

**Giảng viên: Ths. Trần Thanh Huân**

**Lớp: 20241IT6094005**

**Nhóm: 02**

**Thành viên:**

1. **Nguyễn Thu Chang - 2022603342**
2. **Lê Thế Cường - 2022603164**
3. **Trần Văn Cường – 2022606203**
4. **Nguyễn Văn Đại - 2022603694**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

**ĐỀ TÀI: Tìm hiểu thuật toán KNN và ứng dụng gợi ý xem phim**

**BÁO CÁO BÀI TẬP LỚN**

**HỌC PHẦN: TRÍ TUỆ NHÂN TẠO**

*Hà Nội, năm 2024*

|  |  |
| --- | --- |
| BỘ CÔNG THƯƠNG  **TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**  -------------------------- | **CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**  **Độc lập – Tự do – Hạnh phúc**  ------------------------ |

**PHIẾU GIAO ĐỀ TÀI**

**BÁO CÁO THỰC NGHIỆM/ THÍ NGHIỆM**

Nhóm thực hiện:  1) Nguyễn Thu Chang 2022603342

                  2) Lê Thế Cường 2022603164

3) Trần Văn Cường 2022606203

4) Nguyễn Văn Đại 2022603694

Lớp học phần: [20241IT6094005](https://qlht.haui.edu.vn/course/view.php?id=33228) Khóa: K17 Khoa Công Nghệ Thông Tin

Ngành học: Công nghệ thông tin

Tên đề tài: Tìm hiểu thuật toán KNN và ứng dụng gợi ý xem phim

Mục đích:

* Tìm hiểu chuyên sâu về thuật toán KNN
* Ứng dụng thuật toán để xây dụng ứng dụng gợi ý xem phim

Yêu cầu:

* Hoàn thành các nhiệm vụ của đề tài
* Sử dụng kỹ năng làm việc nhóm trong quá trình thực hiện báo cáo
* Trình bày báo cáo đúng yêu cầu, sử dụng kỹ năng “Viết báo cáo” khi thực hiện đề tài
* Sử dụng kỹ năng viết tài liệu kỹ thuật và phi kỹ thuật trong phần mở đầu của báo cáo

Kết quả thu được: Bản báo cáo đề tài (bản cứng và bản mềm) và sản phẩm của đề tài

Ngày giao đề tài: 20/11/2024

Ngày hoàn thành: 20/12/2024

Giảng viên hướng dẫn: Trần Thanh Huân

|  |  |
| --- | --- |
|  | Hà Nội, Ngày 24 Tháng 12 Năm 2024  **GIẢNG VIÊN**  ***Trần Thanh Huân*** |
|  |  |

**PHIẾU PHÂN CÔNG NHIỆM VỤ**

Nhóm 02, gồm 4 thành viên

1. Nguyễn Thu Chang
2. Lê Thế Cường (Nhóm trưởng)
3. Trần Văn Cường
4. Nguyễn Văn Đại

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TT** | **Người thực hiện** | **Công việc** | **Kết quả đạt được** | **Nhận xét của GV** |
| **Tuần 1** | | | | |
| 1 | Nguyễn Thu Chang | Tìm hiểu sơ bộ, phân tích bài toán tổng quát | Hoàn thành |  |
| 2 | Lê Thế Cường |
| 3 | Trần Văn Cường | Thu thập dữ liệu bộ phim, thể loại cho bài toán qua các tài liệu tham khảo, github | Hoàn thành |  |
| 4 | Nguyễn Văn Đại |
| **Tuần 2** | | | | |
| 1 | Cả nhóm | Tìm hiểu các thuật toán Machine Learning | Hoàn thành |  |
| 2 | Nguyễn Thu Chang | Tìm hiểu nguyên tắc hoạt động của thuật toán KNN | Hoàn thành |  |
| 3 | Lê Thế Cường | Nêu phương pháp và đối tượng để nghiên cứu | Hoàn thành |  |
| 4 | Trần Văn Cường | Tìm hiểu về ưu điểm của thuật toán KNN | Hoàn thành |  |
| 5 | Nguyễn Văn Đại | Tìm hiểu về nhược điểm của thuật toán KNN | Hoàn thành |  |
| **Tuần 3** | | | | |
| 1 | Nguyễn Thu Chang | Phân tích đề tài và các bài toán áp dụng | Hoàn thành |  |
| 2 | Lê Thế Cường | Xây dụng bộ dữ liệu nhỏ và giải bằng tay | Hoàn thành |  |
| 3 | Trần Văn Cường | Phân tích bộ dữ liệu dành cho máy | Hoàn thành |  |
| 4 | Nguyễn Văn Đại | Cài đặt thuật toán | Hoàn thành |  |
| **Tuần 4** | | | | |
| 1 | Nguyễn Thu Chang | Hoàn thiện báo cáo | Hoàn thành |  |
| 2 | Lê Thế Cường | Kiểm thử và nhận xét | Hoàn thành |  |
| 3 | Trần Văn Cường | Làm slide | Hoàn thành |  |
| 4 | Nguyễn Văn Đại | Hoàn thiện báo cáo | Hoàn thành |  |

|  |  |
| --- | --- |
| BỘ CÔNG THƯƠNG  **TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**  -------------------------- | **CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**  **Độc lập – Tự do – Hạnh phúc**  ------------------------ |

**PHIẾU ĐÁNH GIÁ BÁO CÁO THÍ NGHIỆM/THỰC NGHIỆM**

**IT6094-TRÍ TUỆ NHÂN TẠO**

**I. THÔNG TIN CHUNG**

Người đánh giá: Trần Thanh Huân                Học hàm, học vị: Thạc sĩ

Đơn vị công tác: Khoa CNTT

Tên lớp: 20241IT6094005          Khóa: K17

Họ và tên sinh viên: Nguyễn Thu Chang

Tên nhóm: 02

Tên sản phẩm: Tìm hiểu thuật toán KNN và ứng dụng gợi ý xem phim

**II. ĐÁNH GIÁ**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TT** | **Mục tiêu/chuẩn đầu ra học phần** | **Tiêu chí đánh giá sản phẩm** | **Điểm tối đa** | **Điểm đánh giá** |
| 1 | L3 | Vận dụng lý thuyết TTNT và công cụ phần mềm vào triển khai thực nghiệm. | 10 |  |
| **Tổng** | | | **10** |  |

*Ngày … tháng … năm 2024*

**GIẢNG VIÊN**

**Trần Thanh Huân**

|  |  |
| --- | --- |
| BỘ CÔNG THƯƠNG  **TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**  -------------------------- | **CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**  **Độc lập – Tự do – Hạnh phúc**  ------------------------ |

**PHIẾU ĐÁNH GIÁ BÁO CÁO THÍ NGHIỆM/THỰC NGHIỆM**

**IT6094-TRÍ TUỆ NHÂN TẠO**

**I. THÔNG TIN CHUNG**

Người đánh giá: Trần Thanh Huân                Học hàm, học vị: Thạc sĩ

Đơn vị công tác: Khoa CNTT

Tên lớp: 20241IT6094005          Khóa: K17

Họ và tên sinh viên: Lê Thế Cường

Tên nhóm: 02

Tên sản phẩm: Tìm hiểu thuật toán KNN và ứng dụng gợi ý xem phim

**II. ĐÁNH GIÁ**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TT** | **Mục tiêu/chuẩn đầu ra học phần** | **Tiêu chí đánh giá sản phẩm** | **Điểm tối đa** | **Điểm đánh giá** |
| 1 | L3 | Vận dụng lý thuyết TTNT và công cụ phần mềm vào triển khai thực nghiệm. | 10 |  |
| **Tổng** | | | **10** |  |

*Ngày … tháng … năm 2024*

**GIẢNG VIÊN**

**Trần Thanh Huân**

|  |  |
| --- | --- |
| BỘ CÔNG THƯƠNG  **TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**  -------------------------- | **CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**  **Độc lập – Tự do – Hạnh phúc**  ------------------------ |

**PHIẾU ĐÁNH GIÁ BÁO CÁO THÍ NGHIỆM/THỰC NGHIỆM**

**IT6094-TRÍ TUỆ NHÂN TẠO**

**I. THÔNG TIN CHUNG**

Người đánh giá: Trần Thanh Huân                Học hàm, học vị: Thạc sĩ

Đơn vị công tác: Khoa CNTT

Tên lớp: 20241IT6094005          Khóa: K17

Họ và tên sinh viên: Trần Văn Cường

Tên nhóm: 02

Tên sản phẩm: Tìm hiểu thuật toán KNN và ứng dụng gợi ý xem phim

**II. ĐÁNH GIÁ**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TT** | **Mục tiêu/chuẩn đầu ra học phần** | **Tiêu chí đánh giá sản phẩm** | **Điểm tối đa** | **Điểm đánh giá** |
| 1 | L3 | Vận dụng lý thuyết TTNT và công cụ phần mềm vào triển khai thực nghiệm. | 10 |  |
| **Tổng** | | | **10** |  |

*Ngày … tháng … năm 2024*

**GIẢNG VIÊN**

**Trần Thanh Huân**

|  |  |
| --- | --- |
| BỘ CÔNG THƯƠNG  **TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**  -------------------------- | **CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**  **Độc lập – Tự do – Hạnh phúc**  ------------------------ |

**PHIẾU ĐÁNH GIÁ BÁO CÁO THÍ NGHIỆM/THỰC NGHIỆM**

**IT6094-TRÍ TUỆ NHÂN TẠO**

**I. THÔNG TIN CHUNG**

Người đánh giá: Trần Thanh Huân                Học hàm, học vị: Thạc sĩ

Đơn vị công tác: Khoa CNTT

Tên lớp: 20241IT6094005          Khóa: K17

Họ và tên sinh viên: Nguyễn Văn Đại

Tên nhóm: 02

Tên sản phẩm: Tìm hiểu thuật toán KNN và ứng dụng gợi ý xem phim

**II. ĐÁNH GIÁ**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TT** | **Mục tiêu/chuẩn đầu ra học phần** | **Tiêu chí đánh giá sản phẩm** | **Điểm tối đa** | **Điểm đánh giá** |
| 1 | L3 | Vận dụng lý thuyết TTNT và công cụ phần mềm vào triển khai thực nghiệm. | 10 |  |
| **Tổng** | | | **10** |  |

*Ngày … tháng … năm 2024*

**GIẢNG VIÊN**

**Trần Thanh Huân**

MỤC LỤC

[DANH SÁCH HÌNH ẢNH 11](#_Toc185716249)

[DANH SÁCH CÁC BẢNG BIỂU 12](#_Toc185716250)

[LỜI MỞ ĐẦU 13](#_Toc185716251)

[CHƯƠNG 1: MỘT SỐ THUẬT TOÁN MACHINE LEARNING 14](#_Toc185716252)

[**1.1 Thuật toán Cây quyết định (Decision Trees)** 14](#_Toc185716253)

[**1.2 Phân loại Bayes (Naive Bayes)** 15](#_Toc185716254)

[**1.3 Thuật toán SVM (Support Vector Machine)** 17](#_Toc185716255)

[**1.4 Thuật toán KNN (K-nearest-neighbors)** 20](#_Toc185716256)

[CHƯƠNG 2: XÂY DỰNG ỨNG DỤNG 23](#_Toc185716257)

[**2.1. Phân tích đề tài** 23](#_Toc185716258)

[**2.2. Thuật toán áp dụng** 23](#_Toc185716259)

[**2.3. Xây dựng bộ dữ liệu nhỏ và giải bằng tay** 23](#_Toc185716260)

[**2.4. Bộ dữ liệu dành cho máy** 25](#_Toc185716261)

[**2.5. Cài đặt thuật toán** 27](#_Toc185716262)

[KẾT LUẬN 31](#_Toc185716263)

[**1. Kết quả đạt được** 31](#_Toc185716264)

[**2. Hạn chế** 31](#_Toc185716265)

[TÀI LIỆU THAM KHẢO 32](#_Toc185716266)

# DANH SÁCH HÌNH ẢNH

[Hình 1.1:Ví dụ minh hoạ về mô hình cây quyết định 6](#_Toc185716291)

[Hình 1.2: Định lý Bayes 8](#_Toc185716292)

[Hình 1.3: Biểu đồ minh hoạ SVM 10](#_Toc185716293)

[Hình 1.4. Biểu đồ minh hoạ SVM 10](#_Toc185716294)

[Hình 1.5: Một số cách tính khoảng cách 2 điểm dữ liệu x, y có k thuộc tính 13](#_Toc185716295)

[Hình 1.6: Minh hoạ cách tính khoảng cách Euclidean 13](#_Toc185716296)

[Hình 2.2: Khai báo các thư viện cho chương trình 19](#_Toc185716297)

[Hình 2.3: Đoạn mã lấy dữ liệu từ file csv 20](#_Toc185716298)

[Hình 2.4: Minh hoạ kết quả trả về sau khi hợp nhất dữ liệu 20](#_Toc185716299)

[Hình 2.5: Xây dựng hàm tính khoảng cách 20](#_Toc185716300)

[Hình 2.6: Minh hoạ khoảng cách tính được giữa 2 bản ghi dữ liệu 21](#_Toc185716301)

[Hình 2.7: Xây dựng hàm tìm ra các phim có khoảng cách nhỏ nhất 21](#_Toc185716302)

[Hình 2.8: Kết quả trả về k(5) người dùng có khoảng cách nhỏ nhất 22](#_Toc185716303)

[Hình 2.9: Xây dựng code tìm danh sách các bộ phim 22](#_Toc185716304)

[Hình 2.10. Tìm ra danh sách phim và sắp xếp 22](#_Toc185716305)

[Hình 2.11: Nhập dữ liệu từ bàn phím và kết quả thu được 22](#_Toc185716306)

# DANH SÁCH CÁC BẢNG BIỂU

[Bảng 2.1. Bảng dữ liệu được minh họa việc giải bằng tay 15](#_Toc185541090)

[Bảng 2.2. Bảng dữ liệu userId - movieId 16](#_Toc185541091)

[Bảng 2.3. Bảng dữ liệu thu thập từ MovieLens Dataset 18](#_Toc185541092)

[Bảng 2.4. Ma trận User - Movie 19](#_Toc185541093)

# LỜI MỞ ĐẦU

Trí tuệ nhân tạo tiếng anh là **Artificial intelligence** – viết tắt là **AI**. Chúng ta có thể hiểu đó như là một ngành của khoa học máy tính. Con người lập trình nên nó với mục đích để giúp máy tính có thể thực hiện các hành vi thông minh và xử lý các yêu cầu giống như con người.

Những năm gần đây, AI nổi lên như một bằng chứng của cuộc cách mạng công nghiệp lần thứ tư. Theo đà phát triển của công nghệ,

Ứng dụng trí tuệ nhân tạo luôn là xu hướng công nghệ tương lai mà các hãng công nghệ trên toàn thế giới đua nhau sáng tạo, nó là nền tảng cốt lõi của cuốc cách mạng công nghệ 4.0. Hầu hết mọi nghành công nghiệp đang làm việc với hàm lượng lớn dữ liệu đều nhận ra tầm quan trọng của công nghệ ML. Những cái nhìn sáng suốt từ nguồn dữ liệu này – chủ yếu dạng thời gian thực – sẽ giúp các tổ chức vận hành hiệu quả hơn hoặc tạo lợi thế cạnh tranh so với các đối thủ. Các ứng dụng của ML đã quá quen thuộc với con người: xe tự hành của Google và Tesla, hệ thống tự tag khuôn mặt trên Facebook, hệ thống gợi ý sản phẩm của Amazon, hệ thống gợi ý phim của Netflix…, chỉ là một vài trong vô vàn những ứng dụng của trí tuệ nhân tạo và cụ thể là ML.

Nhằm vận dụng những kiến thức đã học để áp dụng vào thực tế, trong bài tập lớn này chúng em đã nghiên cứu ứng dụng của trí tuệ nhân tạo trong việc gợi ý xem phim và thuật toán được sử dụng đó là thuật toán KNN- (K-Nearest Neighbors) được viết bằng ngôn ngữ Python.

Với sự hướng dẫn, động viên tận tình của thầy Trần Thanh Huân, chúng em đã hiểu được cách thức hoạt động của thuật toán và hoàn thành bài báo cáo này. Vì lượng kiến thức và trình độ chuyên môn có hạn nên trong tài liệu không tránh khỏi những sai sót, nhóm sinh viên chúng em rất mong nhận được sự góp ý từ thầy và bạn đọc để đề tài của chúng em được hoàn thiện hơn.

Nhóm chúng em xin chân thành cảm ơn.

# CHƯƠNG 1: MỘT SỐ THUẬT TOÁN MACHINE LEARNING

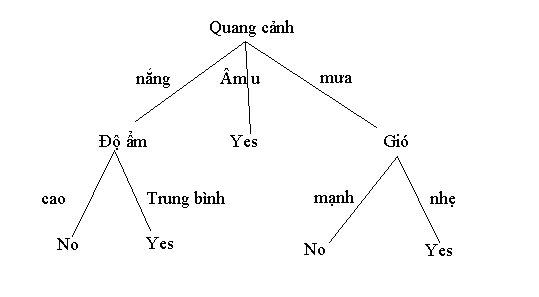
* Cây quyết định (Decision Trees)
* Phân loại bayes
* SVM (Support Vector Machine)
* KNN (K-near-neighbors)

## **1.1 Thuật toán Cây quyết định (Decision Trees)**

Đây là một trong những thuật toán phổ biến nhất được sử dụng hiện nay. Cây quyết định là thuật toán học có giám sát, dùng để phân loại các vấn đề. Thuật toán có thể thực hiện cho cả biến phân loại và biến liên tục. Trong thuật toán này, ta chia dữ liệu thành 2 hoặc nhiều lớp dựa trên phân loại theo các thuộc tính/biến quan trọng.

Đứng dưới góc nhìn thực tế, cây quyết định là một danh sách tối thiểu các câu hỏi dạng yes/no mà người ta phải hỏi, để đánh giá xác suất đưa ra quyết định đúng đắn.

Dưới đây là mô hình ví dụ về cây quyết định:



Hình 1.:Ví dụ minh hoạ về mô hình cây quyết định

Ví dụ như ví dụ trên: Cây quyết định dữ liệu Quang cảnh.

Nhìn từ cây quyết định trên, ta có thể rút ra được kết luận: Nếu trời nắng & độ ẩm trung bình thì người chơi sẽ quyết định “Yes”.

Tuy nhiên, ta cần chú ý tới vấn đề [Overfitting](https://trituenhantao.io/tu-dien-thuat-ngu/overfitting/) trong thuật toán này. Để giảm hiện tượng này, ta thường dùng phương pháp cắt tỉa cây.

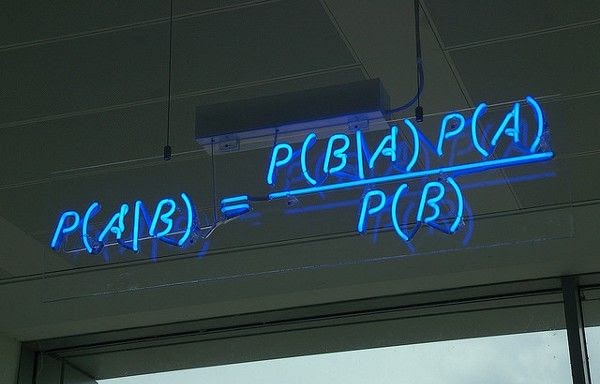
Với [Overfitting](https://trituenhantao.io/tu-dien-thuat-ngu/overfitting/) là hiện tượng mô hình ghi nhớ quá tốt dữ liệu huấn luyện và phụ thuộc vào nó, việc này khiến cho mô hình không thể tổng quát hóa các quy luật để hoạt động với dữ liệu chưa từng được chứng kiến.

* Ưu điểm:
* Mô hình sinh ra các quy tắc dễ hiểu cho người đọc, tạo ra bộ luật với mỗi nhánh lá là một luật của cây.
* Dữ liệu đầu vào có thể là là dữ liệu missing, không cần chuẩn hóa hoặc tạo biến giả**.**
* Có thể làm việc với cả dữ liệu số và dữ liệu phân loại**.**
* Có thể xác thực mô hình bằng cách sử dụng các kiểm tra thống kê.
* Có khả năng là việc với dữ liệu lớn.
* Nhược điểm:
* Mô hình cây quyết định phụ thuộc rất lớn vào dữ liệu. Thậm chí, với một sự thay đổi nhỏ trong bộ dữ liệu, cấu trúc mô hình cây quyết định có thể thay đổi hoàn toàn.
* Cây quyết định hay gặp vấn đề overfitting.

## **1.2 Phân loại Bayes (Naive Bayes)**

Thuật toán phân loại Naive Bayes là một nhóm các phân loại xác suất đơn giản dựa trên định lý Bayes giả định về việc độc lập giữa các thuộc tính. Ngay cả khi, các thuộc tính này có sự tương quan với nhau thì phương pháp này vẫn xem các thuộc tính là độc lập với nhau.

Định lý Bayes cho phép tính xác suất xảy ra của một sự kiện ngẫu nhiên A khi biết sự kiện liên quan B đã xảy ra. Xác suất này được ký hiệu là P(A|B), và đọc là “xác suất của A nếu có B”. Đại lượng này được gọi xác suất có điều kiện hay xác suất hậu nghiệm vì nó được rút ra từ giá trị được cho của B hoặc phụ thuộc vào giá trị đó.



Hình 1.: Định lý Bayes

Theo định lí Bayes, P(A|B) sẽ phụ thuộc vào 3 yếu tố:

* Xác suất xảy ra A của riêng nó, không quan tâm đến B. Kí hiệu là P(A).
* Xác suất xảy ra B của riêng nó, không quan tâm đến A. Kí hiệu là P(B).
* Xác suất xảy ra B khi biết A xảy ra. Kí hiệu là P(B|A). Đại lượng này gọi là khả năng xảy ra B khi biết A đã xảy ra.

Thuật toán này được áp dụng trong một số bài toán như:

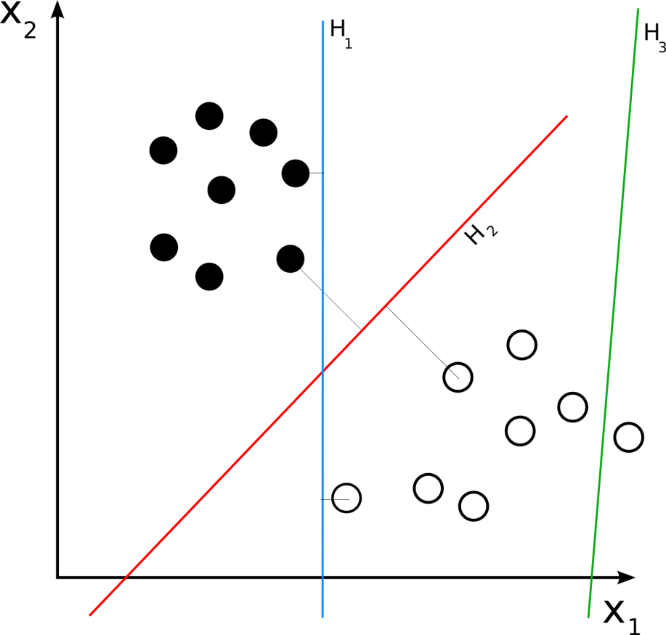
* Đánh dấu một email là spam hay không.
* Phân loại bài viết tin tức thuộc lĩnh vực công nghệ, chính trị hay thể thao.
* Kiểm tra một đoạn văn bản mang cảm xúc tích cực hay tiêu cực.
* Sử dụng cho các phần mềm nhận diện khuôn mặt.
* Ưu điểm:
* Giả định độc lập: hoạt động tốt cho nhiều bài toán/miền dữ liệu và ứng dụng.Đơn giản nhưng đủ tốt để giải quyết nhiều bài toán như phân lớp văn bản, lọc spam,…
* Cho phép kết hợp tri thức tiền nghiệm (prior knowledge) và dữ liệu quan sát được (observed data). Tốt khi có sự chênh lệch số lượng giữa các lớp phân loại.
* Huấn luyện mô hình (ước lượng tham số) dễ và nhanh.
* Nhược điểm:
* Giả định độc lập (ưu điểm cũng chính là nhược điểm): hầu hết các trường hợp thực tế trong đó có các thuộc tính trong các đối tượng thường phụ thuộc lẫn nhau.
* Vấn đề zero (đã nêu cách giải quyết ở phía trên).
* Mô hình không được huấn luyện bằng phương pháp tối ưu mạnh và chặt chẽ. Tham số của mô hình là các ước lượng xác suất điều kiện đơn lẻ. Không tính đến sự tương tác giữa các ước lượng này.

## **1.3 Thuật toán SVM (Support Vector Machine)**

SVM là phương pháp phân loại trong đó dữ liệu thô của bạn sẽ được biểu diễn trên không gian n chiều (n-số thuộc tính). Thông qua không gì biểu diễn diễn dự liệu đó, ta có thể thực hiện phân loại dữ liệu.

**Ý tưởng của SVM**: Tìm một siêu phẳng (hyper lane) phân chia "tốt nhất" tập dữ liệu. Siêu phẳng này sẽ chia không gian thành các miền khác nhau và mỗi miền sẽ chứa một loại dữ liệu.

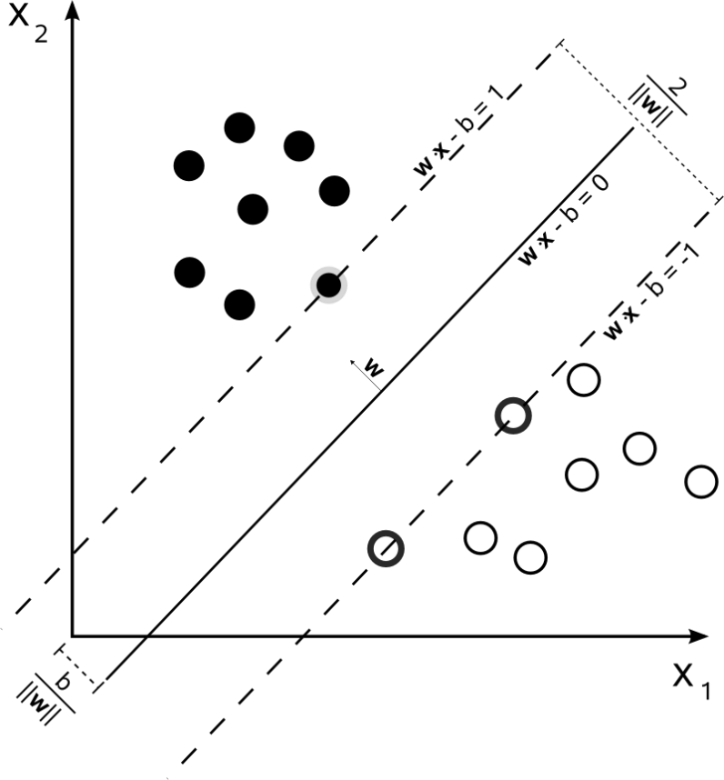
Ví dụ: cho một tập các điểm thuộc 2 loại trong môi trường N chiều, SVM cố gắng tìm ra N-1 mặt phẳng để phân tách các điểm đó thành 2 nhóm.



Hình 1.: Biểu đồ minh hoạ SVM

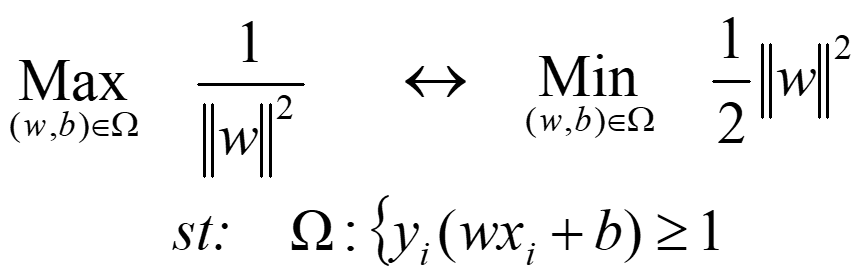
**Siêu phẳng** được biểu diễn bằng hàm số <W.X> = b (W và X là các vector <W.X> là tích vô) Hay W^T=b (W^T là ma trận chuyễn vị)

**Phân chia tốt nhất** là khoảng cách từ các đối tượng gần nhất tới siêu phẳng là cực đại (còn gọi là khoảng cách lề cực đại)

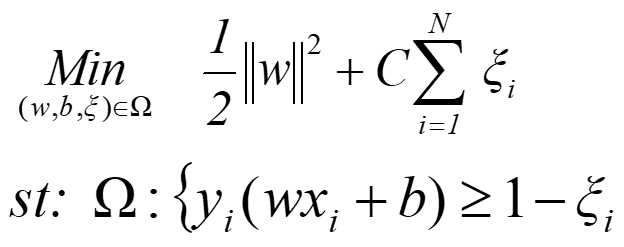


Hình 1.. Biểu đồ minh hoạ SVM

Mô hình SVM lề cứng:



Mô hình SVM lề mềm:



SVM thực hiện giải quyết rất nhiều dạng bài toán như hiển thị quảng cáo, phát hiện giới tính dựa trên hình ảnh, phân loại hình ảnh có quy mô lớn …

* Ưu điểm:
* Xử lý trên không gian số chiều cao: SVM là một công cụ tính toán hiệu quả trong không gian chiều cao, trong đó đặc biệt áp dụng cho các bài toán phân loại văn bản và phân tích quan điểm nơi chiều có thể cực kỳ lớn.
* Tiết kiệm bộ nhớ: Do chỉ có một tập hợp con của các điểm được sử dụng trong quá trình huấn luyện và ra quyết định thực tế cho các điểm dữ liệu mới nên chỉ có những điểm cần thiết mới được lưu trữ trong bộ nhớ khi ra quyết định.
* Tính linh hoạt - phân lớp thường là phi tuyến tính. Khả năng áp dụng Kernel mới cho phép linh động giữa các phương pháp tuyến tính và phi tuyến tính từ đó khiến cho hiệu suất phân loại lớn hơn.
* Nhược điểm:
* Bài toán số chiều cao: Trong trường hợp số lượng thuộc tính (p) của tập dữ liệu lớn hơn rất nhiều so với số lượng dữ liệu (n) thì SVM cho kết quả khá tồi.
* Chưa thể hiện rõ tính xác suất: Việc phân lớp của SVM chỉ là việc cố gắng tách các đối tượng vào hai lớp được phân tách bởi siêu phẳng SVM. Điều này chưa giải thích được xác suất xuất hiện của một thành viên trong một nhóm là như thế nào.

## **1.4 Thuật toán KNN (K-nearest-neighbors)**

K-nearest neighbors là thuật toán học máy có giám sát, đơn giản và dễ triển khai. Thường được dùng trong các bài toán phân loại và hồi quy.

**Bài toán đặt ra:** Giả sử bạn có điểm của một môn học nhưng bạn không biết thuộc loại nào (Giỏi, khá, trung bình, yếu) và không biết bất kỳ quy tắc nào để phân loại cả.

Có một cách giải quyết là bạn phải đi khảo sát những người xung quanh. Để biết điểm của mình thuộc loại nào thì bạn phải đi hỏi những người có điểm gần số điểm của mình nhất. Giả sử trong lớp có 50 người, bạn khảo sát 5 người gần điểm mình nhất và được dữ liệu như sau:

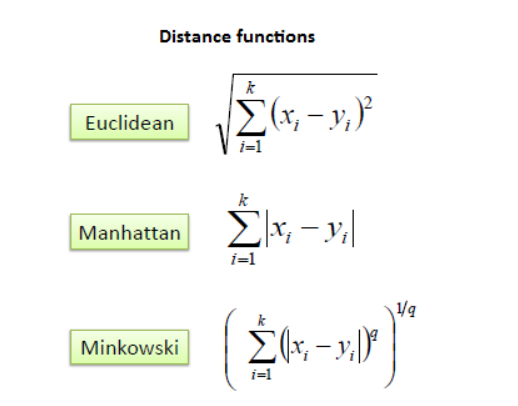
Điểm của bạn: 7

Điểm bạn của bạn:

* 7.1=>Khá
* 7.2=>Khá
* 6.7=>Khá
* 6.6=>Khá
* 6.4=>Trung bình

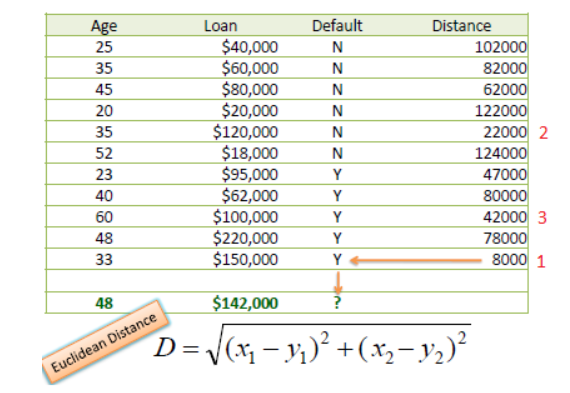
Qua kết quả trên thì bạn sẽ đoán được là mình thuộc loại nào. Với cách này chúng ta có thể phân biệt loại dữ liệu 1 chiều bằng. Nếu ta khảo sát càng nhiều, càng rộng thì dự đoán đưa ra càng chính xác.

**Ý tưởng của KNN**: Thuật toán KNN cho rằng dữ liệu tương quan nhau sẽ tồn tại gần nhau tròn một không gian, từ đó công việc của chúng ta là sẽ tìm k điểm gần với dữ liệu cần kiểm tra nhất. Việc tìm khoảng cách giữa hai điểm cũng có nhiều công thức có thể sử dụng tùy trường hợp mà ta lựa chọn cho phù hợp. Đây là 3 cách cơ bản để tính khoảng cách 2 điểm dữ liệu x, y có k thuộc tính:



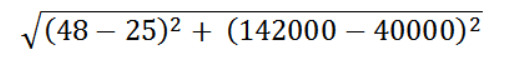
Hình 1.: Một số cách tính khoảng cách 2 điểm dữ liệu x, y có k thuộc tính

Ví dụ chúng ta có dữ liệu là tuổi, khoản vay và khả năng vỡ nợ như hình sau:



Hình 1.: Minh hoạ cách tính khoảng cách Euclidean

Dữ liệu cần phân loại của chúng ta là {age: 48, loan:142000}. Đây là dữ liệu 2 chiều và chúng ta cần dự đoán người này thuộc nguy cơ vỡ nợ hay không. Chúng ta sé dùng một cách khá phổ biến để tính khoảng cách và Euclidean. Ví dụ ở hàng đầu tiên khoảng cách sẽ được tính:



Thực hiện tương tự, ta sẽ tính được khoảng cahcs ở cột Distance, từ đó chọn ra k = 3 khoảng cách nhỏ nhất (gần với dữ liệu nào nhất). Với 3 khoảng cách này chúng ta nhận được 3 label là (Yes, No, Yes). Trong 3 label này Yes xuất hiện nhiều hơn nên chúng ta sẽ đưa ra dự đoán người này có khả năng vỡ nợ.

* Ưu điểm:
* Là một thuật toán đơn giản để hiểu và giải thích.
* Rất hữu ích cho dữ liệu phi tuyến vì không có giả định về dữ liệu trong thuật toán này.
* Là một thuật toán linh hoạt vì chúng ta có thể sử dụng nó để phân loại cũng như hồi quy.
* Có độ chính xác tương đối cao nhưng có nhiều mô hình học có giám sát tốt hơn KNN.
* Nhược điểm:
* Về mặt tính toán, nó là một thuật toán hơi tốn kém vì nó lưu trữ tất cả các dữ liệu huấn luyện.
* Yêu cầu bộ nhớ lưu trữ cao so với các thuật toán học có giám sát khác.
* Dự đoán chậm trong trường hợp N lớn.
* Rất nhạy cảm với quy mô dữ liệu cũng như các tính năng không liên quan.
* Các ứng dụng của KNN:
* Hệ thống ngân hàng: KNN có thể được sử dụng trong hệ thống ngân hàng để dự đoán một cá nhân có phù hợp để phê duyệt khoản vay không? Cá nhân đó có các đặc điểm giống với cá nhân mặc định không.
* Tính toán xếp hạng tín dụng: Các thuật toán KNN có thể được sử dụng để xếp hạng tín dụng của một cá nhân bằng cách so sánh với những người có đặc điểm tương tự .
* Chính trị: Với sự trợ giúp của thuật toán KN, chúng ta có thể phân loại cử tri tiềm năng thành nhiều lớp khác nhau như “Sẽ bỏ phiếu”, “Sẽ không bỏ phiếu”, “Sẽ bỏ phiếu cho Đảng ‘Đại hội’,” Sẽ bỏ phiếu cho Đảng ‘BJP’.
* Các lĩnh vực khác mà thuật toán KNN có thể được sử dụng là nhận dạng giọng nói, phát hiện chữ viết tay, nhận dạng hình ảnh và nhận dạng video.

# CHƯƠNG 2: XÂY DỰNG ỨNG DỤNG

## **2.1. Phân tích đề tài**

Hiện nay, thói quen giải trí, đặc biệt là xem phim, đang ngày càng trở thành một nhu cầu thiết yếu. Tuy nhiên, với số lượng phim ngày càng tăng và sự đa dạng về thể loại, người dùng thường gặp khó khăn trong việc lựa chọn bộ phim phù hợp với sở thích và tâm trạng của mình. Hiểu được nhu cầu cần thiết đó, nhóm em đã quyết định xây dựng bài toán "ứng dụng gợi ý xem phim" được đề xuất nhằm xây dựng một công cụ thông minh, giúp người dùng tìm kiếm và lựa chọn phim một cách nhanh chóng và hiệu quả.

## **2.2. Thuật toán áp dụng**

Bước 1: Xây dựng Rating Matrix từ giữ liệu giao dịch

Bước 2: Giả sử cần đưa ra khuyến nghị cho khách hàng Ui: Tìm ra k khách hàng GIỐNG với Ui nhất, dựa trên Rating Matrix

Bước 3: Dựa vào các Rate Values của k người xem tìm được, ước lượng các giá trị Rate còn thiếu của người xem Ui, các giá trị mới tính được gọi là Rate giả.

Bước 4: Sắp xếp các giá trị Rate giả của Ui và đưa khuyến nghị cho Ui

## **2.3. Xây dựng bộ dữ liệu nhỏ và giải bằng tay**

Bảng 2.1. Bảng dữ liệu được minh họa việc giải bằng tay

|  |  |
| --- | --- |
| movieId | title |
| M1 | The Matrix |
| M2 | The Godfather |
| M3 | Fight Club |
| M4 | Schindler's List |
| M5 | Pulp Fiction |
| M6 | Inception |
| M7 | Forrest Gump |

Bảng 2.2. Bảng dữ liệu userId - movieId

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| MovieId  UserId | M1 | M2 | M3 | M4 | M5 | M6 | M7 |
| U1 | 3 | 2 | 5 |  |  | 5 | 6 |
| U2 | 2 |  | 1 | 3 | 2 |  |  |
| U3 |  | 1 | 5 | 5 |  | 5 | 2 |
| U4 | 5 | 9 |  | 9 | 3 |  |  |
| U5 |  | 1 | 4 |  | 7 |  | 6 |

Ví dụ: Đưa ra gợi ý phim phù hợp cho UserId U2:

Giả thiết các giá trị ô trống là 0, áp dụng công thức tính khoảng cách Euclidean:

d(U2,U1) =

d(U2,U3)=

d(U2,U4)=

d(U2,U5)=

3 láng giềng gần nhất với U2: U3, U5, U1

Tính các Rate giả của 102 bằng các giá trị tương ứng của 101, 103, 105.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| MovieId  UserId | M1 | M2 | M3 | M4 | M5 | M6 | M7 |
| U1 | 3 | 2 | 5 |  |  | 5 | 6 |
| U2 | 2 | 1 | 1 | 3 | 2 | 5 | 6 |
| U3 |  | 1 | 5 | 5 |  | 5 | 2 |
| U4 | 5 | 9 |  | 9 | 3 |  |  |
| U5 |  | 1 | 4 |  | 7 |  | 6 |

Đưa ra khuyến nghị cho U2 theo thứ tự: M7, M6, M2

## **2.4. Bộ dữ liệu dành cho máy**

Giới thiệu bộ dữ liệu: Trong đề tài "Tìm hiểu thuật toán KNN và ứng dụng gợi ý xem phim," bộ dữ liệu được sử dụng gồm thông tin người dùng, phim đã xem, điểm đánh giá cũng như thể loại phim. Dữ liệu này được thu thập từ nguồn **MovieLens Dataset**, hoặc được giả lập tạo ra dựa trên thông tin cần thiết.

Bộ dữ liệu bao gồm:

1. UserID: ID người dùng.
2. MovieID: ID phim.
3. Rating: Điểm đánh giá phim (đặt trong khoảng từ 1 đến 5).
4. Genres: Thể loại phim (Action, Comedy, Drama, Horror, ...).

Dữ liệu bao gồm **610 người dùng** và **9724 bộ phim**, được biểu diễn dưới dạng ma trận User-Movie. Đây là bộ dữ liệu đủ lớn để huấn luyện và đánh giá mô hình KNN.

Cấu trúc bộ dữ liệu (minh họa):

Bảng 2.3. Bảng dữ liệu thu thập từ MovieLens Dataset

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 101 | 5 | Action, Drama |
| 1 | 102 | 4 | Comedy |
| 1 | 103 | 0 | Horror |
| 1 | 104 | 0 | Drama, Romance |
| 2 | 101 | 3 | Action, Drama |
| 2 | 102 | 0 | Comedy |
| 2 | 103 | 2 | Horror |
| 2 | 104 | 0 | Drama, Romance |
| 3 | 101 | 0 | Action, Comedy |
| 3 | 102 | 0 | Comedy |
| 3 | 103 | 0 | Horror |
| 3 | 104 | 5 | Drama, Romance |
| 4 | 101 | 4 | Action, Comedy |
| 4 | 102 | 5 | Comedy |
| 4 | 103 | 3 | Horror |
| 4 | 104 | 0 | Drama, Romance |
| 5 | 101 | 2 | Action, Drama |
| 5 | 102 | 3 | Comedy |
| 5 | 103 | 4 | Horror |
| 5 | 104 | 5 | Drama, Romance |

Ma trận User - Movie

Ma trận dữ liệu dưới dạng User-Movie, trong đó các ô đại diện điểm đánh giá phim:

Bảng 2.4. Ma trận User - Movie

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **User/Movie** | **101** | **102** | **103** | **104** |
| 1 | 5 | 4 | 0 | 0 |
| 2 | 3 | 0 | 2 | 0 |
| 3 | 0 | 0 | 0 | 5 |

Tiền xử lý dữ liệu

1. **Chuẩn hóa dữ liệu**:

* Gán cho những bộ phim mà người dùng chưa đánh giá là 0
* Từ file dữ liệu tạo thành các bảng với tiêu đề hàng là “userId”, tiêu đề cột là “movieId” và giá trị trong bảng là đánh giá của người dùng cho các bộ phim

Mô tả cách sử dụng dữ liệu

* Thuật toán KNN tính khoảng cách giữa người dùng cần gợi ý phim và các người dùng khác dựa trên ma trận User-Movie.
* Tìm ra **k người dùng** có sự tương đồng với người dùng được chọn
* Tìm ra những bộ phim mà người dùng hiện tại chưa xem và rate giả của chúng
* Sắp xếp các rate giả và gợi ý cho người dùng.

Kết quả mong muốn

Bộ dữ liệu sẽ giúp huấn luyện thuật toán KNN để gợi ý phim chính xác, từ đó tăng trải nghiệm cho người dùng

## **2.5. Cài đặt thuật toán**

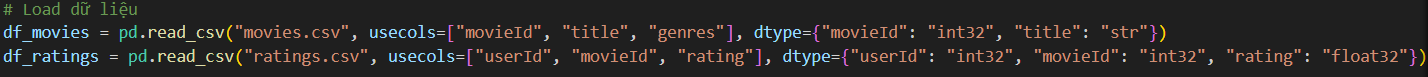
**Bước 1**: Khai báo các thư viện cần thiết

A black screen with white text

Description automatically generated

Hình 2.2: Khai báo các thư viện cho chương trình

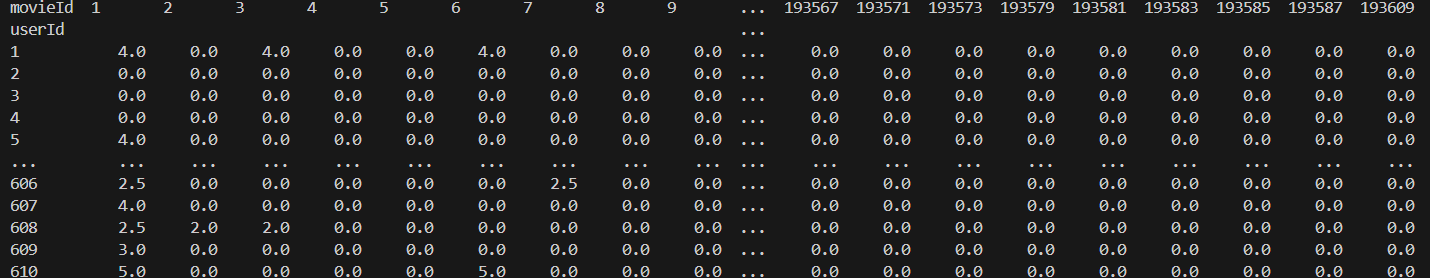
**Bước 2**: Tạo hàm đọc dữ liệu vào từ file csv



Hình 2.3: Đoạn mã lấy dữ liệu từ file csv

Dùng thư viện pandas để tải dữ liệu từ 2 file là ratings.csv và movies.csv và chuyển đổi nó thành các DataFrame và chuẩn hóa kiểu dữ liệu để thuận tiện cho việc tính khoảng cách.

Thao tác tiền xử lí dữ liệu gồm: gọi 2 hàm đọc dữ liệu và gán các data frame vào các biến, sau đó thực hiện hợp nhất 2 bảng đọc được thành 1 ma trận gồm tiêu đề cột là các “movieId” và tiêu đề hàng là “userId”. Thực hiện thay thế các giá trị NaN (bộ phim mà các người dùng chưa đánh giá hoặc chưa xem) bằng 0. Kết quả thu được như hình sau:



Hình 2.4: Minh hoạ kết quả trả về sau khi hợp nhất dữ liệu

**Bước 3**: Xây dựng hàm tính khoảng cách

A black background with text

Description automatically generated

Hình 2.5: Xây dựng hàm tính khoảng cách

Hàm thực hiện việc tính toán dữ liệu 2 điểm truyền vào bằng công thức Euclidean. Gọi thư viện numpy để tính hiệu giữa 2 vector rồi bình phương để tính khoảng cách  
 Ví dụ hàm trả về khoảng cách của 2 dòng dữ liệu:

A black background with white numbers and dots

Description automatically generated

A black background with white numbers

Description automatically generated



Hình 2.6: Minh hoạ khoảng cách tính được giữa 2 bản ghi dữ liệu

**Bước 4**: Xây dựng hàm tìm k điểm dữ liệu gần nhất

A computer code with colorful text

Description automatically generated

Hình 2.7: Xây dựng hàm tìm ra các phim có khoảng cách nhỏ nhất

Hàm này sẽ duyệt qua tất cả các giá trị trong trainSet, tính khoảng cách giữa điểm truyền vào với những điểm trong tập dữ liệu ban đầu. Kết quả của vòng lặp này là một list các dictionary gồm id của phim và khoảng cách đến điểm đó. Tiếp theo chúng ta sẽ sắp xếp tăng dần list này với giá trị so sánh là khoảng cách. Vì kết quả chúng ta chỉ cần biết là số bộ phim gần với bộ phim được chọn nhất nên sắp xếp tăng dần theo khoảng cách và cuối cùng là trả về k điểm dữ liệu đầu tiên của list (nhỏ nhất).

Mình hoạ hàm sắp xếp, tìm 5 người dùng có khoảng cách đến người dùng hiện tại là nhỏ nhất rồi trả về id và khoảng cách của người dùng đó đó:



Hình 2.8: Kết quả trả về k(5) người dùng có khoảng cách nhỏ nhất

**Bước 5**: Xây dựng code tìm danh sách các bộ phim người dùng hiện tại chưa xem và đánh giá của k người dùng gần nhất

A computer code on a black background

Description automatically generated

Hình 2.9: Xây dựng code tìm danh sách các bộ phim

**Bước 6**: Từ danh sách các bộ phim gợi ý, chọn ra 10 bộ phim có rate giả cao nhất

A computer screen with colorful text

Description automatically generated

Hình 2.10. Tìm ra danh sách phim và sắp xếp

**Bước 7**: Nhập dữ liệu từ bàn phím và hiển thị kết quả lên màn hình

A screenshot of a computer

Description automatically generated

Hình 2.11: Nhập dữ liệu từ bàn phím và kết quả thu được

# KẾT LUẬN

## **1. Kết quả đạt được**

* Sự hiểu biết về thuật toán KNN cơ bản tương đối tốt.
* Hiểu biết thêm về AI, ứng dụng của của Machine Learning vào đời sống công nghệ hiện đại.
* Làm quen ngôn ngữ lập trình Python
* Áp dụng được kiến thức đã học vào bài toán thực tiễn (ứng dụng gợi ý phim)

## **2. Hạn chế**

* Thuật toán KNN còn phụ thuộc nhiều vào hệ số K.
* Kết quả đưa ra có sự thay đổi (vì các điểm xét lấy ngẫu nhiên).

# TÀI LIỆU THAM KHẢO

[1]. Giáo trình trí tuệ nhân tạo - Trường Đại học Công nghiệp Hà Nội.

[2]. [Machine learning là gì? Tổng quan về machine learning (nguyenvanhieu.vn)](https://nguyenvanhieu.vn/machine-learning-la-gi/#:~:text=Machine%20learning%20l%C3%A0%20g%C3%AC%3F%20Machine%20learning%20l%C3%A0%20m%E1%BB%99t,t%C3%ADnh%20h%E1%BB%8Dc%20%C4%91%E1%BB%83%20l%C3%A0m%20c%C3%B4ng%20vi%E1%BB%87c%20g%C3%AC%20%C4%91%C3%B3%21).

[3]. [Thuật Toán K-Nearest Neighbors (KNN) Siêu Cơ Bản (codelearn.io)](https://codelearn.io/sharing/thuat-toan-k-nearest-neighbors-knn).

[4]. Tài liệu điện tử trên internet.