**MỤC LỤC**

[CHƯƠNG I: TỔNG QUAN LÝ THUYẾT 1](#_Toc181389614)

[1.1. Ngôn ngữ lập trình Python. 1](#_Toc181389615)

[1.1.1. Lịch sử ra đời và phát triển. 1](#_Toc181389616)

[1.1.2. Cấu trúc dòng lệnh. 3](#_Toc181389617)

[1.1.3. Cài đặt Python. 5](#_Toc181389618)

[1.2. Các thư viện nổi bật. 7](#_Toc181389619)

[1.2.1. Thư viện Pygame 7](#_Toc181389620)

[1.2.2. Thư viện Random. 10](#_Toc181389621)

[1.2.3. Thư viện Math. 11](#_Toc181389622)

[1.2.4. Thư viện Numpy. 13](#_Toc181389623)

[1.2.5. Thư viện Sklearn. 18](#_Toc181389624)

[1.2.6. Thư viện Scipy. 22](#_Toc181389625)

[1.2.7. Thư viện Matplotlib. 26](#_Toc181389626)

[1.2.8. Tổng kết. 31](#_Toc181389627)

[1.3. Hệ màu. 32](#_Toc181389628)

[1.3.1. Giới thiệu về hệ màu 32](#_Toc181389629)

[1.3.2. Thư viện hệ màu trong Python. 35](#_Toc181389630)

[1.4. Lưu trữ dữ liệu và hình ảnh. 37](#_Toc181389631)

[1.4.1. Tạo và sử dụng danh sách 37](#_Toc181389632)

[1.4.2. Thực hiện các phép toán vector 38](#_Toc181389633)

[1.4.3. Lưu trữ ảnh với vector dữ liệu 39](#_Toc181389634)

[1.4.4. Kết hợp với danh sách Numpy 41](#_Toc181389635)

[1.4.5. Tổng kết 41](#_Toc181389636)

[CHƯƠNG II. THUẬT TOÁN K-MEANS 43](#_Toc181389637)

[2.1. Giới thiệu tổng quát. 43](#_Toc181389638)

[2.2. Các thuật ngữ liên quan 44](#_Toc181389639)

[2.3. Cách K-Means hoạt động. 46](#_Toc181389640)

[2.4. Các ứng dụng của Kmeans 47](#_Toc181389641)

[CHƯƠNG III. ỨNG DỤNG THUẬT TOÁN KMEANS TRONG XỬ LÝ ẢNH 49](#_Toc181389642)

[3.1. Triển khai ứng dụng. 49](#_Toc181389643)

[3.1.1. Xây dựng giao diện 49](#_Toc181389644)

[3.1.2. Sơ đồ hoạt động 50](#_Toc181389645)

[3.1.3. Kết quả khởi chạy 51](#_Toc181389646)

[3.1.4. Nhận xét 52](#_Toc181389647)

[3.1.5. Hướng phát triển 53](#_Toc181389648)

[3.2. Ứng dụng Kmeans trong xử lý ảnh. 54](#_Toc181389649)

[3.2.1. Ứng dụng và các xử lý ảnh 54](#_Toc181389650)

[3.2.2. Kết quả minh họa 54](#_Toc181389651)

[CHƯƠNG IV. TỔNG KẾT 56](#_Toc181389652)

[4.1. Kết quả đạt được 56](#_Toc181389653)

[4.2. Hạn chế 56](#_Toc181389654)

[TÀI LIỆU THAM KHẢO 58](#_Toc181389655)

**DANH SÁCH HÌNH ẢNH**

[Hình 1. 1: Chân dung Guido van Rossum 1](#_Toc181341323)

[Hình 1. 2: Sơ đồ tóm tắt quá trình phát triển của python tới phiên bản 3.7 2](#_Toc181341324)

[Hình 1. 3: Cấu trúc cơ bản của một chương trình python đơn giản. 4](#_Toc181341325)

[Hình 1. 4: Giao diện trang web download chính thức của python 6](#_Toc181341326)

[Hình 1. 5: Một ứng dụng game được thiết kế bằng Pygame 7](#_Toc181341327)

[Hình 1. 6: Cài đặt Pygame bằng terminal và pip 8](#_Toc181341328)

[Hình 1. 7: Thông tin về thư viện random2 sau khi cài đặt thành công. 10](#_Toc181341329)

[Hình 1. 8: Sử dụng thư vện Math để tính diện tích hình tròn với Math.pi 12](#_Toc181341330)

[Hình 1. 9: Phương thức tạo mảng từ danh sách 15](#_Toc181341331)

[Hình 1. 10: Phương thức tạo mảng có kích thước 3\*4 với giá trị 0 16](#_Toc181341332)

[Hình 1. 11: Phương thức tạo mảng từ 0 tới 10 với bước nhảy là 2. 16](#_Toc181341333)

[Hình 1. 12: Một số dạng biểu đồ trong Matplotlib 26](#_Toc181341334)

[Hình 1. 13: Ví dụ minh họa về 9 subplot trên một khung hình. 30](#_Toc181341335)

[Hình 1. 14: Minh họa về legend trong Matplotlib. 31](#_Toc181341336)

[Hình 1. 15: Hệ màu RGB 33](#_Toc181341337)

[Hình 1. 16: Hệ màu CMYK 34](#_Toc181341338)

[Hình 2. 1:Minh họa thuật toán phân cụm K-means 43](#_Toc181389656)

[Hình 3. 1: Giao diện chương trình trực quan hóa Kmeans 49](#_Toc181389657)

[Hình 3. 2: Sơ đồ hoạt động của ứng dụng 50](#_Toc181389658)

[Hình 3. 3: Kết quả chạy chương trình với K = 2 51](#_Toc181389659)

[Hình 3. 4: Kết quả chạy chương trình với K = 3 52](#_Toc181389660)

[Hình 3. 5: Kết quả chạy chương trình với K = 4 52](#_Toc181389661)

[Hình 3. 6: Ảnh đầu vào 55](#_Toc181389662)

[Hình 3. 7: Ảnh đầu ra 55](#_Toc181389663)

**DANH SÁCH BẢNG**

[Bảng 1. 1: Các phương thức chính trong thư viện Random 11](#_Toc181341542)

[Bảng 1. 2: Các phương thức chính của thư viện Math 12](#_Toc181341543)

[Bảng 1. 3: Các phương thức tiền xử lý và chuẩn hóa dữ liệu trong Sklearn 21](#_Toc181341544)

[Bảng 1. 4: Các mô hình trong thư viện Sklearn 21](#_Toc181341545)

[Bảng 1. 5: Các phương thức đánh giá mô hình trong Sklearn. 22](#_Toc181341546)

[Bảng 1. 6: Các phương thức chính trong list 37](#_Toc181341547)

[Bảng 1. 7: Các phương thức chính trong phép toán vector 38](#_Toc181341548)

[Bảng 1. 8: Các phương thức chính trong lưu trữ và xử lí dữ liệu với vector 40](#_Toc181341549)

[Bảng 2. 1: Các thuật ngữ chính trong K-mean và ý nghĩa 44](#_Toc181389677)

# CHƯƠNG I: TỔNG QUAN LÝ THUYẾT

## Ngôn ngữ lập trình Python.

### Lịch sử ra đời và phát triển.

Python là một ngôn ngữ lập trình được phát triển bởi Guido van Rossum vào cuối thập niên 1980, với phiên bản đầu tiên ra mắt vào năm 1991. Ban đầu, Guido phát triển Python như một dự án phụ vào kỳ nghỉ Giáng sinh của mình khi làm việc tại Centrum Wiskunde & Informatica (CWI) ở Hà Lan. Ngôn ngữ này lấy cảm hứng từ nhiều ngôn ngữ khác, đặc biệt là ABC – một ngôn ngữ lập trình giáo dục nhằm giúp người mới học lập trình dễ dàng hơn – nhưng cũng chịu ảnh hưởng của C, C++, và Modula-3. Guido đã chọn cái tên "Python" vì yêu thích chương trình hài kịch của nhóm Monty Python, điều này thể hiện phần nào tính chất nhẹ nhàng, hài hước trong triết lý thiết kế của ngôn ngữ này.



**Hình 1. 1: Chân dung Guido van Rossum**

Trong suốt thập niên 1990, Python đã dần phát triển thành một ngôn ngữ lập trình có cộng đồng người dùng tích cực. Một trong những yếu tố quan trọng giúp Python phổ biến hơn là sự tập trung vào tính dễ đọc của mã nguồn, cú pháp rõ ràng và nhất quán. Ngay từ đầu, Guido van Rossum đã đặt ra triết lý “bắt buộc phải dễ đọc”, và điều này đã giúp Python khác biệt so với nhiều ngôn ngữ khác thời bấy giờ. Python phiên bản 2.0 ra mắt vào năm 2000, mang lại nhiều cải tiến đáng kể như thu gom rác tự động (garbage collection) và hỗ trợ Unicode, giúp Python dễ dàng mở rộng ra các thị trường quốc tế. Tuy nhiên, Python 2 vẫn tồn tại một số hạn chế về hiệu suất và khả năng mở rộng, khiến các nhà phát triển bắt đầu nghĩ đến một bước phát triển vượt bậc.

A diagram of a diagram

Description automatically generated with medium confidence

**Hình 1. 2: Sơ đồ tóm tắt quá trình phát triển của python tới phiên bản 3.7**

Năm 2008, Python 3.0 được phát hành, đánh dấu một thay đổi lớn với nhiều cải tiến trong cú pháp và cấu trúc dữ liệu. Python 3 được thiết kế để khắc phục những yếu điểm của Python 2 và mang đến một nền tảng mạnh mẽ hơn cho các ứng dụng phát triển hiện đại. Tuy nhiên, sự khác biệt lớn giữa Python 2 và Python 3 đã khiến cho quá trình chuyển đổi trở nên khó khăn, đặc biệt đối với các dự án và thư viện lớn đã được viết bằng Python 2. Kể từ đó, Python tiếp tục phát triển với các phiên bản 3.x, không ngừng cập nhật và bổ sung các tính năng mới. Phiên bản Python 2 cuối cùng (Python 2.7) được duy trì đến ngày 1 tháng 1 năm 2020, khi Python 2 chính thức bị ngừng hỗ trợ, đánh dấu một bước chuyển đổi hoàn toàn sang Python 3.

Trong thập niên 2010, Python đã trở thành một trong những ngôn ngữ lập trình phổ biến nhất thế giới nhờ vào sự phát triển của khoa học dữ liệu, học máy, và trí tuệ nhân tạo. Các thư viện như NumPy, Pandas, TensorFlow và PyTorch đã giúp Python trở thành một công cụ không thể thiếu trong các lĩnh vực này. Được biết đến với sự linh hoạt và tính đơn giản, Python không chỉ phổ biến trong cộng đồng khoa học dữ liệu mà còn trở thành ngôn ngữ lập trình chính trong nhiều lĩnh vực khác như phát triển web, phát triển ứng dụng di động, và tự động hóa.

Tính đến nay, Python vẫn tiếp tục phát triển mạnh mẽ với cộng đồng đông đảo và sự đóng góp của hàng nghìn nhà phát triển trên toàn thế giới. Với triết lý "Simple is better than complex" (Đơn giản hơn phức tạp) từ Zen of Python, Python luôn duy trì được tính nhất quán trong cú pháp và dễ học, dễ hiểu. Python Foundation, tổ chức phi lợi nhuận quản lý và phát triển Python, vẫn tiếp tục thúc đẩy ngôn ngữ này đi xa hơn, tạo điều kiện cho sự ra đời của các tính năng và cải tiến mới trong tương lai. Điều này giúp Python không chỉ duy trì được vị trí hàng đầu trong thế giới lập trình mà còn tiếp tục phát triển, thích nghi với những thay đổi của ngành công nghệ.

### Cấu trúc dòng lệnh.

Cấu trúc của Python được thiết kế với mục tiêu đơn giản và dễ đọc, giúp người học dễ dàng tiếp cận và các nhà phát triển có thể nhanh chóng xây dựng và bảo trì các ứng dụng. Python là một ngôn ngữ lập trình dạng thông dịch, có nghĩa là mã nguồn được thực thi trực tiếp mà không cần biên dịch trước thành mã máy, điều này giúp rút ngắn thời gian phát triển và thử nghiệm. Một trong những đặc điểm nổi bật của Python là cấu trúc khối được xác định bởi thụt lề (indentation), chứ không phải dấu ngoặc như trong các ngôn ngữ khác. Quy tắc này không chỉ giúp mã nguồn dễ đọc mà còn đảm bảo tính nhất quán và tránh lỗi cú pháp do lạm dụng dấu ngoặc.

Python hỗ trợ nhiều kiểu dữ liệu tích hợp sẵn, từ kiểu cơ bản như số nguyên (integer), số thực (float), chuỗi (string), đến các kiểu cấu trúc dữ liệu phức tạp hơn như danh sách (list), từ điển (dictionary), tập hợp (set), và tuple. Các kiểu dữ liệu này được quản lý động, tức là Python tự động gán kiểu dữ liệu mà không cần khai báo, giúp tối ưu hóa việc quản lý và sử dụng bộ nhớ. Python cũng có cơ chế thu gom rác (garbage collection) tự động, giúp quản lý bộ nhớ và giảm thiểu rò rỉ tài nguyên, làm cho ngôn ngữ này trở nên hiệu quả hơn trong các dự án lớn.

Python còn hỗ trợ lập trình hướng đối tượng (OOP), lập trình thủ tục và lập trình hàm, cho phép người dùng lựa chọn phong cách phù hợp nhất cho dự án của mình. Lập trình hướng đối tượng trong Python hỗ trợ các khái niệm cơ bản như lớp (class), đối tượng (object), kế thừa (inheritance), đóng gói (encapsulation), và đa hình (polymorphism), giúp tổ chức và quản lý mã nguồn theo mô hình rõ ràng và dễ bảo trì. Đối với các yêu cầu xử lý nhanh và tối ưu hiệu suất, Python cung cấp các mô-đun và thư viện tích hợp giúp viết các hàm xử lý phức tạp và thực hiện các phép tính toán một cách nhanh chóng.

A diagram of a program

Description automatically generated with medium confidence

*Hình 1. 3: Cấu trúc cơ bản của một chương trình python đơn giản.*

Một điểm đáng chú ý khác trong cấu trúc của Python là khả năng mở rộng thông qua các thư viện và mô-đun bên ngoài. Python có một kho thư viện phong phú, được phát triển bởi cộng đồng lớn mạnh, từ các thư viện chuẩn cho đến các thư viện nâng cao cho các lĩnh vực như khoa học dữ liệu, học máy, phát triển web, và phân tích dữ liệu lớn. Người dùng có thể cài đặt thêm các thư viện từ Python Package Index (PyPI) chỉ với một dòng lệnh, giúp tiết kiệm thời gian và mở rộng khả năng của ngôn ngữ. Điều này cũng đồng nghĩa với việc Python có thể tích hợp dễ dàng với các công nghệ khác, tạo điều kiện thuận lợi cho các dự án đa nền tảng và đa ngôn ngữ.

Python sử dụng một trình thông dịch đơn (interpreter) trong quá trình thực thi mã, cho phép phát hiện lỗi dễ dàng trong từng dòng mã và tăng khả năng phản hồi nhanh cho nhà phát triển khi kiểm thử. Trình thông dịch này cũng hỗ trợ nhiều hệ điều hành khác nhau như Windows, macOS, và Linux, giúp mã Python có tính di động cao và dễ dàng chạy trên nhiều nền tảng mà không cần điều chỉnh nhiều. Python cũng hỗ trợ các công cụ để đóng gói mã nguồn thành các tệp thực thi, giúp việc triển khai và phân phối ứng dụng trở nên thuận tiện hơn.

Với cấu trúc đơn giản và tính linh hoạt cao, Python không chỉ dễ học mà còn là một công cụ mạnh mẽ cho các nhà phát triển chuyên nghiệp. Sự kết hợp giữa tính năng đa dạng, cú pháp rõ ràng, và khả năng mở rộng mạnh mẽ đã giúp Python trở thành một trong những ngôn ngữ lập trình được ưa chuộng nhất trong nhiều lĩnh vực công nghệ.

### Cài đặt Python.

Python hỗ trợ nhiều nền tảng, nên các bước này sẽ giúp bạn cài đặt Python nhanh chóng và sẵn sàng sử dụng. Để cài đặt Python, có thể truy cập trang web chính thức của Python tại <https://www.python.org/downloads/>. Nơi đây chứa lịch sử các phiên bản cũng như các tệp tin cài đặt. Người dùng có thể tùy chọn các phiên bản phù hợp với cầu hình và hệ điền hành của mình.

A screenshot of a computer

Description automatically generated

Hình 1. : Giao diện trang web download chính thức của python

**Cài đặt Python trên Windows**

1. Mở tệp cài đặt Python vừa tải xuống (python-x.y.z.exe), trong đó x.y.z là số phiên bản của Python.
2. Ở cửa sổ cài đặt đầu tiên, đánh dấu chọn *“Add Python to PATH”* để Python tự động thêm vào biến môi trường của hệ thống. Đây là bước quan trọng giúp bạn có thể sử dụng lệnh python trên mọi thư mục trong Command Prompt.
3. Nhấp vào Install Now để cài đặt Python theo mặc định, hoặc bạn cũng có thể chọn Customize installation để thay đổi thư mục cài đặt và các tính năng bổ sung.
4. Chờ quá trình cài đặt hoàn tất. Khi cài đặt xong, mở Command Prompt và gõ “*python –version”* để kiểm tra.

**Cài đặt Python trên macOS**

Python thường được cài đặt sẵn trên macOS, nhưng phiên bản này có thể đã cũ. Có thể cài phiên bản mới hơn bằng các bước sau:

1. Mở tệp .pkg vừa tải từ trang Python.org và làm theo hướng dẫn trên màn hình để cài đặt.
2. Sau khi cài đặt xong, mở Terminal và kiểm tra bằng lệnh “*python3 --version*”.

## 1.2. Các thư viện nổi bật.

### Thư viện Pygame

#### Giới thiệu.

Pygame là một thư viện được phát triển để hỗ trợ lập trình game trong Python, giúp người dùng dễ dàng xây dựng và quản lý các trò chơi 2D. Được phát triển lần đầu vào năm 2000 bởi Pete Shinners, Pygame hiện đã trở thành một trong những thư viện phổ biến nhất cho lập trình game với Python. Thư viện này cung cấp một bộ công cụ và phương thức mạnh mẽ để xử lý đồ họa, âm thanh, sự kiện và chuyển động, giúp người dùng tập trung vào logic trò chơi thay vì phải tự viết các hàm phức tạp.



Hình 1. : Một ứng dụng game được thiết kế bằng Pygame

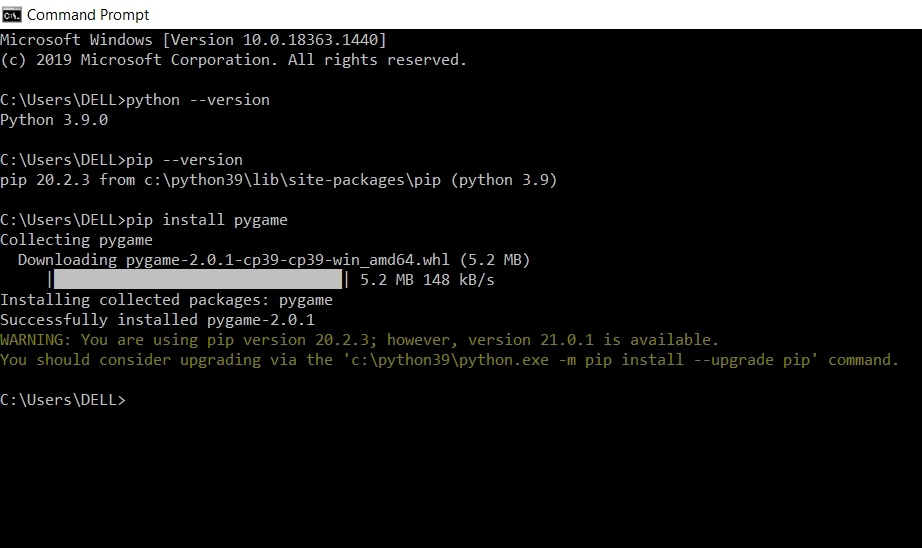
Pygame hoạt động dựa trên thư viện SDL (Simple DirectMedia Layer) – một thư viện mã nguồn mở được thiết kế cho các ứng dụng đồ họa và âm thanh. SDL đảm bảo tính ổn định và hiệu suất cao, đồng thời hỗ trợ đa nền tảng (Windows, macOS, Linux), cho phép các trò chơi Pygame có thể chạy trên nhiều hệ điều hành khác nhau mà không cần thay đổi mã nguồn.

Pygame giúp đơn giản hóa cấu trúc một trò chơi thông qua các mô-đun như surface (bề mặt), sprite (hình ảnh động), và rect (hình chữ nhật). Trong Pygame, các phần tử như nhân vật, đối tượng, hoặc các yếu tố đồ họa khác sẽ được quản lý qua các mô-đun này. Ví dụ, surface là một đối tượng để hiển thị đồ họa và lưu trữ hình ảnh; các đối tượng này có thể dễ dàng di chuyển, vẽ, và cập nhật trong trò chơi.

Để tạo trò chơi với Pygame, người dùng phải khởi tạo thư viện và thiết lập một vòng lặp sự kiện (event loop) – nơi xử lý các sự kiện như nhấn phím, di chuyển chuột, và cập nhật hình ảnh. Vòng lặp sự kiện là trung tâm của mọi trò chơi, giúp trò chơi có thể tương tác với người chơi và cập nhật các khung hình liên tục.

#### Cách cài đặt

Để cài đặt Pygame, có thể sử dụng trình quản lý gói pip của Python. Pygame có sẵn cho các hệ điều hành phổ biến như Windows, macOS và Linux.



Hình 1. : Cài đặt Pygame bằng terminal và pip

Dưới đây là các bước cài đặt chi tiết.

* + Đảm bảo Python và pip đã được cài đặt**:** Trước khi cài đặt Pygame, bạn cần chắc chắn rằng Python và pip đã được cài đặt. Bạn có thể kiểm tra bằng cách mở Terminal (trên macOS hoặc Linux) hoặc Command Prompt (trên Windows) và nhập: *python –version.*
  + Cài đặt Pygame qua pip: Sau khi Python và pip đã được cài đặt, chạy lệnh sau trong terminal để cài đặt pygame: *pip install pygame.*
  + Kiểm tra cài đặt: Sau khi quá trình cài đặt thành công, có thể chạy đoạn mã sau để kiểm tra lại lần cuối: *import pygame - print(pygame.ver).* Nếu không có lỗi và phiên bản Pygame được hiển thị, cài đặt đã thành công và Pygame sẵn sàng để sử dụng.

#### Các chức năng chính

Pygame cung cấp nhiều tính năng để xử lý đồ họa và âm thanh một cách hiệu quả, từ đó tối ưu quá được quá trình làm game của người lập trình. Một số chức năng chính nổi bật bao gồm:

* Xử lý đồ họa: Thư viện hỗ trợ hiển thị ảnh và tạo các hiệu ứng hình ảnh 2D như vẽ các hình cơ bản (hình chữ nhật, đường thẳng, hình tròn) và hỗ trợ xử lý các hiệu ứng trong suốt.
* Âm thanh: Pygame hỗ trợ phát nhạc và âm thanh hiệu ứng (sound effect), cho phép bạn dễ dàng thêm các tệp âm thanh để làm tăng trải nghiệm trong trò chơi. Mô-đun mixer trong Pygame hỗ trợ phát đồng thời nhiều luồng âm thanh, quản lý âm lượng, và điều chỉnh tần số âm thanh.
* Xử lý sự kiện: Pygame cung cấp một hệ thống xử lý sự kiện đơn giản nhưng hiệu quả, giúp dễ dàng theo dõi các tương tác từ bàn phím, chuột và các thiết bị khác. Mỗi khi người chơi nhấn phím hoặc di chuyển chuột, sự kiện đó sẽ được xử lý thông qua vòng lặp sự kiện.
* Sprite và Animation: Pygame cung cấp các lớp hỗ trợ cho việc sử dụng sprite – hình ảnh động hoặc đối tượng đồ họa, rất hữu ích trong các trò chơi để tạo các nhân vật hoặc đối tượng có thể di chuyển. Với các đối tượng sprite, người dùng có thể dễ dàng tạo các hiệu ứng chuyển động và va chạm giữa các đối tượng.

### Thư viện Random.

#### Giới thiệu.

Thư viện random trong Python là một thư viện tiêu chuẩn được thiết kế để tạo các số ngẫu nhiên và thực hiện các thao tác liên quan đến ngẫu nhiên hóa. Thư viện này cung cấp các hàm hữu ích cho việc tạo các số ngẫu nhiên, lựa chọn ngẫu nhiên từ một danh sách, xáo trộn thứ tự các phần tử, và thực hiện nhiều thao tác liên quan đến xác suất. Thư viện random rất hữu ích trong các lĩnh vực như lập trình game, mô phỏng, phân tích thống kê và các thuật toán yêu cầu tính ngẫu nhiên.

#### Cách cài đặt.

Tương tự như cách cài đặt Pygame, có thể cài đặt thư viện Random thông qua gói quản lý pip. Chạy đoạn mã sau trong terminal để cài đặt Random trong Python 3: *pip install random2.* Sau đó kiểm tra lại version Random bằng câu lệnh: *pip show random2;* kết quả sẽ tương tự như hình dưới:

A black screen with white text

Description automatically generated

Hình 1. : Thông tin về thư viện random2 sau khi cài đặt thành công.

#### Các phương thức chính.

Thư viện Random hỗ trợ nhiều phương thức, chúng mang đến tính linh hoạt cao khi làm việc với các bài toán cần tính ngẫu nhiên, từ tạo số ngẫu nhiên đơn giản đến các phân phối xác suất phức tạp.

Dưới đây là bảng liệt kê các phương thức chính trong thư viện random của Python và mô tả ngắn gọn về từng phương thức.

Bảng 1. : Các phương thức chính trong thư viện Random

|  |  |
| --- | --- |
| **Phương thức** | **Mô tả** |
| random() | Trả về một số thực ngẫu nhiên trong khoảng từ 0.0 đến 1.0. |
| randint(a, b) | Trả về một số nguyên ngẫu nhiên nằm trong khoảng từ a đến b (bao gồm cả a và b). |
| uniform(a, b) | Trả về một số thực ngẫu nhiên nằm trong khoảng từ a đến b. |
| randrange(start, stop, step) | Trả về một số nguyên ngẫu nhiên trong khoảng từ start đến stop, với bước nhảy là step. |
| choice(sequence) | Chọn ngẫu nhiên một phần tử từ một chuỗi, danh sách, hoặc tập hợp. |
| choices(population, weights=None, k=1) | Chọn ngẫu nhiên k phần tử từ một danh sách, cho phép lặp lại và tùy chọn trọng số. |
| shuffle(sequence) | Xáo trộn thứ tự các phần tử trong danh sách (thay đổi danh sách gốc). |
| sample(population, k) | Lấy ngẫu nhiên k phần tử từ một danh sách hoặc tập hợp mà không trùng lặp. |
| betavariate(alpha, beta) | Trả về một số ngẫu nhiên theo phân phối Beta với các tham số alpha và beta. |

### Thư viện Math.

#### Giới thiệu.

Thư viện math trong Python là một thư viện tiêu chuẩn được thiết kế để cung cấp các hàm toán học cơ bản và các phép toán phức tạp hơn. Thư viện này chứa nhiều hàm hữu ích cho các tính toán số học, từ các phép toán cơ bản như cộng, trừ, nhân, chia đến các phép toán nâng cao như tính toán logarithm, căn bậc hai, và các hàm lượng giác.

Cài đặt thư viện trên bằng terminal với dòng lệnh: *pip install math.* Sau quá trình cài đặt thành công, sử dụng *ímport math* để sử dụng math trong đoạn mã của chương trình. Math có thể giúp tính toán các số liệu, công thức một các nhanh chóng. Ví dụ vê việc sử dụng hàm *math.pi* để tính diện tích hình tròn như sau:

A computer screen shot of a black background with white text

Description automatically generated

Hình 1. : Sử dụng thư vện Math để tính diện tích hình tròn với Math.pi

#### Các phương thức chính.

Dưới đây là bảng liệt kê các phương thức chính trong thư viện math của Python cùng với chức năng chính của nó.

Bảng 1. : Các phương thức chính của thư viện Math

|  |  |
| --- | --- |
| **Phương thức** | **Mô tả** |
| math.pi | Hằng số π, giá trị khoảng 3.14159. |
| math.e | Hằng số Euler (e), giá trị khoảng 2.71828. |
| math.sin(x) | Trả về giá trị sin của góc x (đơn vị radian). |
| math.cos(x) | Trả về giá trị cos của góc x. |
| math.tan(x) | Trả về giá trị tan của góc x. |
| math.asin(x) | Trả về giá trị arcsin của x, kết quả nằm trong khoảng [-π/2, π/2]. |
| math.acos(x) | Trả về giá trị arccos của x, kết quả nằm trong khoảng [0, π]. |
| math.atan(x) | Trả về giá trị arctan của x, kết quả nằm trong khoảng [-π/2, π/2]. |
| math.sqrt(x) | Trả về căn bậc hai của x. |
| math.pow(x, y) | Tính lũy thừa của x với số mũ y (x^y). |
| math.factorial(x) | Tính giai thừa của số nguyên không âm x. |
| math.log(x[, base]) | Tính logarit của x với cơ số base. Nếu không có base, trả về logarit tự nhiên (cơ số e). |
| math.log10(x) | Tính logarit cơ số 10 của x. |
| math.ceil(x) | Làm tròn số x lên tới số nguyên gần nhất. |
| math.floor(x) | Làm tròn số x xuống tới số nguyên gần nhất. |
| math.trunc(x) | Trả về phần nguyên của số x. |

### Thư viện Numpy.

#### Giới thiệu.

NumPy, viết tắt của “Numerical Python,” là một thư viện toán học mã nguồn mở và mạnh mẽ trong Python, rất phổ biến trong giới lập trình khoa học dữ liệu và máy học. Được phát triển đầu tiên bởi Jim Hugunin và sau này hoàn thiện bởi Travis Oliphant vào năm 2005, NumPy đã nhanh chóng trở thành một công cụ không thể thiếu, đặc biệt với các lập trình viên chuyên về khoa học dữ liệu, khi nhu cầu xử lý các phép toán với mảng và ma trận lớn ngày càng tăng. NumPy cung cấp các hàm toán học tối ưu, mang lại hiệu suất vượt trội so với việc chỉ sử dụng Python thuần túy, giúp giảm thiểu thời gian xử lý với các dữ liệu phức tạp và kích thước lớn.

Các tính năng chính của NumPy bao gồm:

* Hỗ trợ mảng nhiều chiều (ndarray): NumPy cung cấp kiểu dữ liệu mảng n-dimensional array (ndarray), cho phép thao tác với các mảng có số chiều bất kỳ, từ mảng một chiều (vector) đến các ma trận nhiều chiều.
* Các phép toán số học và logic trên mảng: Thư viện này bao gồm hàng loạt các phép toán như cộng, trừ, nhân, chia, mũ và các phép toán logic giúp cho việc xử lý dữ liệu trở nên đơn giản và dễ dàng. NumPy cho phép áp dụng phép toán đồng thời trên từng phần tử của mảng (broadcasting), giúp tối ưu hóa tốc độ xử lý.
* Các hàm thống kê, biến đổi và xử lý dữ liệu: NumPy cung cấp các hàm tính toán các giá trị thống kê như tổng, trung bình, phương sai, và độ lệch chuẩn. Ngoài ra, các phép biến đổi Fourier và các phương pháp xử lý tín hiệu như fft cũng được hỗ trợ, giúp xử lý dữ liệu sóng hoặc tín hiệu nhanh chóng.
* Công cụ đại số tuyến tính và tạo số ngẫu nhiên: NumPy tích hợp sẵn các công cụ để thực hiện các phép toán đại số tuyến tính như nhân ma trận, phân rã trị riêng, và giải phương trình tuyến tính. Thư viện cũng cung cấp một loạt các hàm để tạo số ngẫu nhiên, hỗ trợ trong việc mô phỏng và thực hiện các phép thử thống kê.
* Tích hợp với mã C/C++ và Fortran: NumPy cho phép tích hợp các đoạn mã viết bằng ngôn ngữ C/C++ hoặc Fortran, giúp cải thiện tốc độ và hiệu suất khi cần thiết. Điều này đặc biệt hữu ích khi xử lý các bài toán phức tạp hoặc yêu cầu tính toán tốc độ cao trong thời gian ngắn.

NumPy còn có khả năng tương tác cao với các thư viện khác trong hệ sinh thái Python, chẳng hạn như SciPy (Python Scientific) và Matplotlib (thư viện vẽ đồ thị), tạo thành một bộ công cụ phân tích và mô phỏng mạnh mẽ. Chính nhờ sự kết hợp này mà Python với NumPy có thể thay thế MatLab – phần mềm thương mại thường được dùng trong tính toán kỹ thuật – đặc biệt là khi NumPy là một thư viện mã nguồn mở và miễn phí, trong khi MatLab yêu cầu trả phí sử dụng.

Ngoài ra, NumPy còn là nền tảng cho rất nhiều thư viện và công cụ trong lĩnh vực khoa học dữ liệu, như Pandas (thư viện xử lý dữ liệu) và các công cụ học máy như TensorFlow và PyTorch. Hệ sinh thái khoa học dữ liệu trong Python phần lớn được xây dựng dựa trên khả năng của NumPy trong việc cung cấp các công cụ tính toán mạnh mẽ và hiệu quả. Vì vậy, đối với bất kỳ ai muốn trở thành lập trình viên khoa học dữ liệu chuyên nghiệp, việc nắm vững NumPy là điều thiết yếu, vì thư viện này cung cấp nền tảng toán học và thống kê cần thiết để làm việc với các mô hình học máy và thuật toán phân tích dữ liệu phức tạp.

NumPy là một thư viện không thể thiếu trong Python, đặc biệt là với các lập trình viên chuyên về tính toán khoa học và phân tích dữ liệu. Sự phát triển của NumPy đã giúp Python trở thành một công cụ lý tưởng cho tính toán khoa học và kỹ thuật, vượt ra ngoài phạm vi của một ngôn ngữ lập trình thuần túy

#### Cách cài đặt.

Để cài đặt thư viện Numpy trong Python, người dùng có thể dùng PIP và đảm bảo Python phải ở phiên bản 3.6 trở lên . Sau đây là các bước cài đặt thư viện này:

Bước 1: Mở Command Prompt hoặc Terminal

Trên Windows: Tìm kiếm "Command Prompt" trong menu Start.

Trên macOS hoặc Linux: Mở Terminal từ Spotlight hoặc từ menu ứng dụng.

Bước 2: Chạy Lệnh Cài Đặt NumPy

Nhập lệnh sau vào Command Prompt hoặc Terminal:

*pip install numpy*

Bước 3: Kiểm Tra Cài Đặt

Sau khi quá trình cài đặt hoàn tất, mở Python bằng cách gõ *python* hoặc *python3* vào Command Prompt/Terminal và nhập lệnh sau để kiểm tra:

*import numpy as np*

*print(np.\_\_version\_\_)*

Nếu không có lỗi nào xảy ra và phiên bản của NumPy được in ra, việc cài đặt đã thành công.

#### Các phương thức chính.

1. Phương thức tạo mảng: NumPy cung cấp nhiều phương thức để tạo các mảng đa chiều (mảng NumPy), ví dụ như sau:

* np.array(): Tạo mảng từ danh sách hoặc tuple.



Hình 1. : Phương thức tạo mảng từ danh sách

* np.zeros(): Tạo mảng chứa các giá trị 0



Hình 1. : Phương thức tạo mảng có kích thước 3\*4 với giá trị 0

* - np.arange(): Tạo mảng với các số liên tiếp.



Hình 1. : Phương thức tạo mảng từ 0 tới 10 với bước nhảy là 2.

Kết quả sẽ trả về mảng [0,2,4,6,8]

Ngoài các ví dụ nói trên, Numpy cung cấp nhiều các phương thức khác hỗ trợ lập trình viên trong quá trình tạo các đoạn mã và thực hiện các công việc tính toán trên đó.

1. Các phương thức tính toán trên mảng. NumPy hỗ trợ thực hiện nhiều phép toán số học và hàm tính toán trên mảng, có thể kể đến một vài phương thức điển hình như sau:

Phép toán số học: NumPy cho phép thực hiện các phép toán cộng, trừ, nhân, chia trực tiếp trên các mảng.

*arr1 = np.array([1, 2, 3])*

*arr2 = np.array([4, 5, 6])*

*sum\_array = arr1 + arr2*

Đoạn mã trên sẽ cho kết quả là mảng [5, 7, 9]. Bên cạnh đó, còn có các phương thức như *np.add(), np.subtract(), np.multiply(), np.divide()*: Các hàm hỗ trợ phép cộng, trừ, nhân, chia. np.sqrt(): Tính căn bậc hai cho từng phần tử trong mảng. sqrt\_array = np.sqrt(arr1). np.exp(): Tính mũ e cho từng phần tử

1. Các phương thức thao tác trên mảng: Numpy cung cấp sẵn các hàm, phương thức cho phép thao tác trên cấu trúc và nội dung của mảng. Các phương thức và hàm điển hình có thể kể đến là:

* np.reshape(): Thay đổi hình dạng của mảng mà không làm thay đổi dữ liệu.

*reshaped\_array = arr.reshape((2, 2))*

* np.concatenate(): Nối hai mảng lại với nhau.

*concatenated\_array = np.concatenate((arr1, arr2))*

* np.split(): Chia mảng thành nhiều phần.

*split\_array = np.split(arr, 3*

* np.transpose(): Hoán vị các hàng và cột của mảng.

*transposed\_array = np.transpose(arr).*

1. Các phương thức thống kê: Các phương thức thống kê giúp tính toán các giá trị đặc trưng của dữ liệu trong mảng. Các đặc trung nổi bật bao gồm giá trị trung bình, trung vị, độ lệch chuẩn.v..v.

* np.mean(): Tính giá trị trung bình của các phần tử trong mảng.

*mean\_value = np.mean(arr)*

* np.median(): Tính giá trị trung vị.

*median\_value = np.median(arr)*

* np.std(): Tính độ lệch chuẩn.

*std\_value = np.std(arr)*

* np.sum(): Tính tổng các phần tử.

*sum\_value = np.sum(arr)*

1. Các phương thức đại số tuyến tính: NumPy hỗ trợ các phép toán đại số tuyến tính, đặc biệt hữu ích cho các bài toán khoa học và kỹ thuật như sau:

* np.dot(): Tính tích vô hướng của hai mảng (còn gọi là tích trong).

*dot\_product = np.dot(arr1, arr2)*

* np.linalg.inv(): Tính ma trận nghịch đảo.

*matrix = np.array([[1, 2], [3, 4]])*

*inverse\_matrix = np.linalg.inv(matrix)*

* np.linalg.det(): Tính định thức của ma trận.

*determinant = np.linalg.det(matrix)*

* np.linalg.eig(): Tính trị riêng và vector riêng của ma trận.

*eigenvalues, eigenvectors = np.linalg.eig(matrix)*

### Thư viện Sklearn.

#### Giới thiệu.

Scikit-learn (thường được viết tắt là Sklearn) là một thư viện Python hàng đầu dành cho các thuật toán học máy (machine learning) và mô hình hóa thống kê. Với Sklearn, người dùng có thể thực hiện các bài toán liên quan đến phân loại (classification), hồi quy (regression), phân cụm (clustering), và giảm chiều dữ liệu (dimensionality reduction) một cách hiệu quả. Sklearn được sử dụng rộng rãi trong cộng đồng học máy bởi sự đa dạng trong tính năng và sự tiện dụng trong cách sử dụng, mang lại hiệu quả cao trong việc phát triển và triển khai các mô hình học máy.

Scikit-learn ban đầu được phát triển bởi David Cournapeau trong dự án mùa hè của Google vào năm 2007. Sau đó, Matthieu Brucher tham gia và tiếp tục phát triển dự án như một phần của luận văn tiến sĩ. Năm 2010, dự án nhận được sự tài trợ từ tổ chức INRIA, và phiên bản đầu tiên của thư viện (v0.1 beta) đã ra mắt vào tháng 1 cùng năm. Hiện nay, Scikit-learn vẫn tiếp tục được phát triển bởi đội ngũ nghiên cứu gồm hơn 30 thành viên đến từ các tổ chức như INRIA, Google, Tinyclues và Python Software Foundation.

Để cài đặt Scikit-learn, cần có các thư viện nền tảng quan trọng như SciPy (công cụ tính toán khoa học), NumPy (tính toán mảng số và ma trận đa chiều), và một số thư viện bổ trợ khác như Matplotlib, IPython, SymPy, và Pandas. Các thư viện này cung cấp các công cụ cần thiết cho việc tính toán khoa học, xử lý dữ liệu và trực quan hóa, giúp Sklearn thực hiện các thao tác và thuật toán học máy hiệu quả.

Mặc dù được xây dựng trên nền Python, một số thư viện cốt lõi của Sklearn, như NumPy và LibSVM, được viết bằng ngôn ngữ C để cải thiện hiệu suất tính toán, đặc biệt trong xử lý dữ liệu và tối ưu hóa các mô hình học máy

Scikit-learn mang đến các công cụ mạnh mẽ cho học máy với các ưu điểm sau:

* Công cụ xử lý dữ liệu: Sklearn cung cấp các công cụ linh hoạt giúp xử lý dữ liệu nhanh chóng, bao gồm chuẩn hóa dữ liệu, mã hóa biến, và hỗ trợ xử lý dữ liệu thô.
* Thư viện mã nguồn mở và miễn phí: Được phát triển mã nguồn mở, Sklearn không chỉ tiết kiệm chi phí mà còn có cộng đồng hỗ trợ rộng lớn. Người dùng dễ dàng tìm kiếm giải pháp và trợ giúp khi gặp khó khăn trong quá trình phát triển ứng dụng.
* Các mô hình và công cụ đánh giá phong phú: Thư viện bao gồm nhiều thuật toán học máy, từ các mô hình đơn giản như Hồi quy tuyến tính (Linear Regression) đến các mô hình phức tạp hơn như Random Forest và SVM. Sklearn cũng cung cấp công cụ đánh giá như cross-validation, giúp kiểm tra độ chính xác của mô hình.
* Tiền xử lý dữ liệu: Sklearn có các công cụ chuẩn hóa dữ liệu và mã hóa biến dễ sử dụng, đảm bảo dữ liệu đầu vào phù hợp với các thuật toán. Đây là giai đoạn quan trọng trong bất kỳ dự án nào.
* Tối ưu hóa tham số: Thư viện hỗ trợ các phương pháp tối ưu hóa như GridSearchCV và RandomizedSearchCV để tìm tham số tốt nhất cho mô hình, giúp tăng độ chính xác và đáp ứng yêu cầu từ người dùng.

Scikit-learn cung cấp nhiều tính năng và phương thức quan trọng, phục vụ các giai đoạn khác nhau của quy trình xây dựng mô hình học máy:

* Tiền xử lý dữ liệu (Data Preprocessing): Các công cụ như StandardScaler để chuẩn hóa dữ liệu, LabelEncoder và OneHotEncoder để mã hóa biến phân loại.
* Chọn lựa mô hình và đánh giá (Model Selection & Evaluation): Các công cụ như train\_test\_split để chia dữ liệu và cross\_val\_score để đánh giá mô hình thông qua cross-validation.
* Các thuật toán học máy (Machine Learning Algorithms): Bao gồm các phương pháp như LogisticRegression, KNeighborsClassifier, DecisionTreeClassifier, RandomForestClassifier cho phân loại; LinearRegression, Ridge, Lasso cho hồi quy; và KMeans, DBSCAN cho phân cụm.

Scikit-learn là một công cụ không thể thiếu trong các dự án học máy nhờ tính tiện dụng, hiệu quả và cộng đồng hỗ trợ mạnh mẽ. Thư viện này đáp ứng tốt các yêu cầu của người dùng, từ tiền xử lý dữ liệu, xây dựng mô hình đến đánh giá và tối ưu hóa. Sklearn là sự lựa chọn tối ưu cho cả người mới bắt đầu và các nhà phát triển chuyên nghiệp trong lĩnh vực học máy.

#### Cách cài đặt.

Để cài đặt thư viện Scikit-learn trong Python, người dùng có thể dùng PIP và đảm bảo Python phải ở phiên bản 3.7 trở lên . Sau đây là các bước cài đặt thư viện này.

Bước 1: Hãy mở command prompt hay terminal

Bước 2: Sau đó, chạy lệnh bên dưới để cài đặt thư viện này.

*pip install -U scikit-learn*

Chú ý: ‘-U’ nhằm cho thấy thư viện Scikit-learn đang cài phiên bản mới nhất. Tuy nhiên, người dùng bỏ ‘-U’ để cài phiên bản riêng biệt.

Bước 3: Chờ cho tới khi quá trình cài đặt kết thúc. Khi đã cài đặt xong, người dùng thực hiện import nó trong những script để bắt đầu sử dụng thư viện này.

*import sklearn*

Nếu không thì quá trình cài đặt thư viện đã hoàn tất. Lúc này, hãy bắt tay trải nghiệm những tính năng hữu ích từ bộ công cụ trong thư viện Scikit-learn trong Python.

#### Các phương thức chính

1. Tiền xử lí dữ liệu: Sklearn cung cấp cho người dùng các phương thức tiền xử lý trong bao gồm chuẩn hóa, mã hóa, và xử lý dữ liệu thiếu. Các phương thức chính được trình bày trong bảng sau:

Bảng 1. : Các phương thức tiền xử lý và chuẩn hóa dữ liệu trong Sklearn

|  |  |
| --- | --- |
| **Tên phương thức** | **Chức năng** |
| StandardScaler | Chuẩn hóa dữ liệu về phân phối chuẩn với trung bình 0 và phương sai 1. |
| MinMaxScaler | Chuyển đổi dữ liệu về khoảng từ 0 đến 1 |
| LabelEncoder | Mã hóa các nhãn phân loại thành số nguyên |
| OneHotEncoder | Mã hóa các biến phân loại thành dạng nhị phân |
| train\_test\_split | Phân chia dữ liệu thành các pha khác nhau |

.

1. Xây dựng mô hình học máy: Sklearn được tích hợp sẵn nhiều mô hình học máy, đã bao gồm các phương thức đánh giá cho từng loại mô hình riêng. Các mô hình bao gồm hồi quy, phân loại và phân cụm như trong bảng 1.4 và các phương thức đánh giá mô hình trong bảng 1.5.

Bảng 1. : Các mô hình trong thư viện Sklearn

|  |  |
| --- | --- |
| **Tên mô hình** | **Mô tả** |
| LinearRegression | Mô hình hồi quy tuyến tính cơ bản |
| Ridge và Lasso Regression | Hồi quy tuyến tính với điều chuẩn L2 (Ridge) và L1 (Lasso). |
| LogisticRegression | Mô hình phân loại nhị phân và phân loại đa lớp |
| KNeighborsClassifier | Mô hình phân loại theo thuật toán KNN |
| RandomForestClassifier | Mô hình phân loại theo thuận toán Random Forest |
| KMeans | Phân cụm dữ liệu bằng K-means |
| DBSCAN | Phân cụm dữ liệu theo mật độ |

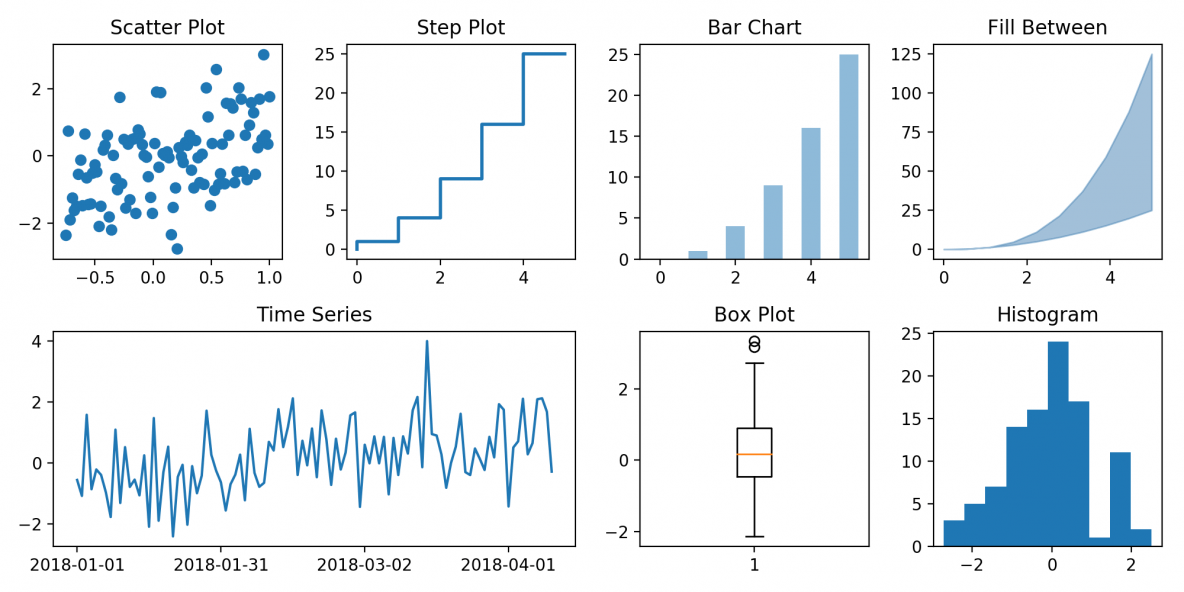
Bảng 1. : Các phương thức đánh giá mô hình trong Sklearn.

|  |  |
| --- | --- |
| **Tên phương thức** | **Mô tả** |
| Accuracy\_score | Đo độ chính xác của mô hình phân loại |
| Confusion\_matrix | Tạo ma trận hỗn loạn cho các bài toán phân loại |
| Classification\_report | Tạo báo cáo phân loại cho mô hình |
| Mean\_squared\_error | Đo lương sai số của bình phương trung bình |
| R2\_score | Đo lường chỉ số R2 score |

### Thư viện Matplotlib.

#### Giới thiệu.

Matplotlib là một thư viện Python nổi tiếng và mạnh mẽ, thường được sử dụng để trực quan hóa dữ liệu dưới dạng biểu đồ 2D. Được phát triển từ nhu cầu tạo các hình ảnh chất lượng cao cho dữ liệu khoa học và kinh doanh, Matplotlib đã trở thành một công cụ tiêu chuẩn cho các nhà phân tích và nhà khoa học dữ liệu. Matplotlib cho phép người dùng tạo ra nhiều loại biểu đồ khác nhau từ những biểu đồ đơn giản đến phức tạp, với khả năng tùy chỉnh cao. Thư viện này là mã nguồn mở và miễn phí, giúp bất kỳ ai cũng có thể sử dụng và đóng góp vào sự phát triển của nó.



Hình 1. : Một số dạng biểu đồ trong Matplotlib

Một số tính năng nổi bật của Matplotlib có thể kể đến như:

* Tạo biểu đồ 2D chất lượng cao: Matplotlib có khả năng tạo ra các biểu đồ chất lượng cao, phù hợp cho cả các bài báo khoa học và báo cáo thương mại.
* Hỗ trợ đa nền tảng: Matplotlib có thể chạy trên nhiều hệ điều hành khác nhau như Windows, macOS, và Linux, cũng như trên nhiều môi trường tương tác như Jupyter Notebook, IPython, và các ứng dụng GUI.
* Hỗ trợ nhiều định dạng đầu ra: Bạn có thể lưu biểu đồ dưới nhiều định dạng như PNG, PDF, SVG, JPG, và nhiều định dạng khác.
* Khả năng mở rộng: Matplotlib có thể được mở rộng bằng cách sử dụng các thư viện khác như NumPy và Pandas để xử lý dữ liệu, cũng như Seaborn để tạo các biểu đồ nâng cao hơn.
* Giao diện người dùng linh hoạt: Người dùng có thể chọn giữa nhiều giao diện để tương tác với thư viện, từ giao diện lập trình hướng đối tượng đến giao diện lệnh thủ tục (scripting).

Trực quan hóa dữ liệu là một kỹ năng thiết yếu đối với mọi nhà phân tích dữ liệu, và Matplotlib là một thư viện rất phổ biến để tạo các hình ảnh trực quan, giúp bạn truyền tải thông tin một cách hiệu quả. Tuy Matplotlib yêu cầu khá nhiều mã để tạo biểu đồ cơ bản nhưng bù lại, nó rất linh hoạt và có thể tùy chỉnh cao. Matplotlib được đánh giá cao bởi những đặc điểm sau đây:

* Dễ sử dụng: Matplotlib có cú pháp và giao diện tương tự MATLAB, cho phép người dùng chuyển đổi nhanh chóng và dễ dàng làm quen, đặc biệt hữu ích đối với những ai đã có kinh nghiệm với MATLAB.
* Đa dạng loại biểu đồ: Matplotlib hỗ trợ hầu hết các loại biểu đồ phổ biến như biểu đồ đường, cột, tròn, phân tán, biểu đồ hộp (box plot), histogram, và nhiều loại biểu đồ khác. Điều này mang đến sự linh hoạt trong việc thể hiện dữ liệu dưới nhiều hình thức.
* Tùy biến cao: Thư viện cho phép người dùng tùy chỉnh gần như tất cả các yếu tố của biểu đồ, từ màu sắc, kích thước, kiểu đường nét, nhãn đến các yếu tố phức tạp hơn như giới hạn trục, lưới, chú thích, và văn bản. Điều này giúp Matplotlib trở thành công cụ lý tưởng khi cần tạo ra các biểu đồ chuyên nghiệp và có thẩm mỹ cao.
* Khả năng mở rộng và cộng đồng lớn: Cộng đồng người dùng và nhà phát triển Matplotlib rất lớn. Bên cạnh tài liệu phong phú, còn có nhiều thư viện mở rộng khác như Seaborn, được phát triển trên nền Matplotlib để giúp việc tạo biểu đồ trở nên nhanh chóng và dễ dàng hơn, phù hợp với các phân tích dữ liệu khám phá.

Matplotlib thường được so sánh với GNUplot và MATLAB, hai công cụ trực quan hóa dữ liệu phổ biến khác. Tuy nhiên, Matplotlib có một số ưu điểm nổi bật:

* Với GNUplot: Khác với GNUplot, Matplotlib là một mô-đun Python, giúp nó tận dụng được sức mạnh và sự phổ biến ngày càng gia tăng của Python. Nhờ đó, Matplotlib dễ dàng tích hợp với các thư viện khác của Python như Numpy và Scipy.

Với MATLAB: Matplotlib được xem như một lựa chọn thay thế tuyệt vời cho MATLAB khi kết hợp với Numpy và Scipy. Trong khi MATLAB là một công cụ thương mại và đắt đỏ, Matplotlib là mã nguồn mở và miễn phí. Ngoài ra, đối với những người quen thuộc với MATLAB, Matplotlib cung cấp một thủ tục "pylab" có giao diện và cú pháp tương tự MATLAB, giúp người dùng chuyển đổi dễ dàng.

#### Cách cài đặt.

Để cài đặt thư viện Numpy trong Python, người dùng có thể dùng PIP. Chú ý, hãy đảm bảo Python ở phiên bản 3.7 trở lên và một số thư viện bắt buộc như Numpy. Sau đây là các bước cài đặt thư viện này:

Bước 1: Mở terminal hoặc command prompt.

Bước 2: Chạy lệnh cài đặt Matplotlib:

*pip install matplotlib*

Bước 3: Xác nhận cài đặt. Mở Python hoặc Jupyter Notebook và chạy lệnh sau để kiểm tra phiên bản Matplotlib:

*import matplotlib*

*print(matplotlib.\_\_version\_\_)*

Nếu không có lỗi và phiên bản hiện ra, người dùng đã cài đặt thành công Matplotlib

#### Các phương thức chính.

Matplotlib có nhiều phương thức hữu ích, nhưng dưới đây là những phương thức chính thường được sử dụng nhất trong việc tạo biểu đồ:

1. Phương thức pyplot.

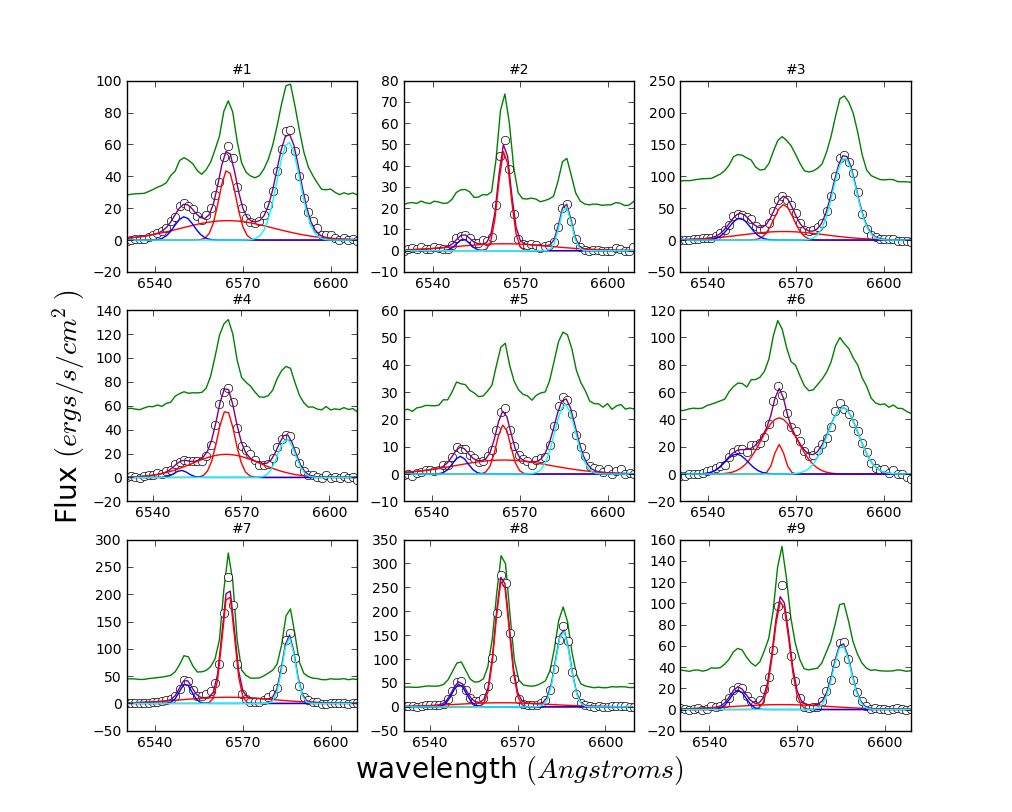
Thư viện pyplot trong Matplotlib là thành phần cốt lõi giúp tạo và tùy chỉnh các biểu đồ. Thông qua pyplot, người dùng có thể tạo ra nhiều loại biểu đồ khác nhau như biểu đồ đường, biểu đồ cột, biểu đồ phân tán, v.v. Một số hàm quan trọng trong pyplot bao gồm:

* plot(): Hàm cơ bản để vẽ biểu đồ đường. Người dùng có thể truyền vào các tham số như danh sách hoặc mảng để tạo ra biểu đồ đường nối các điểm.
* scatter(): Hàm này dùng để tạo biểu đồ phân tán, hiển thị mối quan hệ giữa hai tập dữ liệu.
* bar(): Được sử dụng để tạo biểu đồ cột, hiển thị dữ liệu theo các cột đứng.
* hist(): Tạo biểu đồ histogram, biểu diễn phân phối tần suất của dữ liệu.
* pie(): Dùng để tạo biểu đồ tròn, biểu diễn tỷ lệ giữa các phần trong một tổng thể.

1. Phương thức figure và sublot.

Khi làm việc với các biểu đồ phức tạp hoặc khi cần vẽ nhiều biểu đồ trong cùng một hình, phương thức figure và subplot đóng vai trò rất quan trọng.

* figure(): Tạo một vùng vẽ (canvas) mới. Điều này rất hữu ích khi người dùng muốn vẽ nhiều biểu đồ khác nhau mà không ảnh hưởng lẫn nhau.
* subplot(): Cho phép chia vùng vẽ thành nhiều phần nhỏ hơn. Người dùng có thể chọn số hàng và số cột để tạo các ô con (subplots), qua đó dễ dàng hiển thị nhiều biểu đồ khác nhau trên cùng một hình.



Hình 1. : Ví dụ minh họa về 9 subplot trên một khung hình.

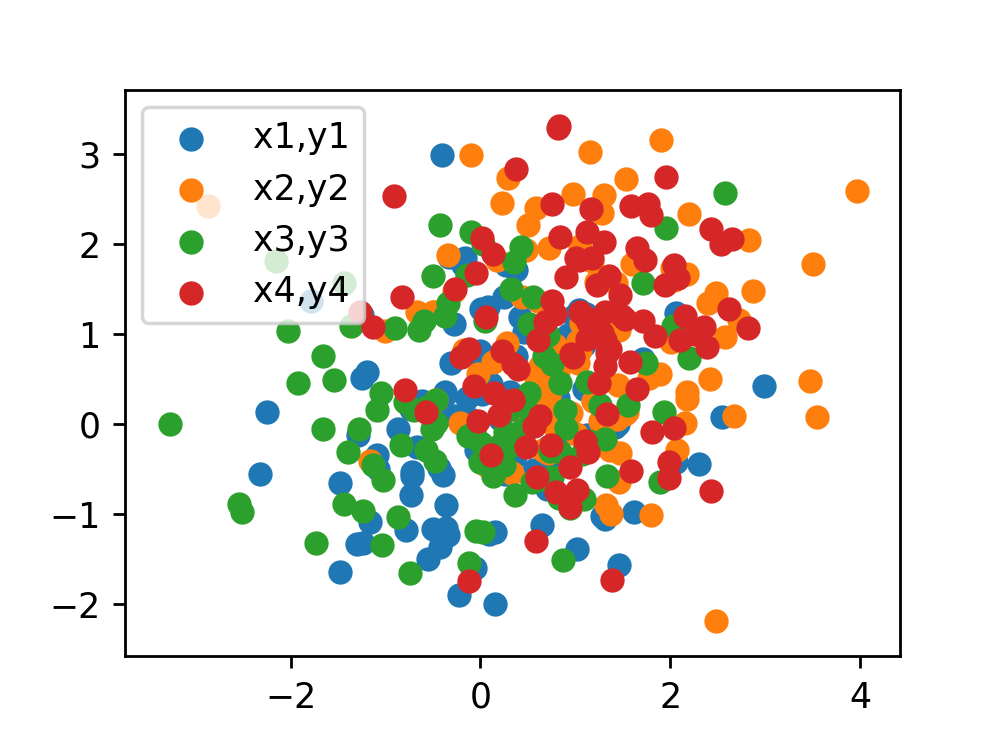
1. Phương thức axes.

Phương thức axes trong Matplotlib giúp tùy chỉnh các thành phần trên trục tọa độ của biểu đồ. Với axes, người dùng có thể chỉnh sửa các thông tin như nhãn trục, phạm vi dữ liệu, tiêu đề biểu đồ, v.v.

* set\_xlabel() và set\_ylabel(): Đặt tên cho các trục x và y của biểu đồ.
* set\_xlim() và set\_ylim(): Đặt phạm vi hiển thị cho trục x và trục y.
* set\_title(): Đặt tiêu đề cho biểu đồ, giúp người xem dễ dàng nhận biết nội dung biểu đồ

1. Phương thức legend.

Khi biểu diễn nhiều tập dữ liệu trên cùng một biểu đồ, phương thức legend() giúp hiển thị chú thích, giải thích ý nghĩa của từng tập dữ liệu. Điều này giúp người xem hiểu rõ hơn về thông tin được biểu diễn trong biểu đồ.



Hình 1. : Minh họa về legend trong Matplotlib.

Trong đó các điểm dữ liệu được chú thích bằng màu sắc khác nhau.

Ngoài các phương thức trên, Matplotlib còn cung cấp một số công cụ bổ sung giúp người dùng dễ dàng tùy chỉnh biểu đồ:

* grid(): Hiển thị lưới trên biểu đồ, hỗ trợ người xem theo dõi các giá trị.
* annotate(): Dùng để chú thích lên biểu đồ, giúp làm nổi bật các điểm dữ liệu quan trọng.
* savefig(): Lưu biểu đồ dưới dạng file ảnh, hỗ trợ các định dạng như PNG, PDF, SVG, v.v.

### Tổng kết.

Python là một ngôn ngữ lập trình mạnh mẽ và linh hoạt, phần lớn nhờ vào hệ sinh thái phong phú của các thư viện hỗ trợ nhiều lĩnh vực khác nhau. Các thư viện như numpy, random, math, sklearn, pygame, scipy, và matplotlib cung cấp các công cụ và phương thức cần thiết để xử lý dữ liệu, thực hiện các phép toán, xây dựng mô hình học máy, lập trình trò chơi, và tạo biểu đồ trực quan.

* NumPy hỗ trợ tính toán mảng và xử lý dữ liệu số lớn, là nền tảng cho nhiều thư viện khác.
* Random cung cấp các hàm tạo số ngẫu nhiên, hỗ trợ trong việc tạo mẫu và mô phỏng.
* Math giúp thực hiện các phép toán cơ bản và mở rộng, tối ưu cho các tính toán số học.
* Scikit-Learn (sklearn) là một thư viện học máy mạnh mẽ, cung cấp các công cụ xây dựng và đánh giá mô hình, phục vụ cho công việc phân tích và khai phá dữ liệu.
* Pygame hỗ trợ phát triển trò chơi và ứng dụng đa phương tiện, tạo trải nghiệm tương tác trên giao diện đồ họa.
* Matplotlib cho phép tạo ra các biểu đồ trực quan, giúp truyền tải và giải thích dữ liệu hiệu quả.

Những thư viện này kết hợp tạo thành một hệ sinh thái hoàn chỉnh, giúp các nhà phát triển, nhà nghiên cứu và nhà khoa học dữ liệu dễ dàng triển khai các giải pháp, từ các phép toán đơn giản đến các mô hình phân tích phức tạp. Sự linh hoạt và tính sẵn có của các thư viện này làm cho Python trở thành lựa chọn hàng đầu trong các lĩnh vực khoa học và công nghệ, mang lại hiệu quả và khả năng tối ưu hóa cao trong công việc.

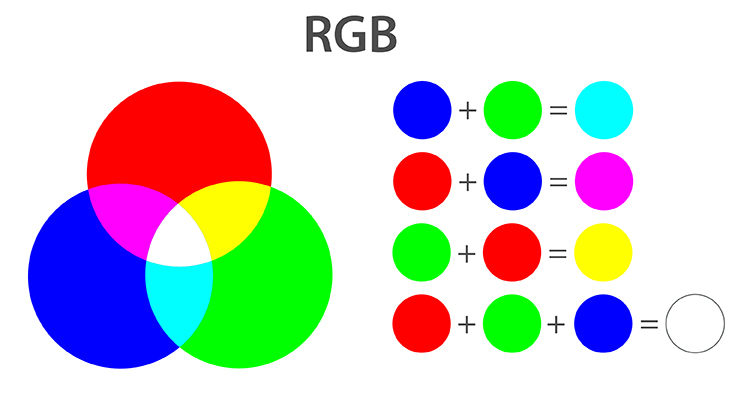
## Hệ màu.

### Giới thiệu về hệ màu

Hệ màu (Color Model) là một phương pháp biểu diễn và mô tả màu sắc theo cách có thể nhận diện và tái tạo lại trên các thiết bị khác nhau như màn hình, máy in, máy ảnh, v.v. Hệ màu quy định cách mã hóa màu sắc dựa trên một tập hợp các giá trị số nhằm định nghĩa các màu khác nhau trong không gian màu (Color Space). Hệ màu giúp chuẩn hóa màu sắc, đảm bảo tính nhất quán và chính xác khi hiển thị trên các thiết bị khác nhau.

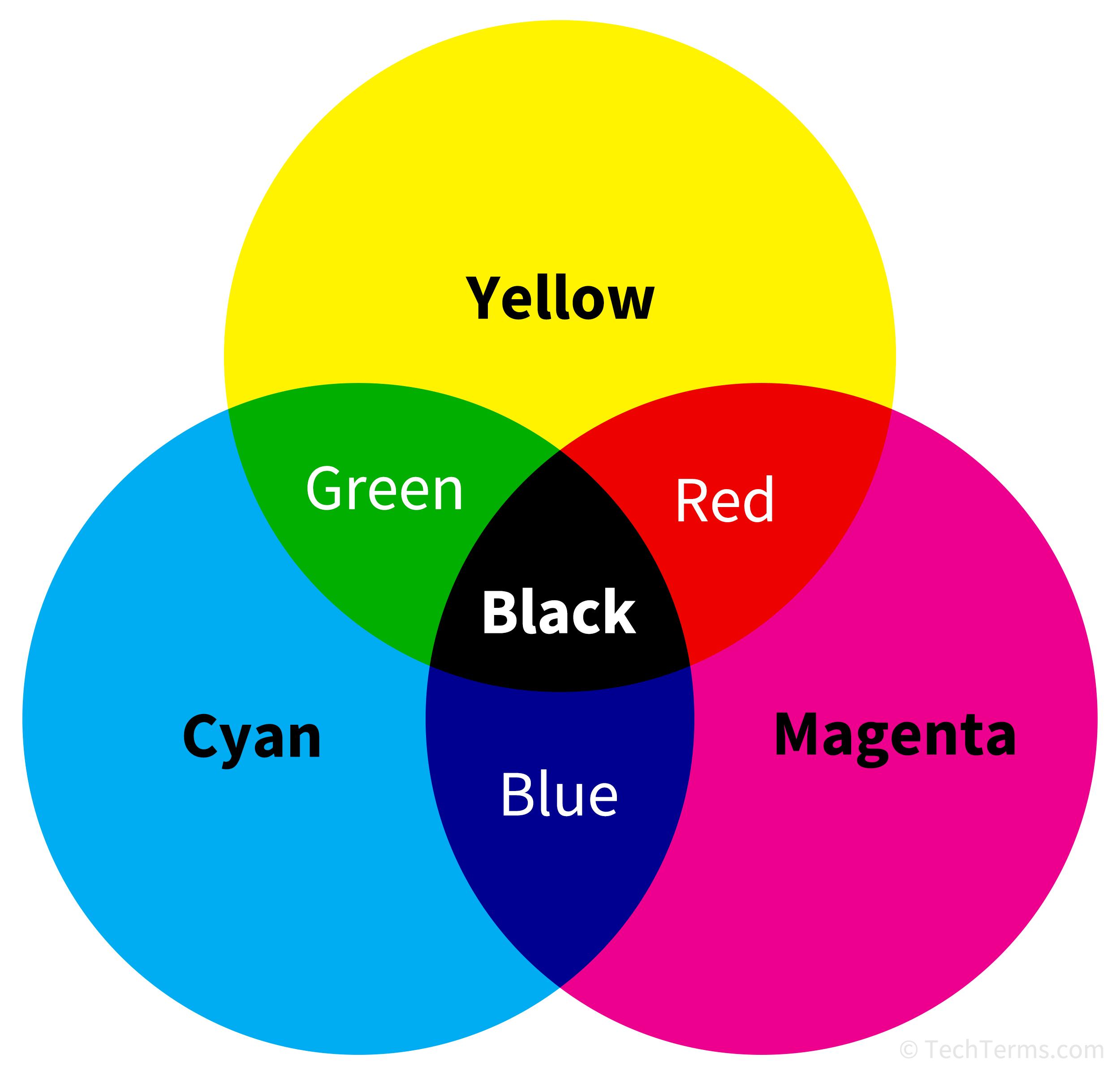
Có nhiều hệ màu khác nhau, mỗi hệ màu được thiết kế phù hợp cho một số ứng dụng cụ thể như hiển thị trên màn hình, in ấn hoặc thiết kế đồ họa. Các hệ màu phổ biến bao gồm:

* 1. Hệ màu RGB (Red – Green – Blue)
     + Mô tả: Hệ màu RGB dựa trên ba màu cơ bản là đỏ (Red), xanh lá cây (Green) và xanh dương (Blue). Đây là hệ màu được sử dụng phổ biến trong các thiết bị điện tử như màn hình máy tính, tivi và điện thoại di động.



Hình 1. : Hệ màu RGB

* + - Cách hoạt động: Mỗi màu được biểu diễn dưới dạng sự kết hợp của ba màu cơ bản với các mức độ khác nhau, thường từ 0 đến 255. Ví dụ, màu trắng là (255, 255, 255) và màu đen là (0, 0, 0).
    - Ứng dụng: Thường dùng cho các thiết bị phát sáng như màn hình và hệ thống hiển thị số.
  1. Hệ màu CMYK (Cyan – Magenta – Yellow – Black)
     + - * Mô tả: Hệ màu CMYK dựa trên bốn màu là xanh lơ (Cyan), hồng cánh sen (Magenta), vàng (Yellow) và đen (Black). Đây là hệ màu được sử dụng chủ yếu trong in ấn.
         * Cách hoạt động: Màu sắc được tạo ra bằng cách kết hợp các mức độ khác nhau của bốn màu cơ bản này. Hệ màu CMYK là hệ màu trừ (subtractive color model), nghĩa là màu càng nhiều sẽ càng tối.



Hình 1. : Hệ màu CMYK

* + - * + Ứng dụng: Được sử dụng trong công nghiệp in ấn, nơi màu sắc được tạo ra bằng cách trộn mực trên giấy
  1. Hệ màu HSL/HSV
* Mô tả: Hệ màu HSL (Hue, Saturation, Lightness) và HSV (Hue, Saturation, Value) là các hệ màu khác biệt dựa trên cách nhìn tự nhiên của mắt người về màu sắc.
  + Hue (Sắc độ): Xác định màu chính của màu sắc (đỏ, vàng, xanh, v.v.).
  + Saturation (Độ bão hòa): Mức độ tinh khiết của màu, từ nhạt đến đậm.
  + Lightness/Value (Độ sáng): Mức độ sáng tối của màu sắc.
* Ứng dụng: Thường sử dụng trong thiết kế đồ họa và chỉnh sửa ảnh, giúp dễ dàng điều chỉnh và lựa chọn màu sắc theo cách trực quan hơn.
  1. Hệ màu Lab (CIELAB)
* Mô tả: Hệ màu Lab được phát triển bởi Ủy ban Quốc tế về Chiếu sáng (CIE) để chuẩn hóa màu sắc cho các mục đích chuyên nghiệp.
  + L biểu thị độ sáng của màu (Lightness).
  + a và b biểu thị màu sắc: a cho màu từ xanh lá cây đến đỏ, và b cho màu từ xanh lam đến vàng.
* Ứng dụng: Được sử dụng trong các lĩnh vực yêu cầu độ chính xác cao về màu sắc như in ấn chất lượng cao, xử lý ảnh và chỉnh sửa màu trong nhiếp ảnh

Mỗi hệ màu có ưu điểm riêng và phù hợp với các mục đích khác nhau, từ hiển thị trên màn hình (RGB), in ấn (CMYK), đến chỉnh sửa ảnh chuyên nghiệp (HSL/HSV, Lab). Việc chọn đúng hệ màu là quan trọng để đảm bảo rằng màu sắc hiển thị chính xác, phù hợp với yêu cầu và đạt được hiệu quả mong muốn trong các ứng dụng cụ thể.

### Thư viện hệ màu trong Python.

Trong Python, có một số thư viện cung cấp các công cụ để làm việc với hệ màu, cho phép bạn chuyển đổi giữa các không gian màu khác nhau, điều chỉnh màu sắc, và tạo ra các bảng màu tùy chỉnh. Dưới đây là một số thư viện phổ biến để làm việc với hệ màu trong Python:

1. Matplotlib

Matplotlib là thư viện phổ biến dùng để tạo biểu đồ và hiển thị màu sắc trong dữ liệu. Thư viện này hỗ trợ rất nhiều công cụ làm việc với màu sắc và cho phép chuyển đổi giữa các hệ màu. Các tính năng màu sắc trong Matplotlib bao gồm:

* Mã màu hex, RGB: Có thể dùng chuỗi hex (#RRGGBB) hoặc tuple RGB để chỉ định màu sắc.
* Tùy chỉnh bảng màu: Matplotlib cung cấp các bảng màu tích hợp như viridis, plasma, inferno, magma, cividis, v.v.
* Hệ màu HSL và HSV: Hỗ trợ chuyển đổi giữa RGB và HSL/HSV, sử dụng colorsys hoặc matplotlib.colors.

1. Pillow

Pillow là một thư viện xử lý ảnh mạnh mẽ trong Python, hỗ trợ các định dạng màu như RGB, RGBA, CMYK, và LAB. Pillow cho phép tạo và chỉnh sửa ảnh màu, chuyển đổi giữa các không gian màu, và thay đổi độ bão hòa hay độ sáng

1. OpenCV

OpenCV là thư viện xử lý ảnh và thị giác máy tính mạnh mẽ, hỗ trợ nhiều hệ màu khác nhau như RGB, BGR, HSV, HLS, YCrCb, LAB, v.v. OpenCV rất hữu ích cho các ứng dụng xử lý ảnh phức tạp, bao gồm cả phân đoạn màu và điều chỉnh màu sắc

1. ColorSys

Colorsys là thư viện tiêu chuẩn của Python cung cấp các công cụ chuyển đổi giữa RGB và các hệ màu khác như HSV, HLS và YIQ. colorsys là một thư viện đơn giản và dễ dùng nếu bạn chỉ cần chuyển đổi giữa các không gian màu

1. Colour-science

Colour-science là một thư viện chuyên sâu về màu sắc, hỗ trợ chuyển đổi giữa nhiều không gian màu phức tạp, từ RGB đến XYZ, LAB và LMS. Thư viện này chủ yếu được sử dụng trong các lĩnh vực yêu cầu độ chính xác cao về màu sắc như xử lý ảnh chuyên nghiệp, đồ họa và thiết kế

1. Scikit-image

Skimage.color là một module trong thư viện scikit-image giúp chuyển đổi giữa nhiều hệ màu như RGB, HSV, LAB, XYZ. Đây là công cụ hữu ích khi bạn làm việc với các tác vụ xử lý ảnh trong học máy và thị giác máy tính

Các thư viện này cung cấp nhiều công cụ để xử lý màu sắc trong Python. Tùy vào ứng dụng cụ thể, bạn có thể lựa chọn thư viện phù hợp, từ Matplotlib và Pillow cho các tác vụ cơ bản đến OpenCV, skimage, và colour-science cho các ứng dụng phức tạp hơn trong lĩnh vực xử lý ảnh và thị giác máy tính.

## Lưu trữ dữ liệu và hình ảnh.

### Lưu trữ ảnh với vector dữ liệu

Ảnh kỹ thuật số có thể được lưu trữ dưới dạng ma trận số, trong đó mỗi điểm ảnh (pixel) có giá trị biểu diễn độ sáng hoặc màu sắc. Đối với ảnh màu, mỗi pixel thường được biểu diễn bằng một vector gồm ba giá trị RGB (đỏ, xanh lục, xanh dương) cho biết cường độ của từng màu.

Việc lưu trữ ảnh dưới dạng vector dữ liệu cho phép ứng dụng các kỹ thuật tính toán và phân tích. Các hình ảnh khi được chuyển thành ma trận số có thể được sử dụng để thực hiện các thao tác như biến đổi ảnh, lọc nhiễu, hoặc nhận diện mẫu. Vector hóa ảnh là một cách tiếp cận rất quan trọng trong các hệ thống nhận dạng, thị giác máy tính và học sâu, giúp các hệ thống này phân tích và nhận diện thông tin trực quan.

Một số phương thức chính trong Python hỗ trợ lưu trữ ảnh với vector dữ liệu như sau:

Bảng 1. : Các phương thức chính trong lưu trữ và xử lí dữ liệu với vector

|  |  |
| --- | --- |
| **Phương thức/Hàm** | **Chức năng** |
| Image.open(path) | Mở một tệp ảnh từ đường dẫn path và trả về đối tượng ảnh. |
| Image.save(path) | Lưu ảnh hiện tại vào tệp với đường dẫn path. |
| Image.convert(mode) | Chuyển đổi ảnh sang chế độ khác (ví dụ: từ màu sang ảnh xám). |
| Image.resize(size) | Thay đổi kích thước ảnh về kích thước mới size. |
| Image.crop(box) | Cắt một phần của ảnh theo hộp box xác định. |
| Image.rotate(angle) | Xoay ảnh một góc angle (đơn vị là độ). |
| Image.transpose(method) | Thực hiện phép biến đổi trên ảnh (ví dụ: lật dọc, lật ngang, xoay 90 độ). |
| np.array(image) | Chuyển đổi đối tượng ảnh thành mảng NumPy để xử lý như một vector dữ liệu. |
| np.mean(image\_array, axis) | Tính giá trị trung bình của các pixel theo trục đã chỉ định, có thể dùng để chuyển ảnh màu sang ảnh xám. |
| np.dstack((r, g, b)) | Kết hợp ba kênh màu (R, G, B) thành một ảnh màu từ ba mảng. |
| cv2.imread(path) | Đọc ảnh từ tệp và trả về mảng NumPy chứa dữ liệu pixel của ảnh. |
| cv2.imwrite(path, image\_array) | Lưu mảng NumPy dưới dạng ảnh vào tệp tại đường dẫn path. |
| cv2.cvtColor(image\_array, code) | Chuyển đổi màu của ảnh từ một không gian màu này sang không gian màu khác (ví dụ: từ BGR sang RGB). |
| cv2.resize(image\_array, dsize) | Thay đổi kích thước của mảng ảnh image\_array về kích thước mới dsize. |
| cv2.threshold(image\_array, thresh, maxval, type) | Thực hiện ngưỡng hóa (thresholding) trên ảnh để phân tách foreground và background. |
| cv2.GaussianBlur(image\_array, ksize, sigmaX) | Làm mờ ảnh bằng bộ lọc Gaussian với kích thước kernel ksize và độ lệch chuẩn sigmaX. |

### Kết hợp với danh sách Numpy

NumPy là một thư viện nổi tiếng trong Python, được thiết kế đặc biệt để làm việc với mảng và các phép toán số học. Khác với danh sách, mảng NumPy tiết kiệm bộ nhớ và hỗ trợ thực hiện các phép toán trên nhiều phần tử đồng thời, nhờ đó nâng cao hiệu suất tính toán, đặc biệt khi xử lý dữ liệu lớn.

So với danh sách, mảng NumPy có tính hiệu quả cao hơn vì dữ liệu được lưu trữ liên tục trong bộ nhớ. Thư viện này cung cấp các phương thức hỗ trợ các phép toán vector và ma trận, giúp thực hiện các tính toán thống kê, đại số tuyến tính và các phân tích dữ liệu phức tạp. NumPy là một công cụ không thể thiếu trong các ứng dụng về xử lý ảnh, khoa học.

### Tổng kết

Trong lĩnh vực lập trình Python, việc lưu trữ dữ liệu và hình ảnh đóng vai trò quan trọng trong nhiều ứng dụng khác nhau. Trước hết, việc sử dụng danh sách trong Python cho phép lưu trữ các vector, giúp thực hiện các phép toán và thao tác một cách linh hoạt và dễ dàng. Điều này mang lại sự tiện lợi trong việc xử lý và quản lý dữ liệu.

Ngoài ra, ảnh có thể được chuyển đổi thành danh sách pixel hoặc mảng NumPy, tạo điều kiện cho việc chỉnh sửa và lưu lại ảnh một cách hiệu quả. Quá trình chuyển đổi này mở ra nhiều khả năng ứng dụng trong lĩnh vực xử lý ảnh và thị giác máy tính.

Cuối cùng, thư viện NumPy nổi bật như một công cụ mạnh mẽ để làm việc với các mảng và vector lớn trong Python. Sự tối ưu hóa hiệu suất mà NumPy cung cấp cho các tác vụ tính toán và xử lý dữ liệu là vô cùng quan trọng, đặc biệt trong các lĩnh vực yêu cầu khả năng xử lý dữ liệu lớn và phức tạp. Tóm lại, sự kết hợp giữa các phương pháp lưu trữ dữ liệu, lưu trữ ảnh và việc sử dụng thư viện NumPy mang lại nhiều lợi ích cho việc phát triển các ứng dụng thực tiễn và nghiên cứu khoa học

# CHƯƠNG II. THUẬT TOÁN K-MEANS

## 2.1. Tổng quan về Machine Learning

2.1.1.  Lịch sử hình thành và phát triển.

Nguồn gốc của học máy có thể bắt nguồn từ những năm 1940 , khi các nhà nghiên cứu bắt đầu khám phá những bài toán nhận dạng mẫu cơ bản đầu tiên và nghiên cứu những mạng lưới thần kinh. Giai đoạn đầu của lịch sử Machine Learning - ML này được đánh dấu bằng những ý tưởng đột phá và những nỗ lực không ngừng nghỉ trong việc tạo ra những chiếc máy tính có thể bắt chước quá trình suy nghĩ của con người. Năm 1943, Walter Pitts và Warren McCulloch đã nghĩ ra mô hình toán học đầu tiên của mạng lưới thần kinh nhân tạo, đặt nền móng cho mạng lưới thần kinh hiện đại và phát triển các bộ công cụ học máy phân tán McCulloch & Pitts, 1943).

Những người tiên phong trong thời kỳ mở đầu của ML có thể kể đến như Donald Hebb , Alan Turing và Arthur Samuel , mặc dù họ không phải là người khởi xướng duy nhất, tuy nhiên những nghiên cứu và công sự của họ đã đóng góp đáng kể vào sự phát triển của ngành học máy . Công trình của Hebb về sự giao tiếp trong hệ thống thần kinh, bài kiểm tra trí tuệ nhân tạo của Turing và việc Samuel đặt ra thuật ngữ “học máy” đều góp phần vào lĩnh vực trí tuệ nhân tạo (AI) đang phát triển và đặt nền móng cho vô số thuật toán học máy mà chúng ta biết ngày nay (Hebb, 1949; Turing, 1950; Samuel, 1959).

Hình 2. 3: Đại kiện tướng Ke Jie thua bẽ bàng 2 trận liên tiếp trước trí tuệ nhân tạo của Google.

Một vài sự kiện tiêu biểu trong quá trình hình thành và phát triển của học máy có thể kể đến những mốc thời gian sau:

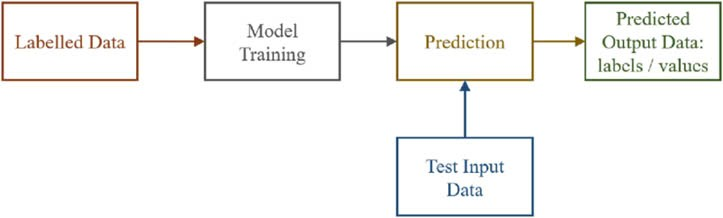
* Năm 1943, Walter Pitts và Warren McCulloch phát triển mô hình học máy đầu tiên để giải quyết thử thách do John von Neumann đặt ra:làm thế nào để các máy tính có thể giao tiếp với nhau.
* Năm 1949, Donald Hebb đưa ra khái niệm về giao tiếp giữa các nơ-ron trong hệ thống thần kinh.
* Năm 1950, Alan Turing giới thiệu bài kiểm tra Turing đánh giấu cột mốc quan trọng trong lĩnh vực AI.
* Năm 1951, Marvin Lee Minsky phát minh ra máy tính tăng cường tương tự thần kinh (SNARC).
* Năm 1967, Cover cùng các cộng sự của mình tạo ra thuật toán kNN.
* Năm 1980, phát hiện ra neocognitron – một mạng lưới thần kinh nhân tạo nhiều lớp, là tiền đề cho mạng tích chập CNN sau này (Fukushima, 1980).
* Năm 1997, Deep Blue của IBM gây sốc cả thế giới khi đánh bại nhà vô địch cờ vua thời bấy giờ.
* Năm 2006, thuật ngữ “Deep Learning” được Geoffrey Hinton đặt ra để mô tả các thuật toán mới cho phép máy tính “nhìn” và phân biệt các đối tượng cững như văn bản trong hình ảnh, video.
* Năm 2017, Google đã công bố nghiên cứu đầu tiên về kiến trúc deep learning mang tên Transformers (Vaswani et al., 2017).
* Năm 2023, OpenAI phát hành ChatGPT.

Từ những ngày đầu nhận dạng mẫu đơn giản cho đến các mô hình học tập phức tạp ngày nay, lịch sử của học máy (ML) đã là một **hành trình hấp dẫn** . Đó là câu chuyện về con người nỗ lực tạo ra những chiếc máy tính có thể học hỏi, thích ứng và đưa ra những quyết định sáng suốt, giống như quá trình nhận thức của chính chúng ta. Chuyến đi này đã định hình lại các ngành công nghiệp, xác định lại sự tương tác giữa con người và máy tính và mở ra một thế giới tiềm năng chưa được khai thác.

2.1.2.  Phân loại.

2.1.2.1. Học có giám sát.

Học có giám sát (Supervised Learning) là một phương pháp quan trọng trong lĩnh vực Machine Learning, trong đó mô hình được huấn luyện trên một tập dữ liệu đã được gán nhãn. Đây là quá trình huấn luyện một mô hình ML bằng cách sử dụng dữ liệu đầu vào (input) và đầu ra mong muốn tương ứng (output) đã biết trước, nhằm mục đích tạo ra một mô hình có khả năng dự đoán chính xác output cho các input mới mà nó chưa từng thấy trước đó. Trong học có giám sát, tập dữ liệu huấn luyện bao gồm các cặp (input, output) đã được gán nhãn, trong đó input là các đặc trưng của dữ liệu, còn output là nhãn hoặc giá trị mục tiêu. Mô hình được huấn luyện để tìm ra mối quan hệ giữa input và output, điều chỉnh các tham số nội bộ để dự đoán output càng chính xác càng tốt. Học có giám sát thường được áp dụng cho hai loại bài toán chính: phân loại (classification), dự đoán một nhãn rời rạc, và hồi quy (regression), dự đoán một giá trị liên tục. Ví dụ điển hình của học có giám sát bao gồm nhận dạng chữ viết tay và dự đoán giá nhà. Phương pháp này có ưu điểm là hiệu suất cao khi có đủ dữ liệu được gán nhãn và dễ đánh giá hiệu quả của mô hình.



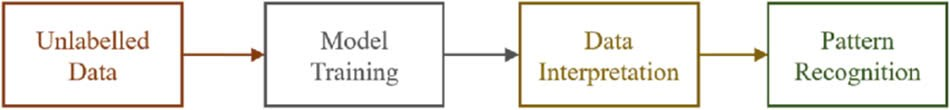
Hình 2.4: Sơ đồ tổng quát của học máy có giám sát.

Tuy nhiên, nó cũng có nhược điểm là cần một lượng lớn dữ liệu đã được gán nhãn, có thể tốn kém và mất thời gian, đồng thời có thể gặp vấn đề overfitting nếu mô hình quá phức tạp hoặc dữ liệu huấn luyện không đại diện. Các thuật toán phổ biến trong học có giám sát bao gồm hồi quy tuyến tính và logistic, cây quyết định và Random Forest, Support Vector Machines (SVM), và Neural Networks. Học có giám sát là một trong những phương pháp cơ bản và quan trọng nhất trong Machine Learning, được ứng dụng rộng rãi trong nhiều lĩnh vực như xử lý ngôn ngữ tự nhiên, thị giác máy tính, và phân tích dự đoán.

2.5.2.2. Học không giám sát.

Học không giám sát (Unsupervised Learning) là một phương pháp quan trọng trong Machine Learning, trong đó mô hình được huấn luyện trên dữ liệu không được gán nhãn. Khác với học có giám sát, phương pháp này không có sẵn thông tin về kết quả mong muốn, mà thay vào đó, mục tiêu là tìm ra cấu trúc hoặc mẫu ẩn trong dữ liệu. Học không giám sát tập trung vào việc khám phá và mô tả các đặc điểm thú vị của dữ liệu mà không cần hướng dẫn cụ thể.

Trong học không giám sát, mô hình phải tự tìm ra các mối quan hệ và cấu trúc trong dữ liệu. Điều này có thể bao gồm việc nhóm các điểm dữ liệu tương tự nhau (clustering), giảm số chiều của dữ liệu để tìm ra các đặc trưng quan trọng nhất (dimensionality reduction), hoặc phát hiện các mẫu bất thường (anomaly detection). Ví dụ phổ biến của học không giám sát bao gồm phân khúc khách hàng trong tiếp thị, phát hiện gian lận trong giao dịch tài chính, và tổ chức các bài báo tin tức theo chủ đề.



Hình 2.5: Sơ đồ tổng quát học máy không giám sát.

Các thuật toán phổ biến trong học không giám sát bao gồm K-means clustering, Hierarchical clustering, Principal Component Analysis (PCA), và các mô hình mạng neural tự mã hóa (autoencoders). Ưu điểm chính của học không giám sát là khả năng xử lý dữ liệu chưa được gán nhãn, cho phép khám phá các insight mới và không được biết trước từ dữ liệu. Nó cũng có thể giúp tiết kiệm thời gian và chi phí liên quan đến việc gán nhãn dữ liệu.

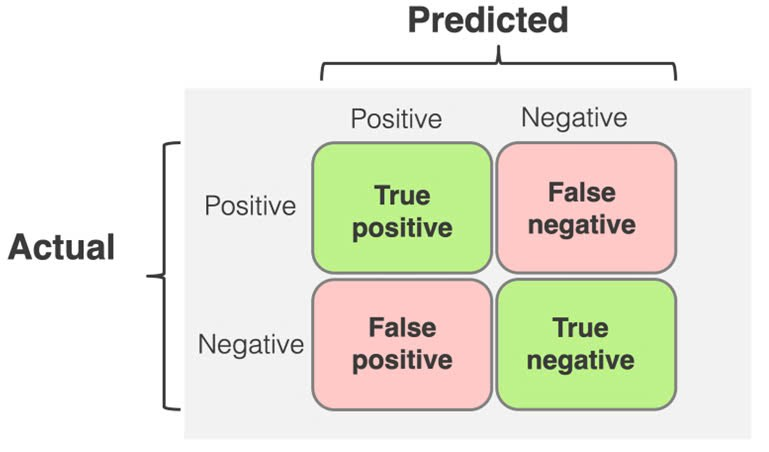
Tuy nhiên, học không giám sát cũng có những thách thức riêng. Việc đánh giá hiệu quả của mô hình có thể khó khăn hơn so với học có giám sát, vì không có tiêu chuẩn "đúng" rõ ràng để so sánh. Ngoài ra, kết quả của học không giám sát có thể khó giải thích hoặc ít có ý nghĩa thực tế nếu không được diễn giải cẩn thận.

Mặc dù có những thách thức, học không giám sát vẫn là một công cụ mạnh mẽ trong kho vũ khí của các nhà khoa học dữ liệu và chuyên gia ML. Nó cho phép khám phá những hiểu biết mới từ dữ liệu, tìm ra các mẫu ẩn, và có thể được sử dụng như một bước tiền xử lý cho các tác vụ học có giám sát phức tạp hơn. Khi kết hợp với các phương pháp khác, học không giám sát đóng vai trò quan trọng trong việc khai thác giá trị từ khối lượng dữ liệu ngày càng tăng trong thế giới hiện đại.

2.1.3.  Các phương pháp đánh giá.

Mô hình học máy thường là các mô hình dự đoán, vì vậy mô hình không có thể dự đoán chính xác 100% những gì chúng ta cung cấp. Với các bài toán như phân loại, phân lớp hay nhận dạng; các phương pháp đánh giá xem một mô hình có hiệu năng cao hay thấp thường được đánh giá thông qua confusion matrix và accuracy.

Confusion Matrix (Ma trận nhầm lẫn) là một bảng thể hiện hiệu suất của mô hình phân loại bằng cách so sánh các dự đoán với giá trị thực tế. Trong trường hợp phân loại nhị phân, ma trận này gồm bốn ô: True Positives (TP), True Negatives (TN), False Positives (FP), và False Negatives (FN). TP và TN là số lượng dự đoán đúng, trong khi FP và FN là số lượng dự đoán sai. Ma trận này cung cấp cái nhìn tổng quan về các loại lỗi mà mô hình mắc phải, giúp hiểu rõ hơn về hiệu suất của mô hình.



Hình 3. 1: Confusion matrix trong mô hình học máy.

Accuracy (Độ chính xác) là tỷ lệ các dự đoán đúng trên tổng số dự đoán. Nó được tính bằng công thức:

Accuracy cung cấp một số đo tổng quát về hiệu suất của mô hình, nhưng có thể gây hiểu nhầm trong trường hợp dữ liệu không cân bằng. Ví dụ, nếu 95% dữ liệu thuộc một lớp, một mô hình luôn dự đoán lớp đó sẽ có accuracy 95%, nhưng không hữu ích trong thực tế.

Mỗi phương pháp này có ưu điểm và hạn chế riêng. Confusion matrix cung cấp thông tin chi tiết nhất nhưng có thể khó diễn giải với dữ liệu lớn. Accuracy dễ hiểu nhưng có thể gây hiểu nhầm với dữ liệu không cân bằng. Trong thực tế, thường sử dụng kết hợp nhiều phương pháp để đánh giá toàn diện hiệu suất của mô hình phân loại.

2.1.4.  Những đột phá và ứng dụng.

Trong những năm gần đây, học máy đã trải qua một loạt đột phá và đổi mới, tạo nên một cuộc cách mạng trong lĩnh vực khoa học và công nghệ. Một số tiến bộ đáng chú ý bao gồm.

* Cuộc cách mạng deep learning vào năm 2012.
* Sự phát triển của các thuật toán học tăng cường, điển hình là AlphaGo của DeepMind.
* Những tiến bộ trong xử lý ngôn ngữ tự nhiên, một trong số đó là Chat GPT của OpenAI.

Những tiến bộ này đã cải thiện đáng kể khả năng và ứng dụng của trí tuệ nhân tạo. Họ cũng cho thấy tiềm năng của máy móc trong việc hiểu và tạo ra ngôn ngữ giống con người, mở đường cho các hệ thống AI phát triển tiên tiến hơn.

*Cuộc các mạng deep learning (2012)*

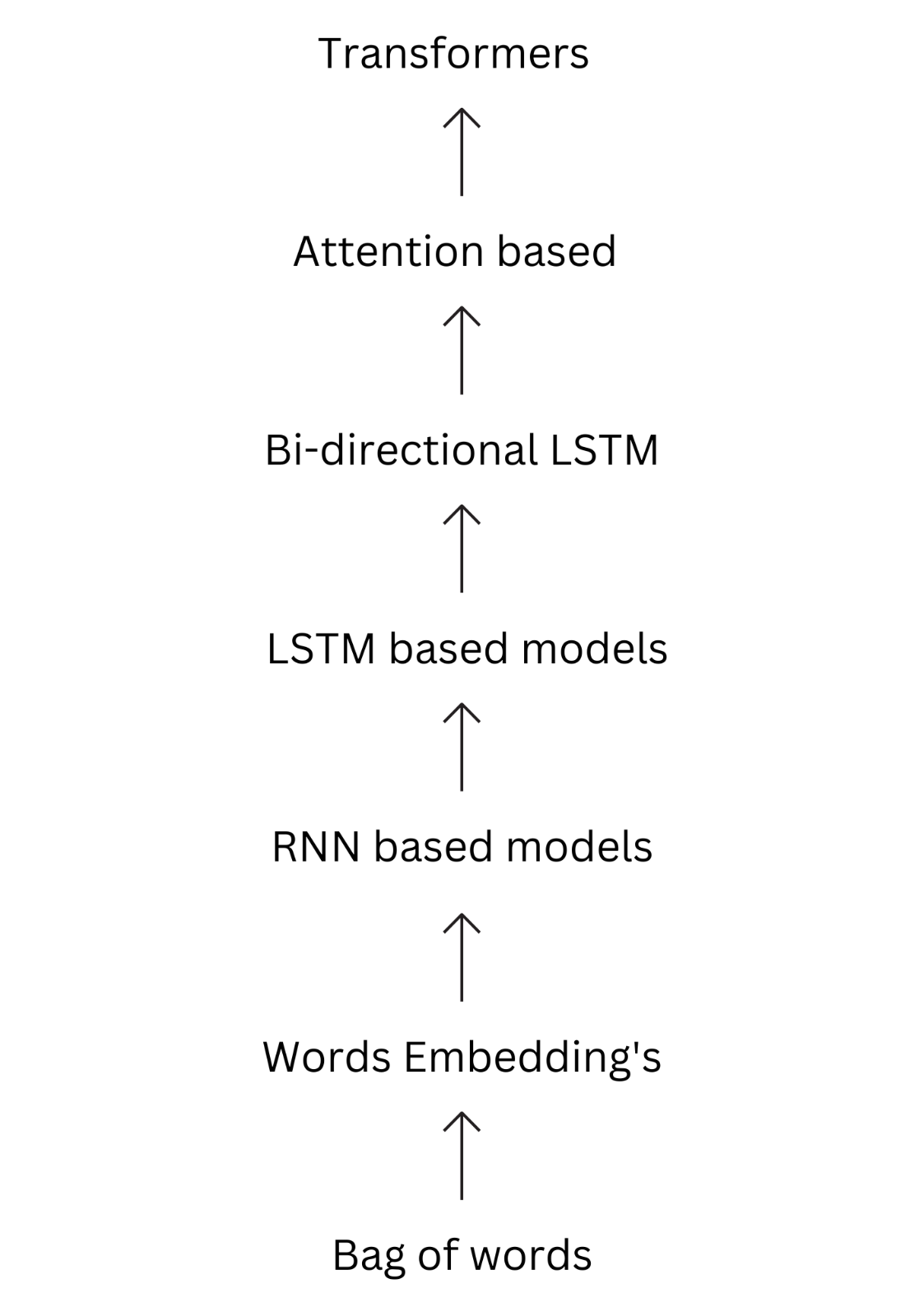
Cuộc cách mạng deep learning (2012) đánh dấu một bước ngoặt quan trọng trong lịch sử của machine learning. Cuộc các mạng này đặc biệt bởi sự ra đời của các kỹ thuật học sâu cho phép khả năng giải quyết vấn đề và nhận dạng mẫu phức tạp hết các mô hình trước đó. Một trong những bước phát triển quan trọng trong cuộc cách mạng này là việc tạo ra AlexNet, một mạng lưới thần kinh sâu giúp cải thiện đáng kể độ chính xác của hệ thống nhận dạng hình ảnh. Sự ra đời của AlexNet đã cho phép máy móc giải quyết những thách thức ngày càng phức tạp hơn, cuối cùng dẫn đến việc tạo ra:

* Hệ thống nhận dạng giọng nói chính xác hơn
* Cải thiện khả năng nhận dạng hình ảnh và video.
* Thuật toán xử lý ngôn ngữ tự nhiên tốt hơn.
* Hệ thống đề xuất nâng cao.
* Xe tự hành hiệu quả hơn

Những hệ thống phức tạp này dựa vào việc phân tích dữ liệu trong quá trình đào tạo của máy tính. Hiệu quả phần lớn phụ thuộc vào các mô hình học máy được sử dụng trong nền tảng học máy của chính chúng.

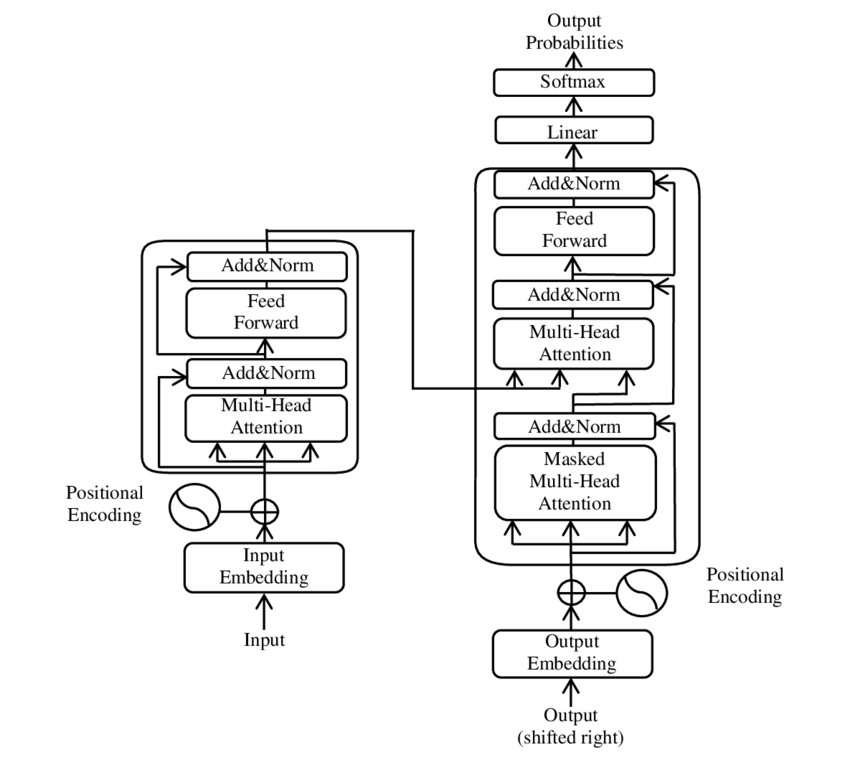
*Mô hình Transformer của Google (2017)*

Transformer là một kiến trúc deep learning dựa trên cơ chế chú ý nhiều đầu vào song song được Google đề xuất vào năm 2017.



Hình 2. 9: Dòng thời gian xuất hiện của các mô hình xử lý ngôn ngữ tự nhiên.

Trước khi có Transformer, các mô hình như RNN (Recurrent Neural Networks) và LSTM (Long Short-Term Memory) đã được sử dụng rộng rãi trong NLP. Tuy nhiên, những mô hình này gặp nhiều khó khăn trong việc xử lý các chuỗi dữ liệu dài do tính chất tuần tự của chúng. Transformer khắc phục những hạn chế này bằng cách sử dụng cơ chế attention, cho phép mô hình có thể xử lý các phần khác nhau của chuỗi đầu vào một cách song song.



Hình 2.10: Cấu trúc chính của mô hình Transformer.

Transformer bao gồm hai phần chính: bộ mã hóa (encoder) và bộ giải mã (decoder).

Bộ mã hóa của Transformer bao gồm một chuỗi các lớp mã hóa giống nhau. Mỗi lớp mã hóa bao gồm hai thành phần chính:

* **Multi-Head Self-Attention Mechanism**: Cho phép mô hình chú ý đến các phần khác nhau của chuỗi đầu vào một cách đồng thời. Cơ chế self-attention tính toán trọng số cho từng phần của đầu vào dựa trên mức độ liên quan của chúng với nhau.
* **Feed-Forward Neural Network**: Một mạng neural đơn giản được áp dụng độc lập cho từng vị trí trong chuỗi đầu vào.
* Mỗi lớp mã hóa còn bao gồm các lớp chuẩn hóa (layer normalization) và các cơ chế bỏ lớp (residual connections) để giúp huấn luyện mô hình hiệu quả hơn.
* Bộ giải mã cũng bao gồm một chuỗi các lớp giải mã giống nhau, với cấu trúc tương tự bộ mã hóa nhưng thêm một thành phần quan trọng:
* **Masked Multi-Head Self-Attention**: Giúp đảm bảo rằng mỗi vị trí trong chuỗi đầu ra chỉ có thể chú ý đến các vị trí trước đó trong chuỗi, để không tiết lộ thông tin tương lai.
* **Multi-Head Attention Over Encoder's Output**: Cho phép bộ giải mã chú ý đến các thông tin từ bộ mã hóa, kết hợp thông tin từ chuỗi đầu vào và chuỗi đầu ra để tạo ra dự đoán.

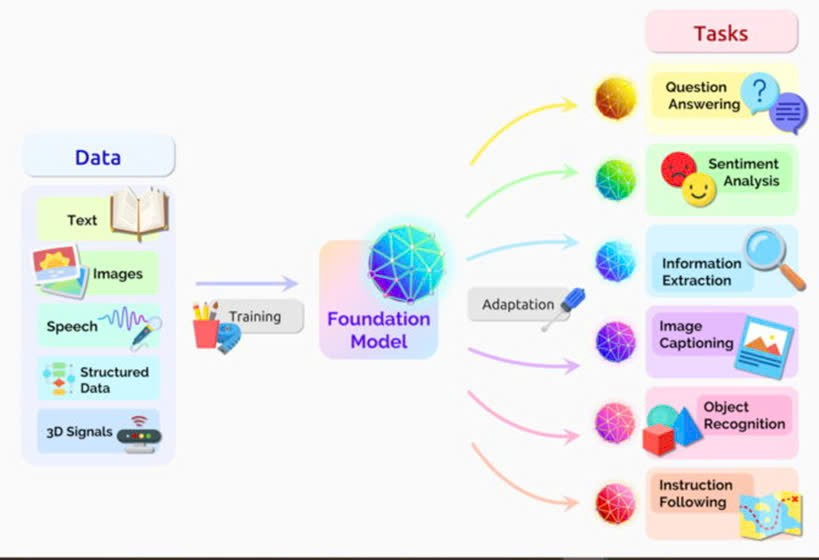
Cơ chế attention là trọng tâm của mô hình Transformer. Nó cho phép mô hình tập trung vào các phần quan trọng của chuỗi đầu vào để tạo ra biểu diễn ngữ nghĩa tốt hơn. Cơ chế này được thực hiện qua các bước:

* **Scaled Dot-Product Attention**: Tính toán trọng số chú ý bằng cách nhân các vector truy vấn (query) với các vector khóa (key), chia cho căn bậc hai của chiều dài vector, và áp dụng hàm softmax để chuẩn hóa.
* **Multi-Head Attention**: Sử dụng nhiều “đầu” attention độc lập để cho phép mô hình học được các khía cạnh khác nhau của thông tin ngữ nghĩa

Mô hình Transformer đã tạo ra bước đột phá lớn trong lĩnh vực xử lý ngôn ngữ tự nhiên nhờ vào khả năng xử lý song song và cơ chế attention mạnh mẽ. Nó đã trở thành nền tảng cho nhiều tiến bộ trong NLP và tiếp tục là một công cụ quan trọng cho các nghiên cứu và ứng dụng tương lai. Cho phép phát triển các mô hình xử lý ngôn ngữ quy mô lớn, chẳng hạn như GPT-3.

* Cải thiện đáng kể hiệu suất và hiệu quả các các tác vụ xử lý ngôn ngữ tự nhiên, ví dụ như dịch máy và mô hình hóa ngôn ngữ.
* Mở đường cho sự phát triển của các mô hình học sâu tiên tiến khác như Switch Transformer, có thể mở rộng thành các mô hình có nhiều tham số với độ chính xác cao và hiện quả hơn.

Transformer đã được sử dụng trong nhiều ứng dụng khác nhau, bao gồm chatbot, trợ lý ảo và hệ thống trả lời câu hỏi.



Hình 2.11: Ứng dụng của mô hình Transformer trên nhiều lĩnh vực khác nhau.

*Xử lý ngôn ngữ tự nhiên và GPT-3 (2020)*

Vào năm 2020, GPT-3 của OpenAI mở ra một chương mới của lĩnh vực trí tuệ nhân tạo khi đã thể hiện tiềm năng của xử lý ngôn ngữ tự nhiên (NLP) trong học máy, cho phép tạo và hiểu ngôn ngữ nâng cao, trong đó có cả ngôn ngữ lập trình. NLP đề cập đến khả năng hiểu và phân tích ngôn ngữ của con người khi nó được nói và viết, cho phép máy móc hiểu và tạo ra ngôn ngữ giống con người. Các khả năng NLP tiên tiến của nó có khả năng cách mạng hóa cách máy móc tương tác với con người, dẫn đến các ứng dụng học máy hiệu quả và chính xác hơn.

*GPT-4, Claude AI và Machine Learning hiện đại (2023)*

Những tiến bộ gần đây trong lĩnh vực học máy bao gồm GPT-4 và Claude AI. Chúng liên tục thách thức giới hạn của những gì có thể làm được trong lĩnh vực trí tuệ nhân tạo. Những tiến bộ mới trong ML cũng dẫn đến các nền tảng tạo hình ảnh, đặc biệt là Midjourney và DALL-E.

**GPT-4** , do OpenAI tạo ra, là một mô hình đa phương thức lớn có thể chấp nhận cả đầu vào văn bản và hình ảnh, đồng thời tạo ra đầu ra văn bản giống với ngôn ngữ của con người. Trí tưởng tượng và khả năng hợp tác của nó khiến nó trở thành một công cụ mạnh mẽ cho các ứng dụng học máy. Rất nhiều ứng dụng phụ thuộc vào GPT-4, bao gồm [Bing AI](https://bing.com/?/ai), [Duolingo](https://duolingo.com/), tác giả AI như [Jasper](https://jasper.ai/) , nền tảng phân tích dữ liệu như [Akkio](https://akkio.com/) và mạng xã hội như Snapchat. [Microsoft đã đầu tư 10 tỷ USD vào OpenAI để phát triển hơn](https://www.nytimes.com/2023/01/23/business/microsoft-chatgpt-artificial-intelligence.html) nữa

**Claude AI** là một nền tảng học sâu tạo điều kiện cho việc phát triển và triển khai nhanh chóng các ứng dụng AI, cung cấp nhiều tính năng, bao gồm xử lý ngôn ngữ tự nhiên, thị giác máy tính và học tăng cường.

[**Midjourney**](https://www.midjourney.com/) là một phòng thí nghiệm nghiên cứu độc lập cung cấp nền tảng cho các công cụ sáng tạo được hỗ trợ bởi AI để tạo hình ảnh trực quan với giao diện ngôn ngữ tự nhiên. Trình tạo hình ảnh AI của Midjourney có thể tạo hình ảnh tùy chỉnh cho các chiến dịch tiếp thị và quảng cáo, chẳng hạn như bài đăng trên mạng xã hội, quảng cáo biểu ngữ và tiếp thị qua email. Midjourney đã sử dụng [**bộ tăng tốc AI được phát triển tùy chỉnh của Google Cloud**](https://www.prnewswire.com/news-releases/midjourney-selects-google-cloud-to-power-ai-generated-creative-platform-301771558.html) , Bộ xử lý Tensor (TPU), để huấn luyện mô hình AI thế hệ thứ tư, có thể hiển thị các hình ảnh được tạo trên nền tảng với tốc độ ngoạn mục. Nền tảng sáng tạo do AI tạo ra của Midjourney đã được hơn **11 triệu thành viên** sử dụng và công ty gần đây đã [hợp tác với Google Cloud để hỗ trợ nền tảng của mình](https://www.prnewswire.com/news-releases/midjourney-selects-google-cloud-to-power-ai-generated-creative-platform-301771558.html) .



Hình 2.12: Một số hình ảnh được tạo ra từ Midjourney.

[**DALL-E**](https://www.axelerant.com/blog/how-to-use-ai-and-ml-to-enhance-dxps) là mô hình AI được phát triển bởi OpenAI, có thể **tạo ra hình ảnh từ các mô tả văn bản** . DALL-E có thể tạo ra hình ảnh của những vật thể không tồn tại trong thế giới thực, chẳng hạn như con ốc sên làm bằng dây đàn hạc hay chiếc ấm trà có hình dạng giống con ốc sên. Tác động của DALL-E đối với lĩnh vực công cụ sáng tạo do AI tạo ra là rất đáng kể và nó đã thu hút sự chú ý của các nhà nghiên cứu, nghệ sĩ cũng như nhà thiết kế.

Những tiến bộ liên tục trong học máy, chẳng hạn như GPT-4 và Claude AI, đang chuyển đổi lĩnh vực này và tạo điều kiện cho các ứng dụng mới và sáng tạo trong nhiều ngành công nghiệp. Khi học máy tiếp tục phát triển và thích ứng, nó được kỳ vọng sẽ đóng vai trò ngày càng quan trọng trong **việc định hình tương lai của công nghệ và xã hội** , chuyển đổi các ngành công nghiệp và xác định lại sự tương tác giữa con người với máy tính

2.1.5. Xu hướng phát triển.

Trong những năm tới, Machine Learning (ML) dự kiến sẽ trải qua một giai đoạn phát triển mạnh mẽ và đa dạng, mang lại những thay đổi quan trọng trong cách chúng ta tận dụng công nghệ để giải quyết các vấn đề phức tạp. Xu hướng phát triển của ML không chỉ tập trung vào việc cải thiện hiệu suất và độ chính xác của các mô hình, mà còn hướng tới việc mở rộng phạm vi ứng dụng và tăng cường khả năng tiếp cận của công nghệ này.

Một trong những xu hướng nổi bật nhất là sự tiến bộ liên tục của Deep Learning. Các kiến trúc mạng neural ngày càng trở nên phức tạp và tinh vi hơn, cho phép xử lý các tác vụ phức tạp với độ chính xác cao hơn. Chúng ta có thể kỳ vọng vào sự xuất hiện của các mô hình ngôn ngữ lớn tiên tiến hơn, có khả năng hiểu và tạo ra ngôn ngữ tự nhiên gần với con người hơn. Trong lĩnh vực thị giác máy tính, các mô hình deep learning sẽ tiếp tục cải thiện khả năng nhận dạng và phân tích hình ảnh, mở ra những ứng dụng mới trong y tế, an ninh, và tự động hóa.

Song song với đó, Reinforcement Learning (RL) đang được kỳ vọng sẽ đạt được những bước tiến quan trọng. RL có tiềm năng to lớn trong việc đào tạo các hệ thống tự động học hỏi và thích nghi với môi trường phức tạp. Trong lĩnh vực robotics, RL có thể giúp các robot thực hiện các nhiệm vụ phức tạp và linh hoạt hơn trong môi trường thực tế. Ngoài ra, RL cũng đang được áp dụng rộng rãi trong tối ưu hóa quyết định, từ quản lý chuỗi cung ứng đến giao dịch tài chính tự động.

Xu hướng học liên tục (Continual Learning) đang nổi lên như một giải pháp cho vấn đề "quên lãng thảm họa" của các mô hình ML. Kỹ thuật này cho phép các mô hình tiếp tục học và thích nghi với dữ liệu mới mà không làm mất đi kiến thức đã học trước đó. Điều này đặc biệt quan trọng trong các ứng dụng thực tế, nơi dữ liệu và môi trường liên tục thay đổi. Continual Learning hứa hẹn sẽ tạo ra các hệ thống AI linh hoạt hơn, có khả năng thích ứng tốt hơn với thế giới thực.

Bên cạnh đó, kỹ thuật học ít mẫu (Few-shot Learning) đang được phát triển mạnh mẽ. Xu hướng này nhắm đến việc tạo ra các mô hình có khả năng học hiệu quả từ lượng dữ liệu huấn luyện hạn chế. Điều này đặc biệt quan trọng trong các lĩnh vực mà việc thu thập dữ liệu là khó khăn hoặc tốn kém, như y học cá nhân hóa hoặc nhận dạng các loài động vật quý hiếm. Few-shot Learning có thể mở rộng đáng kể phạm vi ứng dụng của ML trong các lĩnh vực có dữ liệu hạn chế.

Xu hướng AutoML (Automated Machine Learning) đang dần trở nên phổ biến, hướng tới việc tự động hóa quá trình thiết kế và tối ưu hóa mô hình ML. AutoML không chỉ giúp giảm thời gian và công sức cần thiết để phát triển các mô hình ML hiệu quả, mà còn góp phần democratize ML bằng cách mở rộng khả năng tiếp cận công nghệ này cho nhiều đối tượng hơn, kể cả những người không có chuyên môn sâu về ML.

Cuối cùng, xu hướng hướng tới AI có thể giải thích được (Explainable AI) và công bằng trong ML đang ngày càng được chú trọng. Khi các hệ thống ML ngày càng được áp dụng rộng rãi trong các quyết định quan trọng, việc hiểu được cách thức hoạt động của chúng và đảm bảo tính công bằng trở nên cực kỳ quan trọng. Các nhà nghiên cứu đang phát triển các kỹ thuật mới để làm cho các quyết định của mô hình ML trở nên minh bạch và dễ hiểu hơn, đồng thời phát triển các phương pháp để phát hiện và giảm thiểu thiên kiến trong các mô hình.

## Giới thiệu tổng quát.

K-means clustering là một thuật toán phân cụm phổ biến trong học máy và khai thác dữ liệu, được sử dụng rộng rãi để phân nhóm dữ liệu thành các cụm riêng biệt dựa trên các đặc điểm tương đồng. Được giới thiệu lần đầu bởi Stuart Lloyd vào năm 1957, K-means nhanh chóng trở thành công cụ chủ đạo trong nhiều ứng dụng, từ phân tích hình ảnh, nhận diện mẫu đến phân đoạn thị trường và xử lý tín hiệu. Thuật toán này không chỉ dễ hiểu mà còn hiệu quả về mặt tính toán, đặc biệt phù hợp với những dữ liệu lớn và phức tạp trong các lĩnh vực khác nhau.

Nguyên tắc chính của K-means clustering là tìm cách chia dữ liệu thành *k* cụm riêng biệt sao cho khoảng cách giữa các điểm dữ liệu trong cùng một cụm là nhỏ nhất có thể, đồng thời khoảng cách giữa các cụm là lớn nhất. Cụ thể, thuật toán bắt đầu bằng cách chọn ngẫu nhiên kkk điểm làm tâm cụm, sau đó tiến hành hai bước chính: gán mỗi điểm dữ liệu vào cụm có tâm cụm gần nhất và cập nhật lại vị trí của tâm cụm dựa trên các điểm trong cụm đó. Quá trình này lặp lại cho đến khi các tâm cụm không thay đổi hoặc số lần lặp đạt đến một ngưỡng đã định, khi đó thuật toán đạt trạng thái hội tụ.

A diagram of a diagram of a number of dots

Description automatically generated

Hình 2. :Minh họa thuật toán phân cụm K-means

Một đặc điểm quan trọng của K-means là tính toán trọng tâm dựa trên khoảng cách Euclidean, vì vậy K-means thường thích hợp nhất khi các cụm có hình cầu và kích thước tương đồng. Đối với dữ liệu có hình dạng cụm phức tạp hoặc không cân đối, thuật toán này có thể không phân cụm tối ưu. Thêm vào đó, việc chọn số lượng cụm kkk là một bước quan trọng trong K-means. Nếu chọn kkk quá nhỏ, các cụm sẽ không phản ánh được hết các đặc điểm của dữ liệu, trong khi chọn kkk quá lớn có thể làm mất tính tổng quát của mô hình. Một trong những phương pháp phổ biến để xác định kkk tối ưu là phương pháp Elbow, dựa vào độ giảm của tổng bình phương sai số (SSE) khi tăng số cụm.

K-means được đánh giá cao nhờ tốc độ nhanh và khả năng triển khai dễ dàng. Vì vậy, nó thường được sử dụng trong các bài toán yêu cầu xử lý nhanh chóng trên lượng lớn dữ liệu, đặc biệt trong các bài toán thực tế như nén ảnh, phân đoạn ảnh, và nhận diện mẫu. Khi ứng dụng trong phân đoạn ảnh, K-means có khả năng phân chia các điểm ảnh thành các cụm tương ứng với các vùng màu sắc tương đồng, giúp làm nổi bật các vùng đặc trưng trong ảnh. Trong xử lý tín hiệu và dữ liệu tài chính, K-means có thể giúp phát hiện các nhóm khách hàng có hành vi tương tự, hoặc xác định các kiểu mẫu trong biến động giá cổ phiếu.

Mặc dù K-means clustering rất hiệu quả, thuật toán này cũng có một số hạn chế, chẳng hạn như nhạy cảm với các điểm dữ liệu ngoại lai (outliers) và phụ thuộc vào giá trị khởi tạo của các tâm cụm. Để cải thiện tính ổn định của thuật toán, nhiều biến thể của K-means đã được phát triển, chẳng hạn như K-means++ giúp chọn giá trị khởi tạo tốt hơn cho các tâm cụm, từ đó giảm thiểu nguy cơ hội tụ tại các cực trị cục bộ.

## Các thuật ngữ liên quan

Trong K-means clustering, có một số thuật ngữ quan trọng thường được sử dụng để mô tả quá trình và các yếu tố liên quan. Bảng dưới đây trình bày tóm tắt các thuật ngữ chính và ý nghĩa của nó trong thuật toán K-means.

Bảng 2. : Các thuật ngữ chính trong K-mean và ý nghĩa

|  |  |
| --- | --- |
| **Thuật ngữ** | **Mô tả** |
| Cluster (Cụm) | Nhóm các điểm dữ liệu có đặc điểm tương đồng với nhau. |
| Centroid (Tâm cụm) | Điểm trung tâm của một cụm, đại diện cho cụm, tính bằng giá trị trung bình của các điểm trong cụm. |
| Inertia | Tổng bình phương khoảng cách giữa mỗi điểm dữ liệu và tâm cụm gần nhất, dùng để đánh giá mức độ chặt chẽ của các cụm. |
| Distance Metric | Phương pháp đo khoảng cách giữa các điểm dữ liệu và tâm cụm; thường dùng khoảng cách Euclidean. |
| Initialization | Quá trình chọn vị trí ban đầu cho các tâm cụm trước khi lặp, có thể ngẫu nhiên hoặc dùng K-means++ để tối ưu hóa. |
| Convergence | Trạng thái khi thuật toán đạt hội tụ, các tâm cụm không thay đổi đáng kể nữa. |
| K (Số lượng cụm) | Số cụm mà dữ liệu sẽ được phân thành, xác định trước khi chạy thuật toán. |
| Assignment Step | Bước gán mỗi điểm dữ liệu vào cụm có tâm cụm gần nhất, thực hiện trong mỗi vòng lặp. |
| Update Step | Bước cập nhật lại vị trí tâm cụm bằng cách tính trung bình các điểm trong cụm, thực hiện trong mỗi vòng lặp. |
| Elbow Method | Phương pháp xác định số cụm kkk tối ưu bằng cách chọn điểm "khuỷu tay" trên đồ thị inertia. |
| Silhouette Score | Thước đo hiệu quả phân cụm, giá trị từ -1 đến 1, với giá trị cao cho thấy các cụm được phân tách rõ ràng. |
| Iterations | Số vòng lặp của thuật toán từ khi bắt đầu đến khi hội tụ. |
| Local Minima | Cực tiểu cục bộ của hàm mục tiêu, nơi thuật toán có thể hội tụ nhưng không đạt tối ưu toàn cục. |
| Outliers | Các điểm dữ liệu nằm xa các cụm chính, có thể ảnh hưởng đến độ chính xác của kết quả phân cụm. |
| Dimensionality | Số lượng đặc trưng trong tập dữ liệu, ảnh hưởng đến tính phức tạp của K-means; dữ liệu nhiều chiều có thể cần thêm xử lý hoặc giảm chiều trước khi phân cụm. |

## Cách K-Means hoạt động.

K-means clustering hoạt động dựa trên một quy trình lặp để phân cụm dữ liệu thành các nhóm khác nhau, tối ưu hóa sao cho các điểm dữ liệu trong mỗi cụm có mức độ tương đồng cao nhất. Mỗi cụm được đặc trưng bởi một “tâm cụm” (centroid), được tính toán sao cho khoảng cách giữa tâm cụm và các điểm dữ liệu trong cùng cụm là nhỏ nhất. Thuật toán có thể được chia thành các bước chính như sau:

1. **Khởi tạo tâm cụm ban đầu**: Thuật toán bắt đầu bằng cách xác định trước số cụm kkk và chọn ngẫu nhiên kkk điểm dữ liệu làm tâm cụm ban đầu. Tâm cụm có thể được chọn bằng phương pháp ngẫu nhiên hoặc với các phương pháp tối ưu hơn như K-means++ để đạt được sự phân cụm hiệu quả hơn.
2. **Gán điểm dữ liệu vào các cụm**: Sau khi có các tâm cụm ban đầu, mỗi điểm dữ liệu sẽ được gán vào cụm có tâm cụm gần nhất. Khoảng cách từ mỗi điểm đến tâm cụm được tính toán thường bằng khoảng cách Euclidean, tuy nhiên các khoảng cách khác cũng có thể được sử dụng tùy thuộc vào bài toán.
3. **Cập nhật tâm cụm**: Sau khi các điểm đã được gán vào cụm, thuật toán tính toán lại vị trí của mỗi tâm cụm bằng cách lấy trung bình của tất cả các điểm trong cụm đó. Việc này đảm bảo rằng mỗi tâm cụm được điều chỉnh để nằm chính giữa các điểm trong cụm.
4. **Lặp lại cho đến khi hội tụ**: Bước gán điểm dữ liệu vào cụm và cập nhật tâm cụm sẽ tiếp tục lặp lại cho đến khi các tâm cụm không thay đổi nhiều (hoặc không thay đổi hoàn toàn), tức là đạt trạng thái hội tụ. Đôi khi, quá trình này được dừng lại nếu số vòng lặp đạt đến một ngưỡng nhất định để tránh việc lặp vô hạn.

Giả sử chúng ta có dữ liệu của một chuỗi cửa hàng cà phê muốn phân chia khách hàng thành hai nhóm dựa trên số lần đến cửa hàng (biểu thị thói quen trung thành) và mức chi tiêu trung bình mỗi lần đến. Mỗi khách hàng sẽ là một điểm dữ liệu với hai thuộc tính: “số lần đến” và “mức chi tiêu trung bình”. Các bước thực hiện K-means với ví dụ này như sau:

1. Chọn số lượng cụm k=2: Do chuỗi cà phê muốn phân chia thành hai nhóm khách hàng, chúng ta chọn số lượng cụm k=.
2. Khởi tạo tâm cụm ban đầu: Chọn ngẫu nhiên hai điểm khách hàng từ dữ liệu, hoặc tạo hai điểm đại diện, làm tâm cụm ban đầu.
3. Gán khách hàng vào các cụm: Với mỗi khách hàng (điểm dữ liệu), tính khoảng cách đến hai tâm cụm. Nếu khách hàng có khoảng cách gần tâm cụm 1 hơn tâm cụm 2, thì khách hàng đó sẽ được gán vào cụm 1, và ngược lại.
4. Cập nhật tâm cụm: Sau khi các khách hàng đã được gán vào các cụm, tính lại tâm cụm bằng cách lấy giá trị trung bình của số lần đến và mức chi tiêu trung bình của tất cả các khách hàng trong mỗi cụm. Tâm cụm mới sẽ phản ánh chính xác vị trí trung tâm của các khách hàng trong cụm đó.
5. Lặp lại quá trình: Các bước gán và cập nhật tâm cụm tiếp tục lặp lại cho đến khi các tâm cụm không thay đổi hoặc thay đổi rất ít. Khi đó, thuật toán đã hội tụ và chúng ta có được hai cụm khách hàng với đặc điểm khác biệt rõ ràng: một nhóm có tần suất đến cửa hàng cao và chi tiêu nhiều, và nhóm còn lại đến ít hơn hoặc chi tiêu ít hơn.

Kết quả cuối cùng sẽ là hai cụm khách hàng, mỗi cụm có một đặc trưng riêng. Ví dụ, cụm đầu tiên có thể đại diện cho nhóm khách hàng trung thành, ghé thăm thường xuyên và chi tiêu nhiều, trong khi cụm thứ hai có thể đại diện cho những khách hàng đến ít hơn và chi tiêu ít hơn. Thông tin này giúp chuỗi cà phê tối ưu hóa chiến lược marketing, thiết kế chương trình khuyến mãi cho nhóm khách hàng trung thành, đồng thời tìm cách thu hút và khuyến khích nhóm khách hàng ít đến ghé thăm thường xuyên hơn. Phương pháp K-means, qua ví dụ này, cho thấy khả năng phân nhóm dữ liệu rõ ràng và hiệu quả, đồng thời cung cấp cái nhìn trực quan về đặc điểm của các nhóm dữ liệu trong các ứng dụng thực tế.

## 2.4. Các ứng dụng của Kmeans

Kmeans là thuật toán phổ biến và nổi bật trong việc phân cụm, vì thế nó có nhiều ứng dụng trong nhiều lĩnh vực khác nhau, có thể kể đến như:

* Phân cụm khách hàng:Phân nhóm khách hàng theo hành vi mua sắm hoặc sở thích để tối ưu hóa chiến dịch marketing và cá nhân hóa đề xuất sản phẩm.
* Nén ảnh: Giảm số lượng màu sắc trong hình ảnh bằng cách nhóm màu tương tự lại với nhau, giúp giảm kích thước ảnh mà vẫn giữ được chất lượng.
* Phân tích thị trường chứng khoán: Phân nhóm cổ phiếu dựa trên các đặc điểm tài chính để đánh giá rủi ro và xây dựng danh mục đầu tư.
* Phân loại văn bản: Phân nhóm các văn bản theo chủ đề, hỗ trợ trong xử lý ngôn ngữ tự nhiên như phân loại email, tài liệu hoặc bài báo.
* Phân cụm dữ liệu địa lý: Sử dụng trong GIS để phân nhóm các điểm dữ liệu địa lý, hỗ trợ quản lý tài nguyên hoặc lập kế hoạch đô thị.
* Phân tích gen trong sinh học: Phân nhóm các gen tương tự giúp nghiên cứu sự phát triển và mối liên hệ giữa các gen.
* Phân tích hành vi người dùng: Phân loại hành vi của người dùng trên trang web, giúp cải thiện trải nghiệm người dùng và tối ưu hóa quảng cáo.
* Chẩn đoán lỗi trong công nghiệp: Sử dụng trong IoT để phân tích dữ liệu từ các cảm biến, phát hiện các sự cố và bảo trì phòng ngừa trong hệ thống công nghiệp.

K-means được áp dụng rộng rãi nhờ tính linh hoạt và hiệu quả trong việc phân cụm dữ liệu lớn, từ thương mại đến công nghệ sinh học và kỹ thuật công nghiệp.

# CHƯƠNG III. ỨNG DỤNG THUẬT TOÁN KMEANS TRONG XỬ LÝ ẢNH

## Triển khai ứng dụng.

### Xây dựng giao diện

Sử dụng thư viện Pygame như đã đề cập trong chương I, xây dựng một giao diện desktop như sau:

A screenshot of a computer program

Description automatically generated

Hình 3. : Giao diện chương trình trực quan hóa Kmeans

Trong chương trình minh họa thuật toán K-means với pygame, các thành phần chính của giao diện bao gồm việc khởi tạo cửa sổ và các thiết lập cơ bản, định nghĩa các nút chức năng, xử lý sự kiện của người dùng và vòng lặp vẽ giao diện. Đầu tiên, thư viện pygame được khởi tạo và cửa sổ chính có kích thước 1200x700 được tạo để hiển thị trực quan các thao tác của thuật toán. Màu sắc và font chữ cũng được định nghĩa, giúp giao diện thêm sinh động và trực quan hơn.

Các thành phần tương tác chính của giao diện bao gồm:

* Nút "+" và "-": Tăng hoặc giảm số lượng cụm KKK.
* Nút "Run": Thực hiện phân cụm với các điểm dữ liệu đã chọn.
* Nút "Random": Khởi tạo ngẫu nhiên các cụm trung tâm trên màn hình.
* Nút "Reset": Xóa toàn bộ dữ liệu, đặt lại chương trình về trạng thái ban đầu.

Chương trình cũng hiển thị các thông tin trực quan như tọa độ con trỏ chuột trong khu vực vẽ và giá trị lỗi của thuật toán, giúp người dùng dễ dàng quan sát quá trình phân cụm và đánh giá hiệu quả của thuật toán.

### Sơ đồ hoạt động

Trong ứng dụng minh họa trên, thuật toán K-means được thực hiện qua các bước sau:

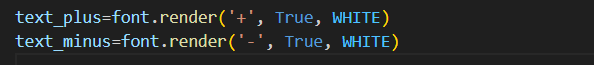
A diagram with text and arrows

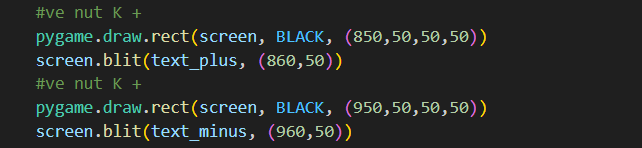
Description automatically generated with medium confidence

Hình 3. : Sơ đồ hoạt động của ứng dụng

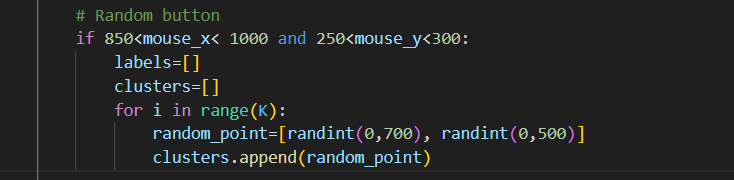
Chi tiết về các bước như sau:

* Khởi tạo số lượng cụm K:Người dùng có thể điều chỉnh số lượng cụm K thông qua các nút “+” và “-” trên giao diện. Giá trị K sẽ xác định số cụm mà thuật toán cần phân chia các điểm dữ liệu.

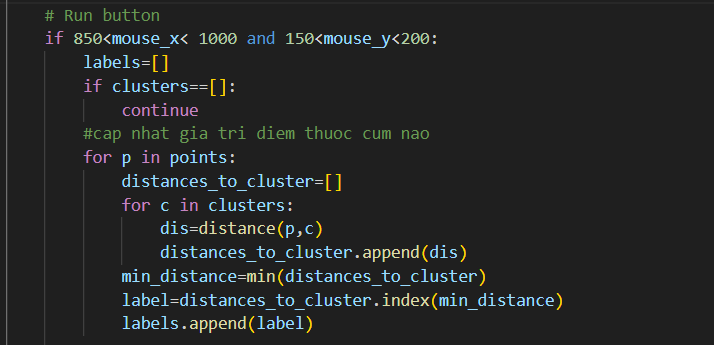




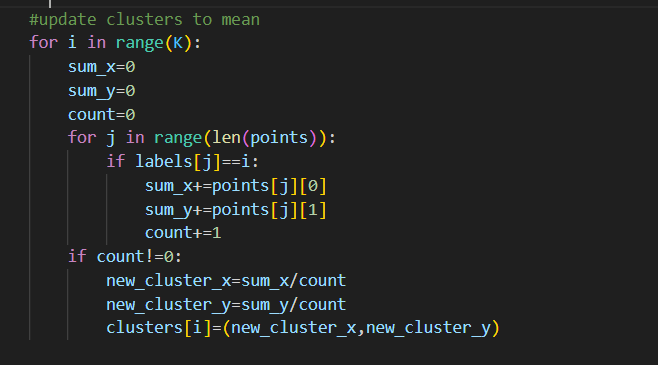
* Tạo các cụm trung tâm ngẫu nhiên: Khi nhấn nút “Random”, chương trình sẽ sinh ra K cụm trung tâm ngẫu nhiên bên trong panel bằng cách chọn ngẫu nhiên tọa độ cho mỗi cụm trung tâm. Những cụm trung tâm này sẽ là điểm bắt đầu cho thuật toán.



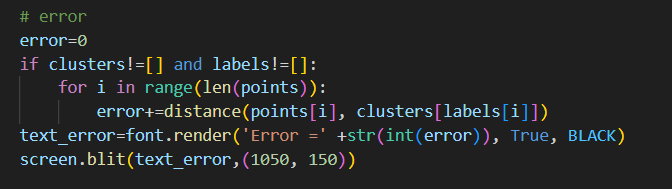
* Gán cụm cho các điểm dữ liệu: Khi nhấn nút “Run”, thuật toán bắt đầu phân cụm. Đối với mỗi điểm dữ liệu đã chọn, chương trình tính khoảng cách từ điểm đó đến từng cụm trung tâm và gán điểm vào cụm có khoảng cách ngắn nhất. Điều này được thực hiện bằng cách tính toán khoảng cách Euclidean giữa điểm và các cụm trung tâm.



* Cập nhật cụm trung tâm: Trong K-means chuẩn, sau khi gán nhãn, thuật toán sẽ tính lại vị trí các cụm trung tâm bằng cách lấy trung bình các tọa độ của các điểm trong mỗi cụm, sau đó lặp lại quá trình gán cụm cho đến khi các cụm trung tâm ổn định.



* Tính toán lỗi (Error): Giá trị lỗi được hiển thị trên giao diện thể hiện tổng khoảng cách giữa các điểm dữ liệu và cụm trung tâm của chúng. Điều này giúp đánh giá hiệu quả của việc phân cụm.



* Lặp lại quá trình (nếu cần): Người dùng có thể nhấn lại nút “Run” để gán lại cụm cho các điểm dựa trên các cụm trung tâm hiện tại. Quá trình này có thể lặp lại để cải thiện kết quả phân cụm cho đến khi người dùng hài lòng hoặc các cụm trung tâm không thay đổi đáng kể

### Kết quả khởi chạy

Chạy chương trình với K =2, kết quả như sau:

A screenshot of a computer program

Description automatically generated

Hình 3. : Kết quả chạy chương trình với K = 2

Với K = 3:

A screen shot of a diagram

Description automatically generated

Hình 3. : Kết quả chạy chương trình với K = 3

Với K = 4:

A screenshot of a computer screen

Description automatically generated

Hình 3. : Kết quả chạy chương trình với K = 4

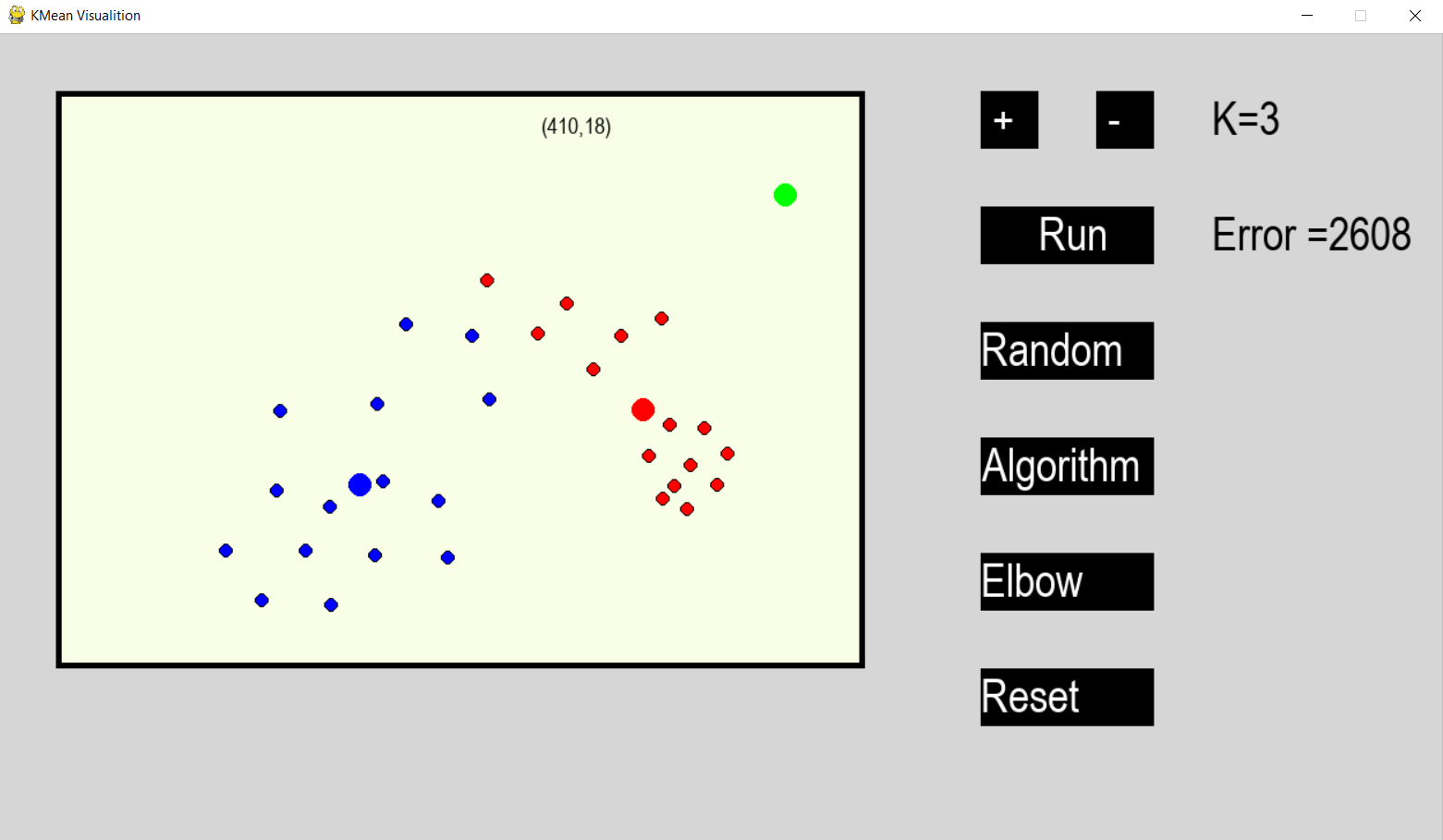
### Nhận xét

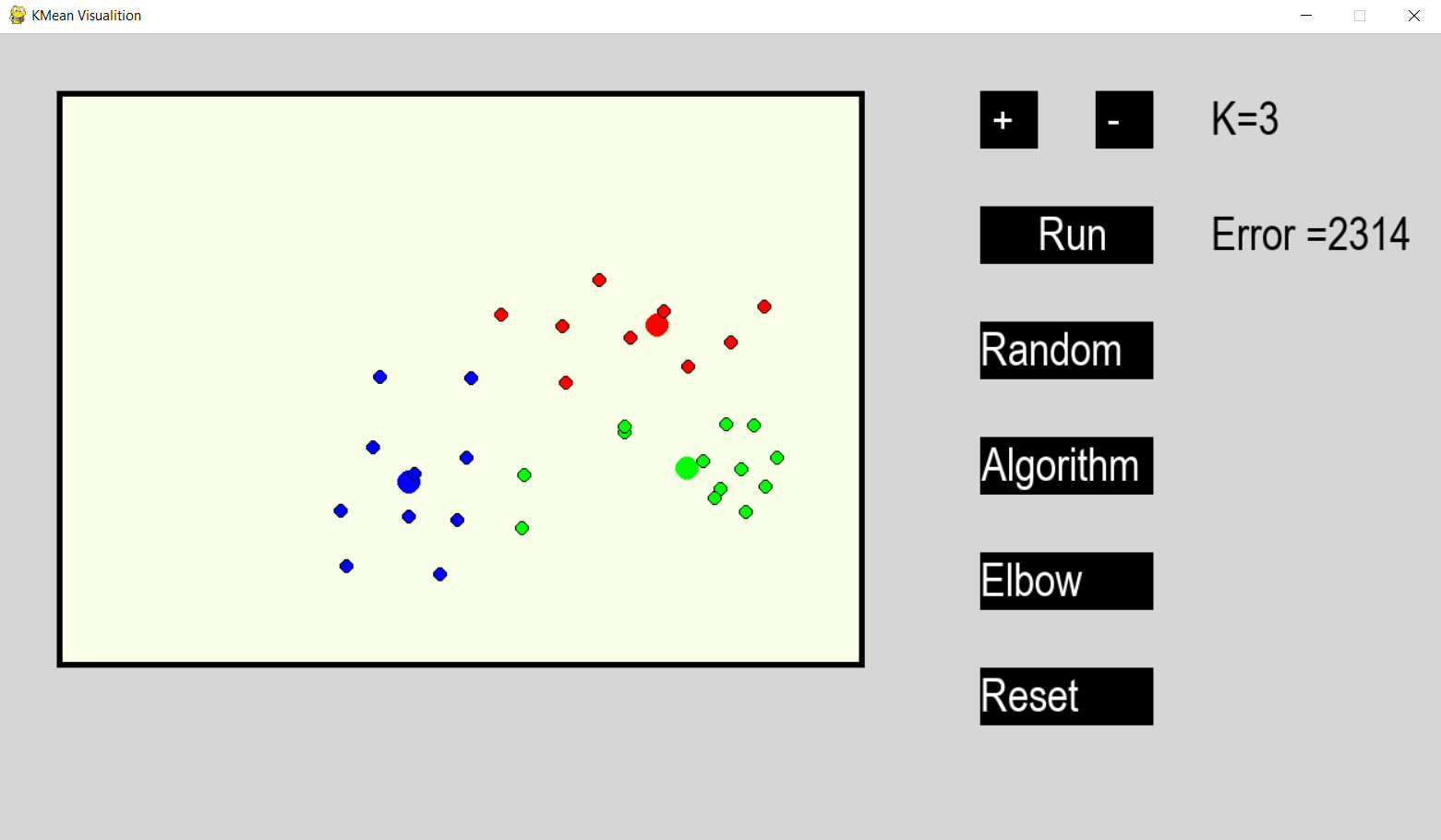
Ưu điểm của phương pháp K-means:

* Dễ hiểu và dễ triển khai: Thuật toán đơn giản, dễ lập trình và trực quan hóa giúp người dùng hiểu rõ quá trình phân cụm.
* Tính toán nhanh: K-means có thời gian tính toán nhanh, xử lý hiệu quả với tập dữ liệu lớn.
* Phân cụm hiệu quả: Thích hợp cho các nhóm có hình dạng cầu hoặc hình tròn.
* Khả năng mở rộng: Dễ dàng mở rộng cho các bài toán lớn hơn

Nhược điểm của phương pháp K-means trong đoạn mã:

* Phụ thuộc vào giá trị KKK: Số lượng cụm phải xác định trước; lựa chọn không đúng có thể dẫn đến kết quả không chính xác.



* Khởi tạo cụm trung tâm ngẫu nhiên: Kết quả có thể khác nhau tùy vào vị trí khởi tạo của các cụm trung tâm.
* 
* Không hiệu quả với cụm hình dạng phức tạp: K-means kém với các cụm không có hình dạng cầu hoặc kích thước khác nhau.
* Khó phát hiện điểm ngoại lai: Nhạy cảm với điểm dữ liệu ngoại lai, có thể làm lệch kết quả phân cụm.

### Đề xuất cải tiến

Để khắc phục nhược điểm phụ thuộc vào giá trị K phù hợp của phương pháp K-means, có thể áp dụng các biện pháp sau:

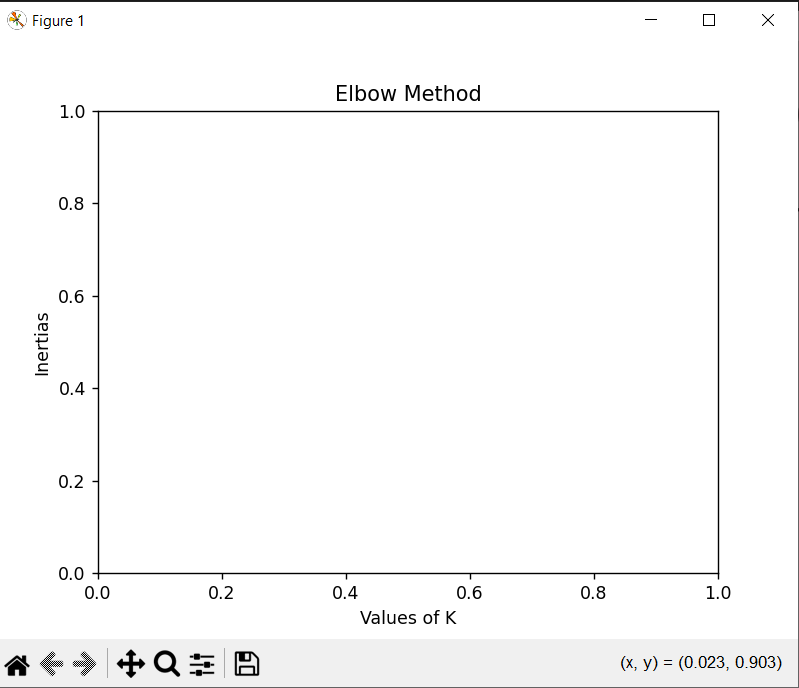
* Lựa chọn giá trị K hợp lý: Phương pháp Elbow:

Phương pháp elbow (hay phương pháp khuỷu tay) là một kỹ thuật thường được sử dụng để xác định số lượng cụm tối ưu trong thuật toán K-means clustering. Dưới đây là lý thuyết cơ bản về phương pháp này:

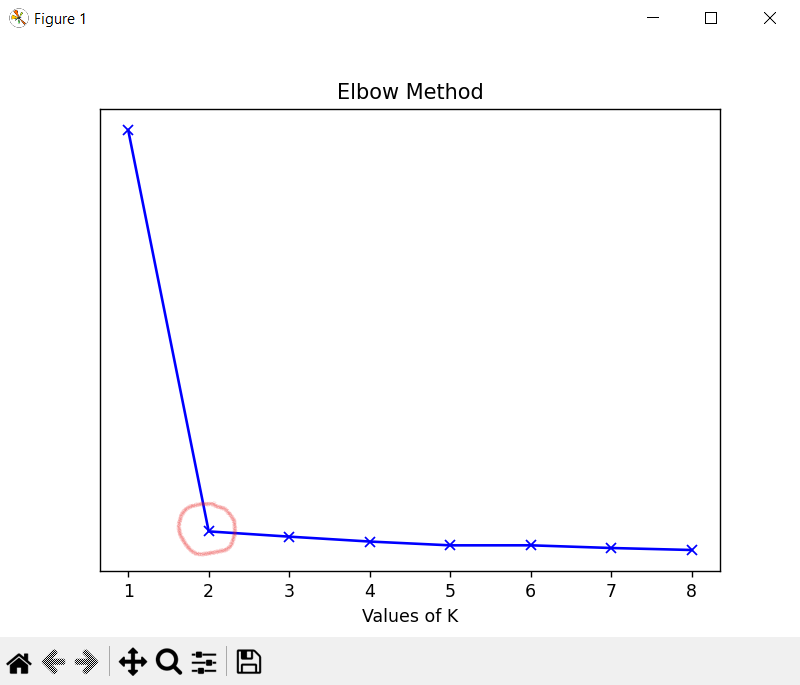
**+ Nguyên lý hoạt động của phương pháp elbow**

**Đo lường độ chính xác**: Trong K-means, khi số lượng cụm kkk tăng lên, độ chính xác của mô hình (thường được đo bằng tổng khoảng cách bình phương từ các điểm đến tâm cụm) sẽ giảm. Giá trị này được gọi là tổng độ lệch bình phương (Within-cluster Sum of Squares - WCSS).

**Vẽ biểu đồ**: Phương pháp elbow liên quan đến việc vẽ một biểu đồ với kkk trên trục hoành (x-axis) và WCSS trên trục tung (y-axis). Khi số lượng cụm kkk tăng lên, WCSS sẽ giảm, nhưng tốc độ giảm này sẽ khác nhau.



**Tìm điểm khuỷu tay**: Khi vẽ biểu đồ, người dùng sẽ thấy rằng WCSS giảm nhanh chóng ở các giá trị kkk nhỏ, nhưng sau một mức nào đó, sự giảm này bắt đầu chậm lại, tạo thành một hình dạng giống như một khuỷu tay. Điểm này được gọi là "điểm khuỷu tay" và được xem là số lượng cụm tối ưu. Số lượng cụm tại điểm này thường được coi là điểm cân bằng giữa sự đơn giản của mô hình và độ chính xác.



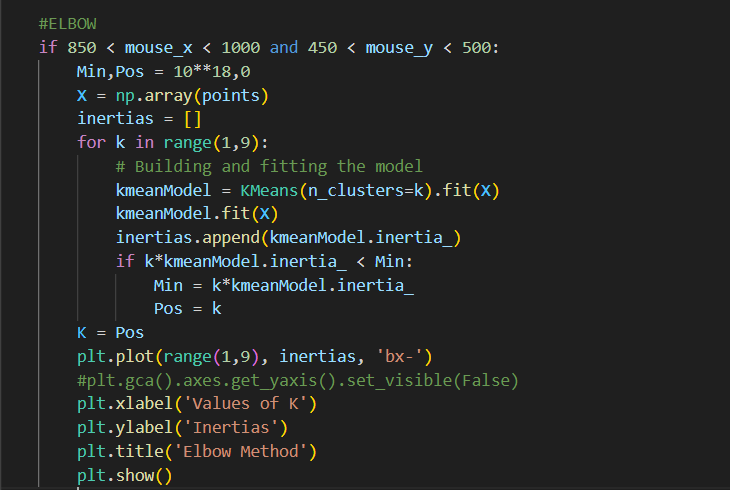
**+ Cách thực hiện phương pháp elbow**

**Chạy thuật toán K-means** cho nhiều giá trị khác nhau của kkk (ví dụ, từ 1 đến 10).

**Ghi lại giá trị WCSS** cho mỗi giá trị của kkk.

**Vẽ biểu đồ** với kkk trên trục x và WCSS trên trục y.

**Phân tích biểu đồ** để xác định điểm khuỷu tay.



**+ Lưu ý**

Phương pháp elbow không phải lúc nào cũng cho kết quả rõ ràng; đôi khi không có điểm khuỷu tay rõ ràng, trong trường hợp này, người phân tích có thể cần sử dụng các phương pháp khác như Silhouette Score hoặc Gap Statistic để xác định số lượng cụm tối ưu.

Mặc dù phương pháp này hữu ích, nhưng nó phụ thuộc vào sự trực quan của người phân tích và có thể mang tính chủ quan.

Phương pháp elbow là một trong những cách phổ biến nhất để xác định số lượng cụm trong K-means và là công cụ hữu ích trong phân tích dữ liệu.

### Hướng phát triển

Để khắc phục những nhược điểm của phương pháp K-means, có thể áp dụng các biện pháp sau:

* Nhiều khởi tạo ngẫu nhiên: Chạy K-means nhiều lần và chọn kết quả tốt nhất.
* Phát triển thuật toán phân cụm khác như DBSCAN hoặc GMM: Sử dụng các thuật toán khác khi dữ liệu có hình dạng phức tạp hoặc có điểm ngoại lai.
* Xử lý điểm ngoại lai: Loại bỏ điểm ngoại lai: Sử dụng các phương pháp thống kê để xác định và loại bỏ chúng.

## Ứng dụng Kmeans trong xử lý ảnh.

### Ứng dụng và các xử lý ảnh

K-Means Clustering là một thuật toán phổ biến được sử dụng trong việc phân cụm màu sắc nhằm đơn giản hóa hoặc làm nổi bật các màu chủ đạo trong hình ảnh. Ứng dụng của K-means trong phân cụm màu sắc cho phép ta rút gọn số lượng màu bằng cách gom nhóm các màu tương đồng lại với nhau, sau đó đại diện cho từng cụm bằng một màu trung bình.

Trong xử lý ảnh, K-means giúp phân chia các điểm ảnh (pixel) dựa trên các giá trị màu sắc, từ đó nhóm các màu tương tự lại thành các cụm. Kết quả là một hình ảnh với số lượng màu giảm, có thể sử dụng để:

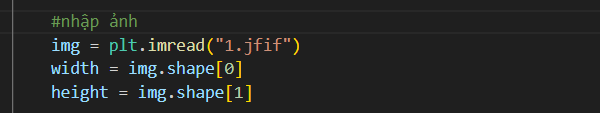
* Giảm dung lượng hình ảnh: bằng cách đại diện mỗi cụm bằng một màu, hình ảnh được nén mà vẫn giữ các đặc trưng chính.
* Làm nổi bật đặc điểm màu: phân cụm màu giúp làm rõ các vùng có màu sắc nổi bật, hỗ trợ trong các ứng dụng nhận diện và phân tích hình ảnh.
* Tạo hiệu ứng thị giác: hình ảnh sau khi phân cụm màu có thể tạo ra các hiệu ứng nghệ thuật bằng cách nhấn mạnh những màu chủ đạo.

Ứng dụng của K-means trong phân cụm màu không chỉ có ý nghĩa trong thị giác máy tính mà còn hỗ trợ trong các lĩnh vực như nhận diện đối tượng, phân đoạn ảnh, và thiết kế đồ họa.

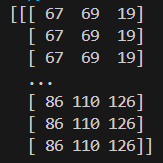
+ Cách thực hiện xử lý ảnh

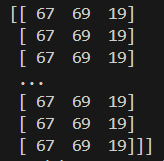
- Đọc ảnh:

Nhập vào 1 ảnh bất kì và khai báo chiều cao, chiều rộng và độ sâu cho ảnh



Ảnh đầu vào có dạng:



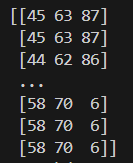


- Chuyển đổi ảnh thành 2D để làm việc với K-means:

Chuyển đổi từ ảnh với 3 giá trị (chiều cao, chiều rộng , độ sâu) thành ảnh với 2 giá trị là (chiều cao \* chiều rộng, 3)

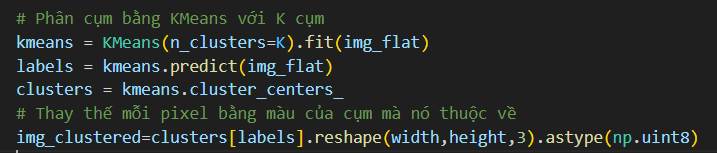


Ảnh 2D sau khi chuyển có dạng:

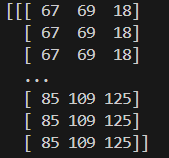


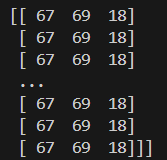
-Phân cụm bằng K-means với K cụm:

Thực hiện cách hàm K-means với xử lý ảnh và thay thế mỗi pixel bằng màu của cụm mà nó thuộc về

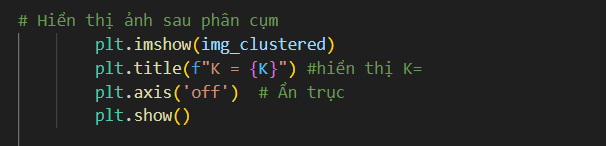


Ảnh sau khi hoàn thành phân cụm có dạng:





-Hiên thị ảnh sau khi phân cụm bằng K-means:



### Kết quả minh họa (làm thêm)

Với ứng dụng này, bắt đầu bằng việc đọc dữ liệu hình ảnh và sử dụng K-means để chia các điểm ảnh thành hai cụm màu (n\_clusters=2). Sau khi huấn luyện, các điểm ảnh trong hình ảnh gốc được gán nhãn dựa trên màu sắc gần nhất từ các cụm. Kết quả cuối cùng là một hình ảnh mới với các màu sắc đại diện cho từng cụm, làm nổi bật sự khác biệt giữa các vùng màu của hình ảnh. Hình ảnh minh họa kết quả như dưới đây.

A tree in a field

Description automatically generated

Hình 3. : Ảnh đầu vào

Kết quả ảnh đầu ra cuối cùng là một hình ảnh đã được đơn giản hóa màu sắc, trong đó toàn bộ các điểm ảnh (pixel) ban đầu được thay thế bằng các màu sắc đại diện cho từng cụm của K-means. Cụ thể:

* Mỗi điểm ảnh của ảnh gốc được gán cho một trong các màu cụm trung bình (cluster center) mà K-means đã tìm ra, dựa trên sự tương đồng về màu sắc.
* Hình ảnh cuối cùng chỉ chứa các màu chủ đạo tương ứng với số cụm màu đã chọn (trong đoạn mã là 2 cụm), tạo ra một hiệu ứng phân vùng rõ rệt giữa các khu vực khác nhau của bức ảnh theo màu sắc

A tree in the middle of a field

Description automatically generated

Hình 3. : Ảnh đầu ra

# CHƯƠNG IV. TỔNG KẾT

## Kết quả đạt được

Báo cáo này đã đạt được những kết quả quan trọng trong việc nghiên cứu và ứng dụng thuật toán K-means vào xử lý ảnh, đồng thời cung cấp nền tảng lý thuyết và kỹ thuật cần thiết để thực hiện dự án. Các kết quả chính đạt được bao gồm:

* Hiểu biết về Python và các thư viện hỗ trợ: Đã trình bày khái quát về ngôn ngữ Python, các thư viện quan trọng như Numpy, Sklearn, Scipy, và Matplotlib. Những thư viện này là nền tảng cho việc triển khai thuật toán và xử lý dữ liệu, hình ảnh một cách hiệu quả.
* Nghiên cứu về hệ màu và lưu trữ dữ liệu: Hiểu rõ hệ màu và các cách lưu trữ dữ liệu ảnh giúp tối ưu hóa việc sử dụng tài nguyên, chuẩn bị tốt cho quá trình phân cụm và xử lý ảnh.
* Nắm bắt lý thuyết K-means và các ứng dụng trong xử lý ảnh: Đã phân tích kỹ thuật hoạt động của K-means, giải thích các thuật ngữ, ý nghĩa và ứng dụng thực tế của thuật toán này, đặc biệt trong việc phân cụm màu sắc cho ảnh.
* Ứng dụng K-means trong xử lý ảnh: Đã triển khai và thử nghiệm thuật toán K-means cho phân cụm màu ảnh, với kết quả đạt được là hình ảnh đơn giản hóa nhưng vẫn giữ được các vùng màu chính. Điều này chứng minh hiệu quả của thuật toán trong việc nén và phân tích hình ảnh.

Báo cáo đã hoàn thành mục tiêu đặt ra, cung cấp kiến thức nền tảng và minh họa rõ ràng hiệu quả của thuật toán K-means trong xử lý và phân tích hình ảnh

## Hạn chế

Dù báo cáo đã đạt được nhiều kết quả tích cực, nhưng vẫn còn một số hạn chế cần lưu ý:

* Giới hạn trong việc lựa chọn số cụm (k): Thuật toán K-means yêu cầu người dùng xác định trước số cụm, điều này có thể dẫn đến kết quả phân cụm không tối ưu nếu lựa chọn số cụm không chính xác. Điều này đặc biệt khó khăn khi làm việc với các hình ảnh phức tạp có nhiều màu sắc.
* Phụ thuộc vào vị trí khởi tạo: K-means có thể cho kết quả khác nhau tùy thuộc vào cách chọn các tâm cụm ban đầu. Việc khởi tạo không chính xác có thể khiến thuật toán bị kẹt trong cực trị cục bộ, làm giảm độ chính xác của kết quả phân cụm.
* Không hiệu quả với các cụm có hình dạng phức tạp: K-means hoạt động tốt nhất khi các cụm có dạng hình cầu và có kích thước tương đồng. Đối với các hình ảnh có cấu trúc phức tạp hoặc các cụm có hình dạng phi tuyến tính, thuật toán có thể không nhận diện đúng các cụm màu.
* Độ nhạy với nhiễu: K-means dễ bị ảnh hưởng bởi các điểm ảnh nhiễu hoặc các điểm ảnh ngoại lệ (outliers), dẫn đến việc phân cụm không chính xác hoặc mất đi màu sắc chủ đạo trong ảnh.
* Tính toán tốn kém cho dữ liệu lớn: Với các hình ảnh có độ phân giải cao, số lượng điểm ảnh rất lớn, làm tăng khối lượng tính toán và thời gian xử lý của thuật toán. Điều này có thể ảnh hưởng đến hiệu suất, đặc biệt khi áp dụng cho những dự án xử lý ảnh quy mô lớn.
* Khả năng phân cụm hạn chế trong xử lý ảnh phức tạp: Trong một số hình ảnh phức tạp, việc phân cụm chỉ dựa vào màu sắc có thể không đủ để tách biệt các đối tượng mong muốn, dẫn đến việc phân vùng không chính xác giữa các đối tượng trong ảnh.

Những hạn chế này có thể được cải thiện trong các nghiên cứu tiếp theo, như thử nghiệm các thuật toán phân cụm khác hoặc tối ưu hóa quy trình khởi tạo và xử lý nhiễu.

# TÀI LIỆU THAM KHẢO

1. Aggarwal, C. C. (2013). *Data mining: Concepts and techniques*. Springer.
2. Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.
3. Bivand, R. S., Pebesma, E., & Gomez-Rubio, V. (2013). *Applied spatial data analysis with R* (2nd ed.). Springer.
4. Bradski, G., & Kaehler, A. (2008). *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media.
5. Duda, R. O., Hart, P. E., & Stork, D. G. (2001). *Pattern classification* (2nd ed.). Wiley.
6. Géron, A. (2019). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems* (2nd ed.). O'Reilly Media.
7. Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference, and prediction* (2nd ed.). Springer.
8. Hunter, J. D. (2007). *Matplotlib: A 2D graphics environment*. *Computing in Science & Engineering, 9*(3), 90–95
9. Lloyd, S. (1982). *Least squares quantization in PCM*. *IEEE Transactions on Information Theory, 28*(2), 129–137.
10. McKinney, W. (2010). *Data structures for statistical computing in Python*. *Proceedings of the 9th Python in Science Conference*, 51–56.
11. Murphy, K. P. (2012). *Machine learning: A probabilistic perspective*. MIT Press.
12. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). *Scikit-learn: Machine learning in Python*. *Journal of Machine Learning Research, 12*, 2825–2830.
13. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., ... & Fei-Fei, L. (2015). *ImageNet large scale visual recognition challenge*. *International Journal of Computer Vision, 115*(3), 211–252.