với dữ liệu ngôn ngữ của con người. Nó cung cấp các giao diện để sử dụng cho hơn 50 tài nguyên ngữ liệu và từ vựng như classification, tokenization, stemming, tagging, parsing,... NLTK đã được gọi là "một công cụ tuyệt vời để

giảng dạy và làm việc trong ngôn ngữ học tính toán sử dụng Python” và “một thư viện tuyệt vời để chơi với ngôn ngữ tự nhiên.”

2.2.3. Một số ứng dụng khác

Khai phá dữ liệu mang đến một cơ hội lớn cho sự phát triển của các thuật toán trong học máy, trong đó có K-means. Cụ thể, khi các dữ liệu ngày càng lớn, nhu cầu đòi hỏi việc sắp xếp và phân cụm dữ liệu ngày càng tăng, K-means là thuật toán đi đầu trong việc phân cụm dữ liệu và có nhiều ứng dụng trong các bài toán thực tế. Trong lĩnh vực Marketing, K-means có thể giúp xác định các nhóm khách hàng, các khách hàng tiềm năng và khách hàng giá trị, phân loại và dự đoán hành vi của khách hàng. Trong các lĩnh vực khác như phân loại người dùng web, phân loại tài liệu, theo dõi độc giả cũng như phân tích dự đoán nhu cầu đọc giả...

Ngoài việc áp dụng đối với dữ liệu, K-means còn có ứng dụng trong việc nén ảnh. Một bức ảnh được tạo bởi các Pixels hay có thể gọi là các điểm ảnh. Trong một hình ảnh có màu, mỗi pixels có 3 bytes chứa các giá trị RGB (Red-Green-Blue) tương ứng với 3 màu đỏ, xanh lá và xanh lam cho mỗi pixel. Phân cụm K-means trong việc nén ảnh này sẽ nhóm các màu tương tự lại với nhau thành các cụm màu khác nhau. Do đó, mỗi trung tâm cụm là đại diện của vector màu tương ứng, các trung tâm cụm này sẽ thay thế tất cả các vector màu trong cụm tương ứng, hay có thể nói là màu các điểm ảnh trong một cụm sẽ được thay bằng màu của cụm. Càng ít cụm bức ảnh càng ít màu kích thước bức ảnh sẽ càng nhỏ lại tuy nhiên chất lượng ảnh sẽ giảm xuống.



Hình 14: Ứng dụng K-means trong việc nén ảnh

CHƯƠNG 3: ỨNG DỤNG THUẬT OÁN K-MEANS CHO BÀI TOÁN PHÂN CỤM LỊCH SỬ TÌM KIẾM CÁ NHÂN

3.1 Mô tả bài toán

3.1.1. Ứng dụng của bài toán:

Khi cuộc sống của con người ngày càng gắn liền với Internet; học tập, làm việc cho đến giải trí. Người dùng truy cập web, tìm kiếm những thông tin họ yêu thích hoặc với mục đích học tập nghiên cứu và trình duyệt sẽ lưu trữ lịch sử truy cập đó nhằm giúp người dùng có thể tìm kiếm lại những trang web họ đã truy cập. Con người càng sử dụng Internet nhiều, lịch sử web của họ cũng sẽ chứa nhiều thông tin về sở thích và công việc. Dựa vào lịch sử web cá nhân hoàn toàn có thể đoán ra và đánh giá các thông tin về người dùng đó.

Trong cuộc sống hiện đại thì khi truy cập web, phần lớn người dùng sẽ chủ yếu dùng 2 chức năng: mạng xã hội và tìm kiếm Google. Việc mạng xã hội phát triển là điều mà ai cũng đã biết bởi bản thân họ hầu như ai cũng có cho mình ít nhất một tài khoản mạng xã hội. Ngoài mạng xã hội, người dùng thường sẽ truy cập Google nhằm tìm kiếm thông tin về sở thích hay về công việc của họ. Điều này hoàn toàn có thể giải thích được bởi lẽ Google hiện đang là công ty đi đầu trong công nghệ tìm kiếm mà chắc chắn đang không có đối thủ. Thông qua các từ khóa mà một người tìm kiếm có thể suy ra được sở thích và công việc của người đó.

Khi điều tra một cá nhân, việc tìm kiếm các thông tin về người đó gặp rất nhiều khó khăn. Một ứng dụng nhằm đưa ra một cái nhìn bao quát về tính cách, sở thích, công việc của một người dựa trên lịch sử truy cập web của họ có thể là bước đi đầu tiên trong việc nắm bắt thông tin về người đó.

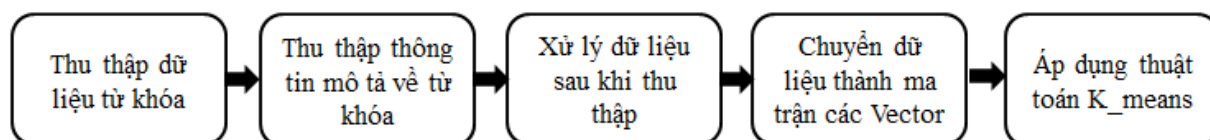
Ngoài ra, hiện nay có nhiều bạn trẻ đang hoang mang không biết họ yêu thích hay đam mê về vấn đề gì. Thông qua ứng dụng có thể giúp người dùng tự đánh giá bản thân thông qua lịch sử tìm kiếm Google của chính họ.

3.1.2. Mô tả bài toán

Công việc đầu tiên đó là thu thập dữ liệu về từ khóa, tuy nhiên việc phân cụm các từ khóa là rất khó khăn do các từ khóa có độ dài khác nhau nhưng đôi khi lại mang ý nghĩa về một lĩnh vực giống nhau. Ví DỤ: từ khóa “Python” và “Hướng dẫn cài đặt và sử dụng ngôn ngữ lập trình Python”. Qua 2 từ khóa có thể thấy người dùng trên quan tâm đến “Python”, tuy nhiên từ khóa tìm kiếm lại khác nhau. Vấn đề trên có thể giải quyết bằng việc tìm kiếm từ khóa trên Google một lần nữa và qua kết quả tìm kiếm có thể thu được dữ liệu dạng văn bản chứa các từ ngữ có liên quan đến vấn đề được tìm kiếm và cần phân cụm.

Dữ liệu mô tả thu thập được sẽ được xử lý nhằm loại bỏ các ký tự đặc biệt, số và các từ không có giá trị. Đồng thời cần đưa các từ về dạng gốc của nó, Ví DỤ như “go” và “goes” có cùng nghĩa nhưng khác nhau về hình thức. Điều này có thể ảnh hưởng đến thuật toán Tf-idf.

Sau khi xử lý dữ liệu, áp dụng thuật toán Tf-idf chuyển đổi dữ liệu các văn bản thành ma trận các Vector. Đưa ma trận thu được vào thuật toán K_means, áp dụng phương pháp khuỷu tay tìm ra số cụm và phân cụm. Quá trình xử lý được mô tả như hình dưới.



Hình 15. Sơ đồ các bước thực hiện bài toán phân cụm dữ liệu lịch sử tìm kiếm

3.2 Thu thập dữ liệu lịch sử tìm kiếm cá nhân

3.2.1. Lịch sử truy cập web cá nhân

Lịch sử duyệt web là danh sách các trang web mà người dùng đã truy cập, cũng như dữ liệu liên quan như tiêu đề và thời gian truy cập trang web đó. Nó thường được lưu trữ cục bộ bởi các trình duyệt web (Chrome, Firefox, Safari...) nhằm cung cấp cho người dùng danh sách lịch sử để quay lại các trang web đã truy cập trước đó. Lịch sử duyệt web có thể phản ánh sở thích, nhu cầu và thói quen của người sử dụng web

Khoảng thời gian lưu trữ trên các trình duyệt web là khác nhau. Ví dụ: Firefox ghi lại vô thời hạn lịch sử web bên trong một tệp có tên “places.sqlite” nhưng sẽ xóa lịch sử khi dung lượng ổ đĩa cạn (xóa lịch sử truy cập lâu nhất đến gần nhất). Trong khi đó Chrome lưu trữ mặc định là 90 ngày nhưng nếu đồng bộ với tài khoản Google thì thời gian lưu trữ sẽ lâu hơn và lưu trữ cả lịch sử web trên các thiết bị khác có cùng tài khoản Google. Một tệp lịch sử lưu trữ vô thời hạn có tên “Archived History” đã từng được ghi lại và đã bị xóa và tự động xóa trong phiên bản 37 phát hành vào tháng 9 năm 2014

Trình duyệt web lưu trữ trang web, ảnh và các file khác khi người dùng truy cập vào một trang web. Từ đó giúp nó tránh việc phải tải lại trang, giảm thiểu thời gian truy cập. Việc này cũng có lợi cho máy chủ web khi được giảm tải bằng cách không cần gửi lại dữ liệu mà trình duyệt đã có trên máy.

Như đã đề cập bên trên lịch sử web cá nhân có thể phản ánh sở thích, nhu cầu và thói quen của người sử dụng web. Và chắc chắn phần lớn mọi người đều không thích việc người khác biết lịch sử truy cập web của mình. Tuy nhiên lịch sử web không thực sự riêng tư một chút nào. Nó có thể được theo dõi bởi các trang web, trình duyệt, ISP và thậm chí là chính phủ.

Tuy nhiên, hầu hết các trình duyệt phổ biến đều lưu giữ nhật ký lịch sử duyệt web trong cơ sở dữ liệu SQLite. Thông tin trong cơ sở dữ liệu này bao gồm URL đã được truy cập, tiêu đề của trang, thời gian trang được truy cập và số lần trang được truy cập. Dữ liệu này có thể được thu thập và xử lý, trong khi cơ sở dữ liệu chứa lịch sử duyệt web nằm trong thư mục data của người dùng và đường dẫn sẽ tùy thuộc vào hệ điều hành tuy nhiên khi đề mặc định, đường dẫn là giống nhau với mỗi loại trình duyệt. Cùng với đó, nhận thức về an toàn dữ liệu cá nhân của người dùng Internet hiện nay là chưa cao, họ chưa có các kiến thức cơ bản nhằm giúp phòng ngừa các cuộc tấn công mã độc hoặc các đường link tấn công phishing. Điều này giúp các kẻ tấn công có cơ hội để lấy được lịch sử truy cập cá nhân của người sử dụng, sau đó đe dọa hoặc sử dụng với mục đích khác.

3.2.2. Ứng dụng thu thập lịch sử tìm kiếm cá nhân

Trong tất cả các ứng dụng, trang web thì Google đang khẳng định mình là một ông lớn đi đầu trong công nghệ tìm kiếm. Giao diện đơn giản dễ sử dụng và khả năng tìm kiếm nhanh với tài nguyên phong phú. Không khó để nói Google quá phổ biến trên mạng Internet hiện nay với lượng người sử dụng truy cập tìm kiếm là nhiều đáng kể.

Con người sử dụng Google để tìm kiếm tài liệu học tập, nghiên cứu để tìm kiếm những thứ họ yêu thích, đam mê. Điều này rất có lợi cho Google để có thể tiếp cận các nhà quảng cáo, bởi khi có ai đó tìm kiếm một cái gì đó thì chắc hẳn họ đang có quan tâm đến từ khóa đó. Chỉ cần dựa vào toàn bộ các từ khóa mà một người tìm kiếm thì chắc chắn sẽ nắm được rất nhiều thông tin về người này như nghề nghiệp, sở thích...

Vậy câu hỏi đặt ra là có phương pháp nào có thể lấy được thông tin về lịch sử tìm kiếm cá nhân hay không? Google có tính năng cho phép người dùng có thể truy cập vào kiểm tra hoạt động trên Google. Tính năng này đòi hỏi người sử dụng cần đăng nhập và hoạt động trên Google thì có thể kiểm tra lịch sử xem của rất nhiều video trên Youtube, hoặc truy cập Gmail, Google Meet...

Tuy nhiên nếu người dùng có lưu lịch sử duyệt web cá nhân cục bộ thì việc trích xuất lịch sử tìm kiếm Google là không khó. Google tìm kiếm sử dụng phương thức GET để gửi các Parameters đến máy chủ và trong các Parameters đó có từ khóa tìm kiếm. Một url tìm kiếm Google thông thường sẽ có dạng:

<https://www.google.com.vn/search?q=tu+khoa+gi+do&...>

Ở đây có thể thấy “<https://www.google.com.vn/>” là tên miền của Google, “search?” gọi đến chức năng tìm kiếm, “q=” là tên payload từ khóa, “tu+khoa+gi+do” là từ khóa, tiếp theo dấu & có thể là một số parameters khác như “lang_en”: tìm kiếm kết quả tiếng anh, “lang_vi” với tiếng việt ...

Với quy tắc đó hoàn toàn có thể tạo một bộ lọc giúp lọc ra các từ khóa tìm kiếm từ lịch sử web cá nhân. Cùng với thư viện browserhistory trong Python việc xây dựng một con mã độc nhằm trích xuất dữ liệu tìm kiếm cá nhân là một điều dễ dàng.

```

history=bh.get_browserhistory()
for browser_name,history_list in history.items():
    for tup in history_list:
        if ('google' in tup[0]) and ('search' in tup[0]):

            x=re.search("q=.*&r",tup[0])
            if x:
                x=x.group()[2:-2]
                # print(x)
                x2=re.search(".*&t",x)
                x3=re.search(".*&s",x)

                if x2:
                    x=x2.group()[:-2]
                elif x3:
                    x=x3.group()[:-2]
                if len(x)<50:
                    ls.append(x)
            xx=re.search("q=.*&oq",tup[0])
            if xx:

                xx=xx.group()[2:-3]
                # print (xx)
                x2=re.search(".*&t",xx)
                x3=re.search(".*&r",xx)
                if x2:
                    xx=x2.group()[:-2]
                elif x3:
                    xx=x3.group()[:-2]
                if len(xx)<50:
                    ls.append(xx)

```

Hình 16. Phương pháp lọc, trích xuất từ khóa từ url tìm kiếm Google

3.2.3. Thu thập thông tin về từ khóa

Dữ liệu lịch sử tìm kiếm cá nhân đã thu thập được nó là những từ khóa và từ khóa thì rất đa dạng về cách người sử dụng tìm kiếm. Nó có thể là một câu hoặc đơn giản chỉ là một tên riêng, dẫn đến việc có thể tìm ra được các điểm chung của các dữ liệu này là điều rất khó. Cùng với đó là thuật toán K-means chỉ có thể áp dụng với dữ liệu dạng số. Vậy nên bài toán tiếp theo được đưa ra đó là tìm cách chuyển các dữ liệu từ khóa tìm kiếm đã thu thập được thành dữ liệu dạng số. Việc các từ khóa đa dạng khiến việc tìm điểm chung để chuyển đổi là rất khó, biện pháp tối ưu hiện tại có thể đưa ra đó là thu thập thêm thông tin về các từ khóa, cụ thể đó là thêm phần mô tả cho các từ khóa rồi tiến hành phân cụm dữ liệu văn bản (Document Clustering) với thuật toán kinh điển trong vấn đề này là Tf-idf.

Trước tiên cần phải có dữ liệu mô tả các từ khóa. Phương pháp đơn giản nhất là sử dụng Google tìm kiếm lại các từ khóa đó và sau đó thu thập dữ liệu tìm kiếm được. Python có rất nhiều các thư viện hữu dụng cho công việc này, BeautifulSoup là thư viện điển hình. Tuy nhiên vấn đề nữa đặt ra ở đây đó là việc các ngôn ngữ khi tìm kiếm khác nhau, điều này ảnh hưởng tới thuật toán khi mà Tf-idf dựa trên tần số xuất hiện của các từ. Chính vì thế cần đưa các từ khóa qua một chương trình dịch, và khi tìm kiếm Google cần phải cố định ngôn ngữ (Nên đặt lang_en, ngôn ngữ phổ biến hơn cả).

Đưa dữ liệu các từ khóa vào vòng lặp để thu thập từng dữ liệu một. Công việc có vòng lặp là đưa các từ khóa lần lượt vào url cho trước như hình bên dưới.

```
url = 'https://www.google.com/search?q='  
  
for i in t:  
    u=url  
    i=i.replace(' ','+')  
    u=u+i+'+wiki&lr=lang_en'
```



Hình 17: Phương pháp tạo list địa chỉ tìm kiếm Google mới

t ở đây là danh sách các từ khóa cần tìm kiếm, giả sử từ khóa là “day la tu khoa” thì sau khi qua vòng lặp ta sẽ có một url mới có dạng:

https://www.google.com/search?q=day+la+tu+khoa+wiki&lr=lang_en

Từ khóa đã được bỏ khoảng trắng và thay thế bằng dấu “+” cùng với đó được gán thêm từ “wiki” để khi tìm kiếm sẽ ưu tiên đến trang wikipedia bởi để thu thập dữ liệu dạng text thì wikipedia sẽ cho khá nhiều chữ, điều mà việc phân cụm văn bản sẽ trở nên chính xác hơn.

Vòng lặp sẽ cho ra một dãy các địa chỉ url Google search. Công việc tiếp theo chỉ cần sử dụng thư viện “requests” với phương thức GET và quét từng url trong dãy url trên cùng với việc sử dụng BeautifulSoup để lấy các dữ liệu là sẽ có một danh sách các mô tả tương ứng với từng từ khóa.

 des.txt	6/9/2021 9...	Text Document	5,754 KB
 key.txt	4/14/2021 ...	Text Document	9 KB

Hình 18. File des.txt chứa mô tả của hơn 400 từ khóa trong file key.txt

3.3. Xử lý dữ liệu văn bản với thuật toán Tf-idf và bộ thư viện NLTK

3.3.1. Khái quát về thuật toán Tf-idf

TF-IDF là thuật toán kinh điển và phổ biến nhất cho việc chuyển hóa dữ liệu dạng text nhằm mục đích phân cụm văn bản (Document Clustering). Là cầu nối giữa dữ liệu với K-means, bản thân thuật toán đã có sự phân cụm bên trong. TF-IDF là viết tắt của Term Frequency–Inverse Document Frequency, có nghĩa là tần suất xuất hiện của từ (TF) với tần suất nghịch đảo văn bản hay mức độ quan trọng của một từ trong văn bản (IDF). Thuật toán là một thống kê số học thể hiện tầm quan trọng của một từ trong văn bản, thường được sử dụng trong mảng truy xuất thông tin và khai phá dữ liệu dạng văn bản (Text mining).

Trong các văn bản luôn có một tập các từ xuất hiện và được sử dụng nhiều hơn so với các từ khác. Vì vậy việc thuật toán thường được ứng dụng trong các bài toán tìm kiếm, nhằm giúp hệ thống biết được từ khóa nào là từ khóa quan trọng mà người sử dụng ứng dụng quan tâm .

3.3.2. Xử lý dữ liệu với NLTK (Natural Language Toolkit):

Một vấn đề nữa xảy ra khi áp dụng thuật toán đó là các từ ngữ có nghĩa giống nhau nhưng hình thức lại khác nhau. Chẳng hạn như trong tiếng anh “go” và “goes” có cùng nghĩa là “đi” tuy nhiên hình thức thì 2 từ đó khác nhau, điều này có ảnh hưởng đến thuật toán. Hoặc chữ in hoa hay in thường cũng gặp phải trường hợp tương tự. Tuy nhiên vấn đề có thể giải quyết đơn giản với thư viện xử lý ngôn ngữ tự nhiên “nltk”. Trong đó có hàm xử lý “SnowballStemmer” giúp chuyển đổi các từ thành dạng nguyên thủy của nó và có một danh sách các “stopwords” giúp việc loại bỏ các từ ngữ không cần thiết trở nên đơn giản. Trong quá trình loại bỏ các từ không cần thiết và chuyển đổi từ nên loại bỏ luôn các ký hiệu không có ý nghĩa như số và các ký tự đặc biệt

Như đã trình bày về hạn chế của thuật toán, thuật toán không thể xác định các từ ngay cả với một chút thay đổi về thì của nó. Chính vì vậy để tăng độ chính xác cho thuật toán, cần phải xử lý dữ liệu đầu vào. Thư viện NLTK có sẵn trên Python có chức năng stemming

```

stemmer = SnowballStemmer("english")

def tokenize_and_stem(text):
    tokens = [word for sent in nltk.sent_tokenize(text) for word in nltk.word_tokenize(sent)]
    filtered_tokens = []
    for token in tokens:
        if re.search('[a-zA-Z]', token):
            filtered_tokens.append(token)
    stems = [stemmer.stem(t) for t in filtered_tokens]
    return filtered_tokens
    return stems

```

Hình 19: Phương pháp xử lý dữ liệu với thư viện nltk

Việc đầu tiên là sử dụng chức năng tokenize trong nltk để tách văn bản thành các từ. Với sự trợ giúp của phương thức nltk.tokenize.word_tokenize() nó trả về các âm tiết từ một từ duy nhất, một từ đơn có thể chứa một hoặc hai âm tiết. Sau đó loại bỏ các từ không phải là chữ. Và áp dụng SnowballStemmer để chuyển các từ về từ gốc, giảm sự phức tạp của các từ khi các từ có nghĩa giống nhau.

3.3.3. Áp dụng thuật toán Tf-idf

Sau khi đã có dữ liệu là các từ khóa cùng với mô tả của các từ khóa. Việc tiếp theo là lập trình và đưa dữ liệu vào thuật toán. Trong thư viện sklearn của Python có luôn một hàm tên TfidfVectorizer giúp việc tính toán hoàn toàn có sẵn trong Python. Các tham số thuật toán mặc định giúp việc tính toán chuẩn xác hơn, một số tham số cần điều chỉnh tùy thuộc vào loại văn bản.

- ngram_range: cụm từ, Ví DỤ: ‘Bảng’ – ngram_range = 1, ‘Bảng chữ cái’ – ngram_range = 3
- min_df: Đưa ra một giá trị mà khi một từ có tần suất xuất hiện thấp hơn mức đó sẽ bị bỏ qua
- max_df: Đưa ra một giá trị mà khi một từ có tần suất xuất hiện cao hơn mức đó sẽ bị bỏ qua

Khi đã có dữ liệu được lọc, cần thiết lập các thông số cần thiết cho TfidfVectorizer. Thông thường nên để mặc định, tuy nhiên tùy loại tài liệu mà các thông số sẽ ảnh hưởng ít nhiều đến quá trình phân cụm về sau.

```

tfidf_vectorizer = TfidfVectorizer(max_df=0., max_features=200000,
                                   min_df=0.2, stop_words='english', ngram_range=(1,3))
tfidf_matrix = tfidf_vectorizer.fit_transform(synopses)

```

Hình 20: Hàm TfidfVectotizer

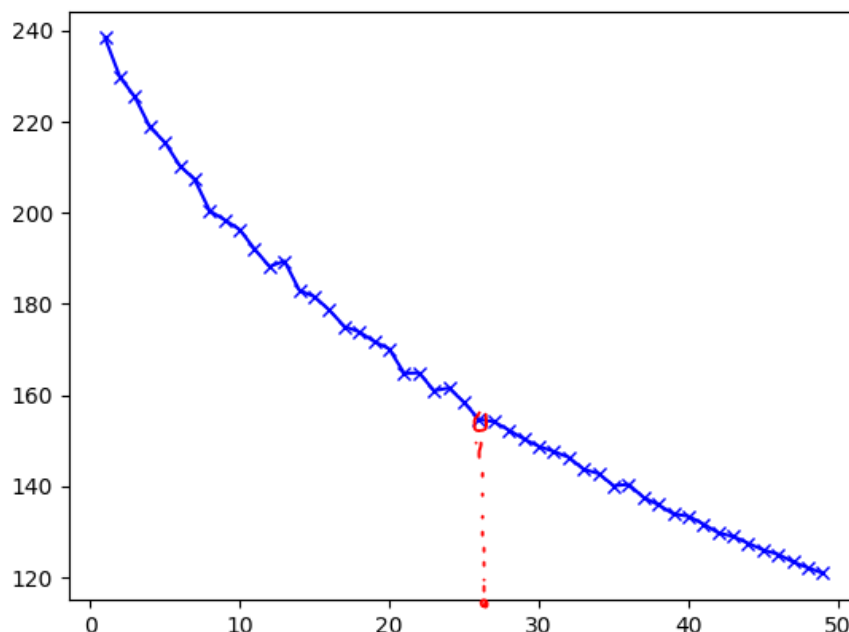
3.4. Áp dụng thuật toán K-means

Thư viện scikit learn của Python đã có sẵn luôn một hàm K-means. Công việc tiếp theo cần làm như bao bài toán có K-means khác đó là chọn số cụm K. Để làm được việc này thì phương pháp đơn giản nhất có thể sử dụng là “elbow” phương pháp khuỷu tay. Phương pháp này đơn giản tuy nhiên sẽ mất thêm thời gian do việc phải tạo một vòng lặp thay các cụm k khác nhau để có thể cho ra một biểu đồ tương quan giữa số cụm và tổng khoảng cách các điểm với cụm.

```
sumdn= []
K= range(1,50)
for k in K:
    km= KMeans(n_clusters=k)
    km.fit(tfidf_matrix)
    sumdn.append(km.inertia_)

plt.plot(K, sumdn, 'bx-')
plt.show()
```

Hình 21: Sử dụng phương pháp Elbow chọn K



Hình 22: Biểu đồ thể hiện số K và tổng khoảng cách giữa các điểm tới trung điểm trong cụm

Sau khi đã có số cụm thì các tham số cho hàm K-means đã đủ để có thể tiến hành phân cụm dữ liệu từ ma trận đã nhận: `tfidf_matrix`

Để có thể in kết quả ra, phương pháp hiệu quả nhất là sử dụng thư viện `pandas`. Tạo một dataframe chứa 2 cột là “cluster” và ”từ khóa tìm kiếm”

```
Cluster 9: casio 570, One Piece Anime, one piece english, serie a, aspire a315 - 54, thinkpad 1540,  
Cluster 10: vpn works, vpn, client vpn, vpn vpn, see the atm card release date, vtv24, Kevin Mitnick, OSCP,  
Cluster 11: adblock, remove ads youtube extension, windows, rom stands for, matplotlib, numpy package, numpy, pip install numpy, ereg php, psd online viewer, numpy replace values, numpy array replace, numpy array insert,  
Cluster 12: python 2. 7, python, python processing function in python, python file, python python processing function, no class named ipython, process python file, python 3 86, python 3 vs python 2, python online, get search history google python, Two years experience python interview questions,  
Cluster 13: wandavision episode 8, wandavision episode 7, wandavision episode 7, wandavision episode 6 leak, WANDA VISION, WANDA AND VISION, wandavision,
```

Hình 23. Kết quả sau khi phân cụm

Dựa vào kết quả có thể đánh giá người sử dụng của lịch sử dữ liệu này có sở thích với “wandavision” (một bộ phim), “One piece” (Truyện tranh), “VPN” (ứng dụng) hay “python” (Ngôn ngữ lập trình). Thay vì phải đọc và tìm hiểu hơn 400 từ khóa thì chỉ cần lướt qua kết quả phân cụm 26 cụm.

KẾT LUẬN

Trong quá trình phát triển vượt bậc hiện nay, các công ty sẽ cần được tiếp cận và cập nhật các phương pháp phân tích xử lý dữ liệu. Bởi lẽ dữ liệu ngày nay quá lớn và phức tạp, nó sẽ vô giá trị nếu không có phương pháp phân tích phù hợp và sẽ mang tới rất nhiều lợi ích nếu được khai phá. Phân cụm dữ liệu là một trong những hướng nghiên cứu trọng tâm của khai phá dữ liệu có ứng dụng quan trọng trong việc phân tích dữ liệu, đặc biệt là những dữ liệu lớn và phức tạp. Việc phân cụm dữ liệu thành các nhóm, cụm riêng biệt giúp cho việc phân tích, đánh giá dữ liệu đồng thời mang tới một cái nhìn bao quát tới dữ liệu cần phân tích. Đề tài này đề cập tới ứng dụng của khai phá dữ liệu nói chung và phân cụm dữ liệu nói riêng, phân tích thuật toán K-means và ứng dụng của nó trong việc phân cụm dữ liệu văn bản. Đồng thời, đồ án cũng trình bày ứng dụng phân cụm dữ liệu lịch sử tìm kiếm Google, cho thấy tầm quan trọng của dữ liệu cũng như lợi ích mà phân cụm dữ liệu đem lại. Cùng với đó nâng cao nhận thức của người dùng trong việc bảo vệ dữ liệu cá nhân.

Đánh giá đề tài:

1. Những công việc đã làm được
 - Tìm hiểu các chức năng và ứng dụng của khai phá dữ liệu
 - Tìm hiểu phân cụm dữ liệu
 - Phân tích thuật toán K-means và các thuật toán nhằm cải tiến K-means
 - Đưa ra mô hình phân cụm dữ liệu văn bản, thực hiện chương trình phân cụm dữ liệu lịch sử tìm kiếm cá nhân
2. Những vấn đề chưa giải quyết được
 - Phương pháp khuỷu tay yêu cầu việc phải chạy thuật toán nhiều lần
 - Ứng dụng phân cụm văn bản chỉ có thể làm việc với ngôn ngữ tiếng anh
3. Hướng phát triển đề tài
 - Ứng dụng có thể áp dụng với tiếng Việt các ngôn ngữ khác nhau
 - Triển khai ứng dụng phân cụm văn bản có thể phân cụm các dạng văn bản khác như email, sách ebook ...

TÀI LIỆU THAM KHẢO

- [1]. Ajitesh Kumar. Kmeans Silhouette Score Explained With Python Example
- [2]. Brandon Rose. Document Clustering with Python
- [3]. Vũ Hữu Tiệp. (2018) Machine Learning cơ bản
- [4]. Charu C. Aggarwal. (2015) Data Mining The Textbook
- [5]. Covenant University Ota Ogun State, Nigeria. Data Clustering: (2019)
Algorithms and Its Applications
- [6]. Documentation of scikit-learn 0.21.3
- [7]. Jelili Oyelade & Olufunke Oladipupo & Ibidun Christiana Obagbuwa. (2010)
Application of k Means Clustering algorithm of prediction of Students
Academic Performance.
- [8]. Nguyễn Hồng Phúc. (2018) Kho dữ liệu và khai phá dữ liệu
- [9]. NXLog User Guide. Browser History Logs
- [10]. ODSC - Open Data Science. (2019) Intro to Language Processing with the
NLTK
- [11]. Raffael Vogler. (2014) The tf-idf-Statistic For Keyword Extraction
- [12]. Shahzad Qaiser & Ramsha Ali. (2018) Text Mining: Use of TF-IDF to
Examine the Relevance of Words to Documents