**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC ĐẠI NAM**

Logo, company name

Description automatically generated

**BÀI TẬP LỚN**

**TÊN HỌC PHẦN: NHẬP MÔN HỌC MÁY**

**ĐỀ TÀI: PHÂN LOẠI CHỦ ĐỀ BÀI BÁO**

**Giáo viên hướng dẫn: Trần Thu Trang**

**Sinh viên thực hiện:**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Mã sinh viên** | **Họ và tên** | **Lớp** |
| 1 | 1771020179 | Lê Thiên Dũng | CNTT 17-12 |
| 2 | 1771020192 | Giàng A Dụng | CNTT 17-12 |

**Hà Nội, năm 2025**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC ĐẠI NAM**

Logo, company name

Description automatically generated

**BÀI TẬP LỚN**

**TÊN HỌC PHẦN: NHẬP MÔN HỌC MÁY**

**ĐỀ TÀI: PHÂN LOẠI CHỦ ĐỀ BÀI BÁO**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Mã Sinh Viên** | **Họ và Tên** | **Ngày Sinh** | **Điểm** | |
| **Bằng Số** | **Bằng Chữ** |
| **1** | **1771020179** | **Lê Thiên Dũng** | **17/10/2005** |  |  |
| **2** | **1771020192** | **Giàng A Dụng** | **11/08/2005** |  |  |

**CÁN BỘ CHẤM THI**

**Hà Nội, năm 2025**

**LỜI NÓI ĐẦU**

Trong kỷ nguyên bùng nổ thông tin hiện nay, số lượng bài báo khoa học được công bố trên toàn thế giới ngày càng tăng nhanh, đặc biệt là trong các kho dữ liệu mở như **arXiv**. Việc tìm kiếm, phân loại và khai thác thông tin từ các nguồn dữ liệu khổng lồ này đóng vai trò vô cùng quan trọng, giúp các nhà nghiên cứu, sinh viên và tổ chức khoa học tiếp cận tri thức một cách nhanh chóng và hiệu quả. Tuy nhiên, khối lượng dữ liệu khổng lồ cùng sự đa dạng về lĩnh vực đã đặt ra yêu cầu cấp thiết về các phương pháp tự động hóa trong xử lý và phân loại văn bản.

Trước bối cảnh đó, **học máy (Machine Learning)** trở thành công cụ mạnh mẽ, hỗ trợ tự động hóa quá trình phân tích dữ liệu văn bản, trong đó có việc phân loại chủ đề bài báo khoa học. Bằng cách khai thác các kỹ thuật như **tiền xử lý ngôn ngữ tự nhiên (NLP), biểu diễn văn bản bằng vector**, và **thuật toán phân loại,** chúng ta có thể xây dựng những mô hình có khả năng nhận diện và sắp xếp bài báo vào đúng lĩnh vực nghiên cứu. Điều này không chỉ tiết kiệm thời gian, công sức mà còn nâng cao tính chính xác, mở ra hướng đi mới trong khai phá tri thức khoa học.

Trong khuôn khổ môn học **Nhập môn học máy,** nhóm chúng em lựa chọn đề tài “Phân loại chủ đề bài báo” nhằm áp dụng các kiến thức đã học vào một bài toán thực tế. Qua đó, chúng em mong muốn tìm hiểu sâu hơn về cách thức áp dụng các thuật toán học máy vào xử lý văn bản, rèn luyện kỹ năng thực hành trên dữ liệu thực tế và củng cố tư duy nghiên cứu khoa học. Báo cáo này trình bày quá trình tìm hiểu, xây dựng và đánh giá mô hình phân loại, đồng thời thể hiện sự nỗ lực của nhóm trong việc vận dụng lý thuyết vào thực tiễn.

MỤC LỤC

[**LỜI NÓI ĐẦU 3**](#_Toc209745484)

[**MỤC LỤC HÌNH ẢNH 6**](#_Toc209745485)

[**DANH MỤC CÁC TỪ VIẾT TẮT 7**](#_Toc209745486)

[**chương 1 giới thiệu đề tài 8**](#_Toc209745487)

[**1.1. Giới thiệu đề tài 8**](#_Toc209745488)

[**1.2. Mô tả dữ liệu 8**](#_Toc209745489)

[***1.2.1. Thông tin chung của dữ liệu 8***](#_Toc209745490)

[***1.2.2. Mô tả dữ liệu 9***](#_Toc209745491)

[***1.2.3. Đặc điểm dữ liệu 9***](#_Toc209745492)

[**1.3. Trực quan hóa dữ liệu 10**](#_Toc209745493)

[**CHƯƠNG 2 SƠ SỞ LÝ THUYẾT 11**](#_Toc209745494)

[**2.1. Thuật toán K-NEAREST NEIGHBOR (KNN) 11**](#_Toc209745495)

[***2.1.1. Khái niệm 11***](#_Toc209745496)

[***2.1.2. Ý tưởng chính 11***](#_Toc209745497)

[***2.1.3 Các công thức tính toán khoảng cách 11***](#_Toc209745498)

[***2.1.4. Cách chọn tham số K 12***](#_Toc209745499)

[**2.2. Thuật toán Naïve Bayes 12**](#_Toc209745500)

[***2.2.1. Khái niệm 12***](#_Toc209745501)

[***2.2.2. Ý tưởng chính 12***](#_Toc209745502)

[***2.2.3. Công thức 12***](#_Toc209745503)

[***2.2.4. Các biến thể của Naïve Bayes 13***](#_Toc209745504)

[**2.3. Thuật toán cây quyết định (Decision Tree) 13**](#_Toc209745505)

[***2.3.1. Khái niệm 13***](#_Toc209745506)

[***2.3.2. Ý tưởng chính 14***](#_Toc209745507)

[***2.3.3. Các chỉ số chọn thuộc tính 14***](#_Toc209745508)

[**2.4. Thuật toán K-Means 15**](#_Toc209745509)

[***2.4.1. Khái niệm 15***](#_Toc209745510)

[***2.4.2. Ý tưởng chính 15***](#_Toc209745511)

[***2.4.3. Các công thức 16***](#_Toc209745512)

[**CHƯƠNG 3 thực nghiệm và kết quả 17**](#_Toc209745513)

[**3.1. Tiền xử lý dữ liệu 17**](#_Toc209745514)

[**3.2. Chuẩn hoá dữ liệu 19**](#_Toc209745515)

[**3.3. Các mô hình học máy áp dụng 20**](#_Toc209745516)

[**3.4. Huấn luyện mô hình thuật toán 21**](#_Toc209745517)

[***3.4.1. Thuật toán KNN 21***](#_Toc209745518)

[***3.4.2. Thuật toán Naïve Bayes 22***](#_Toc209745519)

[***3.4.3. Thuật toán cây quyết định 23***](#_Toc209745520)

[***3.4.4. Thuật toán K-Mean 24***](#_Toc209745521)

[**3.5. Kết quả thực nghiệm 26**](#_Toc209745522)

[**3.6. Đánh giá và nhận xét 26**](#_Toc209745523)

[**KẾT LUẬN 28**](#_Toc209745524)

[**1. Kết quả đạt được 28**](#_Toc209745525)

[**2. Nhược điểm 28**](#_Toc209745526)

[**3. Hướng phát triển 28**](#_Toc209745527)

[**TÀI LIỆU THAM KHẢO 29**](#_Toc209745528)

# MỤC LỤC HÌNH ẢNH

[Hình 1: Kích thước dữ liệu 8](#_Toc209745468)

[Hình 2: Làm sạch dữ liệu 17](#_Toc209745469)

[Hình 3: Dữ liệu trước khi được làm sạch 18](#_Toc209745470)

[Hình 4: Dữ liệu sau khi được làm sạch 18](#_Toc209745471)

[Hình 5: Chuẩn hoá dữ liệu 19](#_Toc209745472)

[Hình 6: Dữ liệu sau khi được chuẩn hoá 20](#_Toc209745473)

[Hình 7: Vector hoá các mô hình 21](#_Toc209745474)

[Hình 8: Kết quả thuật toán KNN 22](#_Toc209745475)

[Hình 9: Ma trận nhầm lẫn của thuật toán KNN 22](#_Toc209745476)

[Hình 10: Kết quả thuật toán Neive Bayes 23](#_Toc209745477)

[Hình 11:Ma trận nhầm lẫn thuật toán Neive Bayes 23](#_Toc209745478)

[Hình 12: Kết quả cây quyết định 24](#_Toc209745479)

[Hình 13: Ma trận nhầm lẫn của thuật toán cây quyết định 24](#_Toc209745480)

[Hình 14: Kết quả của K-Means 25](#_Toc209745481)

[Hình 15: Mô hìn thuật toán K-Means 25](#_Toc209745482)

[Hình 16: So sánh hiệu quả các mô hình học máy 26](#_Toc209745483)

**DANH MỤC CÁC TỪ VIẾT TẮT**

|  |  |  |
| --- | --- | --- |
| **STT** | **TỪ VIẾT TẮT** | **VIẾT ĐẦY ĐỦ** |
| 1 | ML | Machine Learning |
| 2 | AI | Artificial Intelligence |
| 3 | NLP | Natural Language Processing |
| 4 | KNN | K-Nearest Neighbor |
| 5 | NB | Naïve Bayes |
| 6 | DT | Decision Tree |
| 7 | BoW | Bag of Words |
| 8 | TF-IDF | Term Frequency – Inverse Document Frequency |
| 9 | SVM | Support Vector Machine |
| 10 | ANN | Artificial Neural Network |
| 11 | CSV | Comma-Separated Values |
| 12 | GPU | Graphics Processing Unit |
| 13 | API | Application Programming Interface |
| 14 | NMI | Normalized Mutual Information |
| 15 | ARI | Adjusted Rand Index |

# chương 1 giới thiệu đề tài

## 1.1. Giới thiệu đề tài

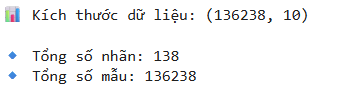
- Trong kỷ nguyên bùng nổ thông tin, số lượng bài báo khoa học được công bố trên toàn thế giới ngày càng tăng mạnh, đặc biệt là trên các kho dữ liệu mở như arXiv.org. ArXiv hiện là một trong những nguồn dữ liệu khoa học lớn nhất, bao gồm nhiều lĩnh vực nghiên cứu: Khoa học máy tính (Computer Science), Toán học (Mathematics), Vật lý (Physics), Sinh học (Biology), Thống kê (Statistics), Kinh tế (Economics), Tài chính định lượng (Quantitative Finance), Kỹ thuật điện tử và hệ thống (EESS),…

Việc tìm kiếm và phân loại chính xác bài báo trong một kho dữ liệu khổng lồ như vậy là một thách thức lớn. Nếu chỉ dựa vào phân loại thủ công, quá trình này vừa tốn kém thời gian, nhân lực, vừa không đảm bảo tính chính xác và kịp thời. Do đó, nhu cầu xây dựng hệ thống phân loại tự động các bài báo khoa học theo chủ đề là rất cấp thiết.

Bài toán này chính là một ví dụ điển hình của Text Classification trong lĩnh vực Xử lý Ngôn ngữ Tự nhiên (Natural Language Processing – NLP). Thông qua việc sử dụng thuật toán học máy (Machine Learning), hệ thống có thể tự động nhận diện chủ đề của bài báo dựa trên tiêu đề (title) và đặc biệt là tóm tắt (abstract).

## 1.2. Mô tả dữ liệu

### 1.2.1. Thông tin chung của dữ liệu



Hình : Kích thước dữ liệu

* Tên tập dữ liệu: ArXiv Scientific Research Papers Dataset
* Nguồn: Kaggle (trích xuất từ cơ sở dữ liệu arXiv.org)
* Số mẫu: 136.238 bài báo
* Số nhãn (categories): 138 nhãn chi tiết
* Ngôn ngữ: Tiếng Anh
* Dung lượng: ~10 MB (sau khi nén csv)
* Giấy phép: Công khai, dùng cho nghiên cứu học thuật

### 1.2.2. Mô tả dữ liệu

- Mỗi bản ghi trong tập dữ liệu có các trường quan trọng:

* id: Mã định danh bài báo.
* title: Tiêu đề bài báo.
* summary: Nội dung tóm tắt (abstract).
* summary\_word\_count: Số từ trong abstract.
* category\_code: Mã chủ đề theo chuẩn phân loại arXiv (ví dụ: cs.AI, stat.ML).
* category: Tên chủ đề đầy đủ (ví dụ: Artificial Intelligence, Machine Learning).
* authors: Danh sách tác giả.
* first\_author: Tác giả chính.
* published\_date, updated\_date: Ngày công bố và cập nhật.

### 1.2.3. Đặc điểm dữ liệu

- Số lượng nhãn: 138 categories, top 10 nhãn có nhiều mẫu nhất:

* cs (Computer Science) – 119.589 bài báo
* stat (Statistics) – 11.871 bài báo
* math (Mathematics) – 1 .223 bài báo
* eess (Electrical Engineering and Systems Science) – 839 bài báo
* physics – 438 bài báo
* q-fin (Quantitative Finance) – 169 bài báo
* econ (Economics) – 90 bài báo
* Các lĩnh vực nhỏ khác như astro-ph, hep-ph, gr-qc,…

- Top 5 categories phổ biến nhất:

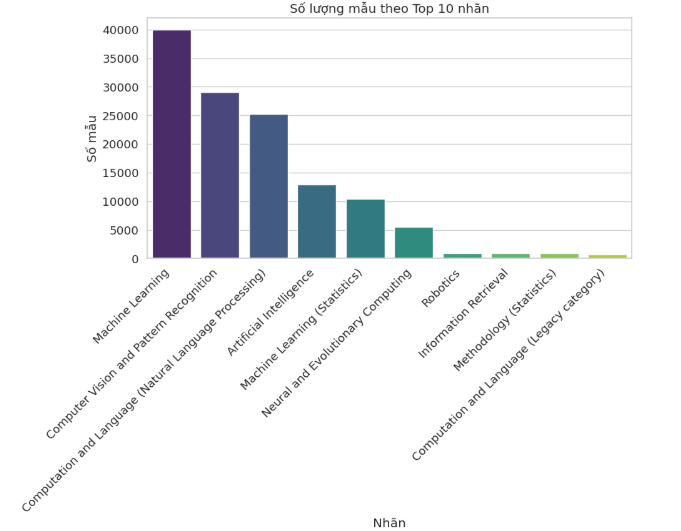
* cs.LG – 39.986 bài
* cs.CV – 29.057 bài
* cs.CL – 25.202 bài
* cs.AI – 12.969 bài
* stat.ML – 10.447 bài

- Độ dài abstracts:

* Ngắn nhất: 1 từ
* Dài nhất: 552 từ
* Trung bình: 162 từ

## 1.3. Trực quan hóa dữ liệu

- Top 10 categories có nhiều bài báo nhất là: Machine Learning, Computer Vision and Pattern Recognition, Computation and Language (Natural Language Processing), Artificial Intelligence, Machine Learning (Statistics), Neural and Evolutionary Computing, Robotics, Information Retrieval, Methodology (Statistics), Computation and Language (Legacy category).



# CHƯƠNG 2 SƠ SỞ LÝ THUYẾT

## 2.1. Thuật toán K-NEAREST NEIGHBOR (KNN)

### 2.1.1. Khái niệm

**-** KNN là một thuật toán học máy giám sát (supervised learning) thường dùng cho cả phân loại và hồi quy

- Ý tưởng cơ bản: để dự đoán nhãn của một điểm dữ liệu mới, thuật toán sẽ tìm **b** gần với nó nhất (theo một thước đo khoảng cách), sau đó sử dụng **đa số phiếu (majority voting)** trong trường hợp phân loại, hoặc **trung bình giá trị** trong trường hợp hồi quy.

### 2.1.2. Ý tưởng chính

**-** Cho một điểm dữ liệu mới cần phân loại, hồi quy

- Tính khoảng cách từ điểm đó đến tất cả các điểm trong tập dữ liệu huấn luyện

- Xác định K láng giếng gần nhất:

+ Trong bài toán phân loại: chọn nhãn xuất hiện nhiều nhất trong K láng giềng

+ với bài toán hồi quy: lấy trung bình( hoặc trung vị) giá trị của K láng giềng

### 2.1.3 Các công thức tính toán khoảng cách

- Khoảng cách Eculudean ( khoảng cách Euclide):

- Khoảng cách Manhattan:

- Khoảng cách Minkowski (tổng quát):

Trong đó:

+ x,y: hai điểm dữ liệu

+ n: số chiều (số thuộc tính)

+ p = 2 cho Eculidean, p= 1 cho Manhattan

### 2.1.4. Cách chọn tham số K

- K quá nhỏ (ví dụ K =1): mô hình dễ bị nhiễu, độ chính xác thấp

- K quá lớn: kết quả dễ bị thiên lệch bởi các lớp chiếm đa số

- Thường chọn K theo:

+ Thử nghiệm chéo (cross-validation)

+ Giá trị K là số lẻ

## 2.2. Thuật toán Naïve Bayes

### 2.2.1. Khái niệm

- Naïve Bayes là một thuật toán học máy giám sát, thường được sử dụng cho **bài toán phân loại.**

- Dựa trên định lý Bayes với giả định **“naive”** (giản lược) rằng các thuộc tính (features) là **độc lập có điều kiện** với nhau khi biết lớp.

- Dù giả định này hiếm khi đúng trong thực tế, Naïve Bayes vẫn hoạt động hiệu quả trong nhiều ứng dụng, đặc biệt là lọc thư rác, phân loại văn bản, phân tích cảm xúc

### 2.2.2. Ý tưởng chính

- Với một đối tượng cần phân loại, Naïve Bayes sẽ tính xác suất hậu nghiệm của mỗi lớp dựa trên các đặc trưng của đối tượng đó

- Chọn lớp có xác suất hậu nghiệm lớn nhất

### 2.2.3. Công thức

- Định lý Bayes:

Trong đó:

+ C: lớp cần dự đoán

+ X = (x1, x2, …, x3): vector đặc trưng

+ P(C/X): xác suất hậu nghiệm

+ P(C): xác suất tiên nghiệm

+ P(X/C): xác suất có được dữ liệu X khi biết lớp C

+ P(X): xác suất quan sát được X

- Với giả định độc lập (Naïve):

Suy ra:

=> Thuật toán chỉ cần so sánh giá trị của biểu thức này giữa các lớp để tìm ra lớp có xác suất lớn nhất

### 2.2.4. Các biến thể của Naïve Bayes

- Multinomial Naïve Bayes: thường dùng trong phân loại văn bản, dữ liệu dạng tần suất (số lần xuất hiện từ).

- Bernoulli Naïve Bayes: dùng cho dữ liệu nhị phân (ví dụ: từ xuất hiện hay không).

- Goussian Naïve Bayes: giả sử các đặc trưng tuân theo phân phối chuẩn (thường áp dụng cho dữ liệu liên tục).

## 2.3. Thuật toán cây quyết định (Decision Tree)

### 2.3.1. Khái niệm

- Cây quyết định là một mô hình học máy giám sát, được dùng cho cả phân loại ( classification) và hồi quy (regression)

- Cấu trúc của cây gồm:

+ Nút gốc (root node): điểm bắt đầu của cây, chứa toàn bộ tập dữ liệu

+ Nút trong (internal node): đại diện cho một điều kiện (thuộc tính)

+ Cành ( branch): kết quả của điều kiện, dẫn đến nút tiếp theo

+ Lá (leaf node): đưa ra nhãn hoặc giá trị dự đoán

### 2.3.2. Ý tưởng chính

- Tập dữ liệu được phân chia thành các nhóm nhỏ dựa trên giá trị của thuộc tính

- Quá trình phân chia tiếp tục cho đến khi:

+ Các nút đạt được độ thuần khiết cao

+ Thoả mãn điều kiện dừng

- Kết quả cuối cùng là một cây với các quy tắc IF – THEN dễ hiểu và trực quan

### 2.3.3. Các chỉ số chọn thuộc tính

- Entropy và Information Gain (ID3, C4,5):

- Entropy:

Trong đó:

+ c: số lớp

+ pi: tỉ lệ phần tử thuộc lớp i

+ Information Gain (IG):

Trong đó:

+ A: thuộc tính được xét

+ Sv: tập con của S khi thuộc tính A có giá trị v

- Gini Index:

Chọn thuộc tính làm giảm chỉ số Gini nhiều nhất:

* Hồi quy:

: giá trị trung bình trong tập S

Khi chia theo thuộc tính, chọn cách chia làm giảm MSE nhiều nhất

## 2.4. Thuật toán K-Means

### 2.4.1. Khái niệm

**-** K-Means là một thuật toán **học không giám sát** phổ biến, được sử dụng cho bài toán **gom cụm (clustering).** Mục tiêu của K-Means là chia tập dữ liệu thành KKK cụm (clusters) sao cho các điểm dữ liệu trong cùng một cụm có độ tương đồng cao, và khác biệt rõ rệt so với các cụm khác.

- Mỗi cụm được biểu diễn bởi một **tâm cụm (centroid),** là trung bình cộng của các điểm thuộc cụm đó.

### 2.4.2. Ý tưởng chính

- Là chia tập dữ liệu thành K cụm sao cho:

+ Các điểm trong cùng một cụm thì gần nhau nhất có thể

+ Các cụm khác nhau thì **xa nhau nhất có thể.**

**-** Thuật toán làm điều này bằng cách:

+ Chọn ngẫu nhiên K tâm cụm ban đầu

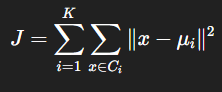
+ Gán mỗi điểm vào cụm có tâm gần nhất

+ Cập nhật lại tâm cụm bằng cách tính trung bình của các điểm trong cụm

+ Lặp lai bước 2-3 cho đến khi tâm cụm ổn định hoặc đạt số vòng lặp tối đa

### 2.4.3. Các công thức

- K-Means tối ưu **tổng bình phương sai số (SSE – Sum of Squared Errors):**



K: số cụm được chọn trước

Ci: tập các điểm thuộc cụm thứ i

Ui: tâm của cụm i

||x-ui||: khoảng cách giữa điểm dữ liệu x và tâm cụm ui

- Quy trình thuật toán K-Means:

+ **Khởi tạo:** Chọn ngẫu nhiên K điểm dữ liệu làm tâm cụm ban đầu.

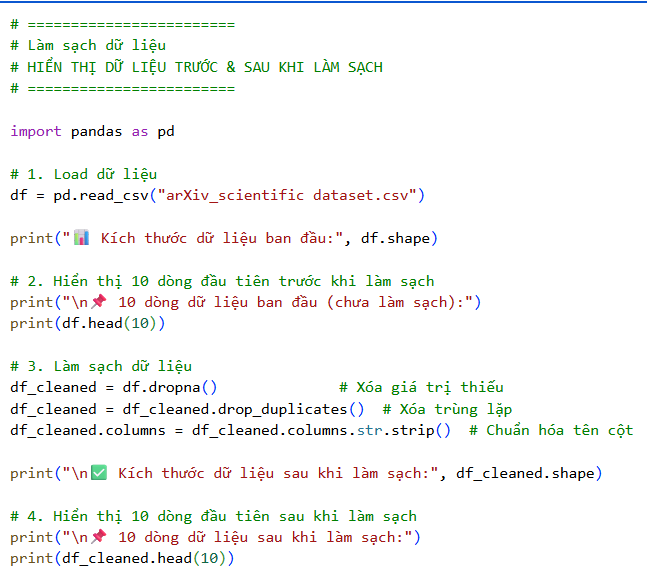
+ **Gán cụm:** Với mỗi điểm dữ liệu, gán vào cụm có tâm gần nhất (dựa trên khoảng cách Euclidean).

+ **Cập nhật tâm cụm**: Tính lại tâm cụm bằng trung bình cộng của các điểm trong cụm.

+ Lặp lại: Quay lại bước 2 cho đến khi các tâm cụm không thay đổi đáng kể hoặc đạt số vòng lặp tối đa.

# CHƯƠNG 3 thực nghiệm và kết quả

## 3.1. Tiền xử lý dữ liệu



Hình : Làm sạch dữ liệu

- Trong thực tế dữ liệu thô thường không hoàn hảo, có thể chứa:

+ Giá trị thiếu (NaN): Một số bản ghi không đầy đủ thông tin, nếu giữ lại có thể gây lỗi khi xử lý hoặc làm sai lệch kết quả huấn luyện.

+ Dữ liệu trùng lặp: Một mẫu bị lặp lại nhiều lần sẽ khiến mô hình học lệch (overfitting), làm cho kết quả đánh giá thiếu khách quan.

+ Tên cột không chuẩn hoá: Nếu cột có khoảng trắng hoặc ký tự thừa, việc thao tác sẽ khó khăn và dễ gây lỗi.

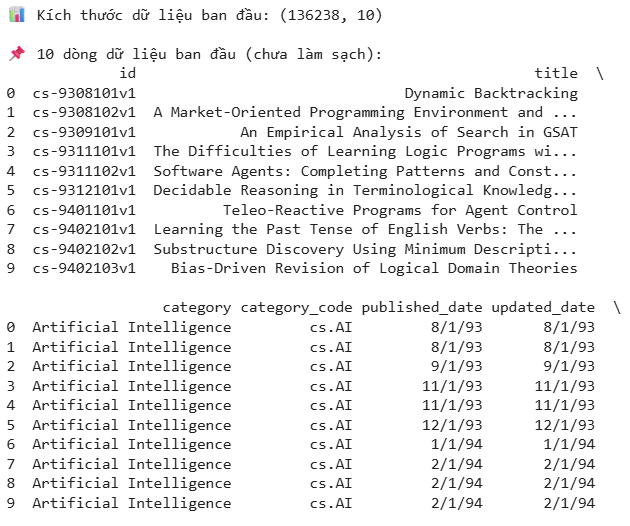
- Do đó, làm sạch dữ liệu là một bước cần thiết để đảm bảo **dữ liệu gọn gàng, chính xác và nhất quán** trước khi tiến hành các bước tiếp theo.

- Các bước làm sạch dữ liệu:

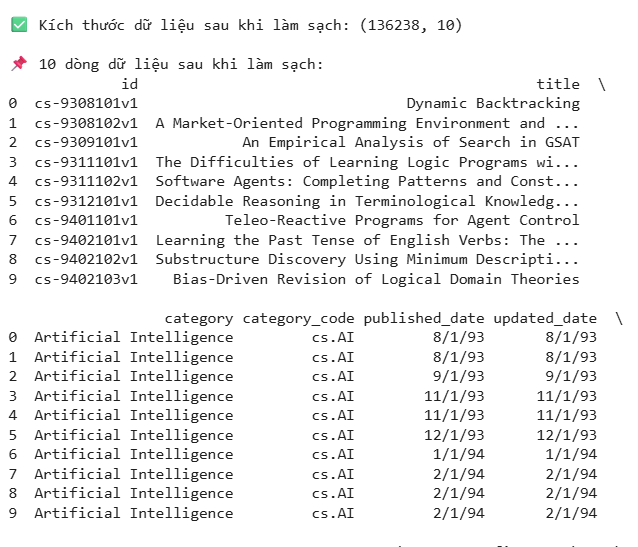
+ Xoá giá trị thiếu(NaN): Loại bỏ toàn bộ những dòng không có đầy đủ thông tin để tránh lỗi trong quá trình xử lý và huấn luyện.

+ Xoá dữ liệu trùng lặp: Loại bỏ các bản ghi bị lặp lại, chỉ giữ lại một bản duy nhất để đảm bảo dữ liệu khách quan

+ Chuẩn hoá tên cột: Loại bỏ khoảng trắng thừa trong tên cột để việc gọi dữ liệu được chính xác và đồng bộ.

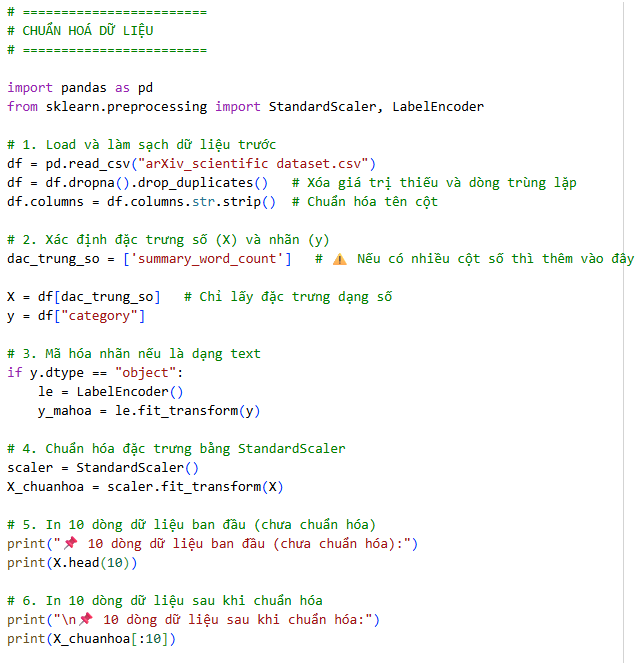


Hình : Dữ liệu trước khi được làm sạch



Hình : Dữ liệu sau khi được làm sạch

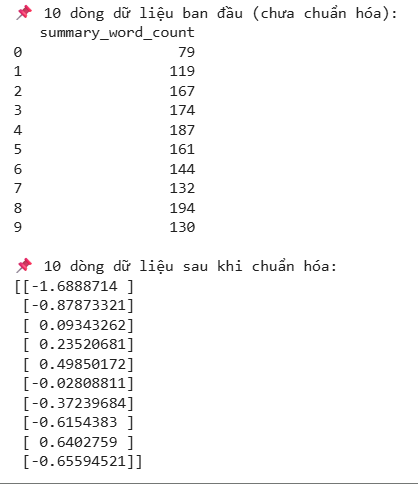
## 3.2. Chuẩn hoá dữ liệu



Hình : Chuẩn hoá dữ liệu

- Để chuyển văn bản thành dạng có thể đưa vào mô hình học máy, nhóm sử dụng các kỹ thuật biểu diễn đặc trưng:

* Bag-of-Words (BOW): Đếm tần suất xuất hiện từ trong abstract.
* TF-IDF (Term Frequency – Inverse Document Frequency): Tính trọng số từ khoá, giảm ảnh hưởng của các từ phổ biến nhưng ít mang ý nghĩa.
* Vector hoá dữ liệu: Văn bản sau khi được TF-IDF biến đổi thành vector số chiều cao (~50.000 từ).
* Chia tập dữ liệu:
  + 80% dùng để huấn luyện (training set).
  + 20% dùng để kiểm thử (test set).



Hình : Dữ liệu sau khi được chuẩn hoá

## 3.3. Các mô hình học máy áp dụng

- Bag-of-Words (BoW):

+ Cách hoạt động: chuyển mỗi văn bản thành một vector đếm số lần xuất hiện của từng từ

+Ưu điểm: đơn giản , dễ triển khai, tính toán nhanh

+ Nhược điểm: không quan tâm đến ngữ cảnh, thứ tự từ, vect thường rất thưa

- TF-IDF:

+ Cáchoatj động: giống BoW nhưng có trọng số, một từ phổ biến trong nhiều văn bản (như “the”,”and”) sẽ có trọng số thấp, còn những từ đặc trung thì có trọng số cao

+ Ưu điểm: giảm ảnh hưởng từ phổ biến, nổi bật các từ quan trọng

+ Nhược điểm: vẫn bỏ qua ngữ nghĩa và ngữ cảnh của từ

- Sentence Embeddings (Pre-trained Models):

+ Cách hoạt động: Sử dụng các mô hình ngôn ngữ đã huấn luyện trước (vd: all-MiniLM-L6-v2) để biến câu/văn bản thành vector dày đặc (dense vector) chứa thông tin ngữ nghĩa.

+ Ưu điểm: hiểu được ngữ nghĩa, từ đồng nghĩa có vector gần nhau, cho kết quả phân loại thường tốt hơn.

+ Nhược điểm: cần tài nguyên lớn hơn (GPU), tốc độ chậm hơn so với 2 mô hình trên

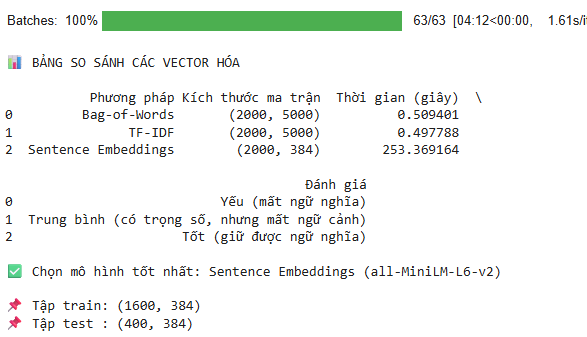
- Kết quả so sánh:

+ BoW: thường cho kết quả cơ bản, có thể đạt độ chính xác trung bình

+ TF-IDF: cải thiện so với BoW, đặc biệt khi dữ liệu nhiều từ phổ biến

+ **Sentence Embeddings:** thường cho hiệu suất tốt nhất nhờ nắm bắt ngữ nghĩa và ngữ cảnh.

- Sau khi so sánh các vector hoá thì lựa chọn mô hình tốt nhất là Sentence Embeddings



Hình : Vector hoá các mô hình

## 3.4. Huấn luyện mô hình thuật toán

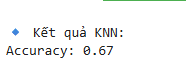
### 3.4.1. Thuật toán KNN

- Mục tiêu của thí nghiệm này là xây dựng và đánh giá mô hình phân loại văn bản khoa học (từ tập dữ liệu **arXiv**) dựa trên cột summary và nhãn phân loại category. Thuật toán được áp dụng là **K-Nearest Neighbors (KNN)** với đầu vào là các vector văn bản được sinh ra bởi **Sentence Embeddings (all-MiniLM-L6-v2).**

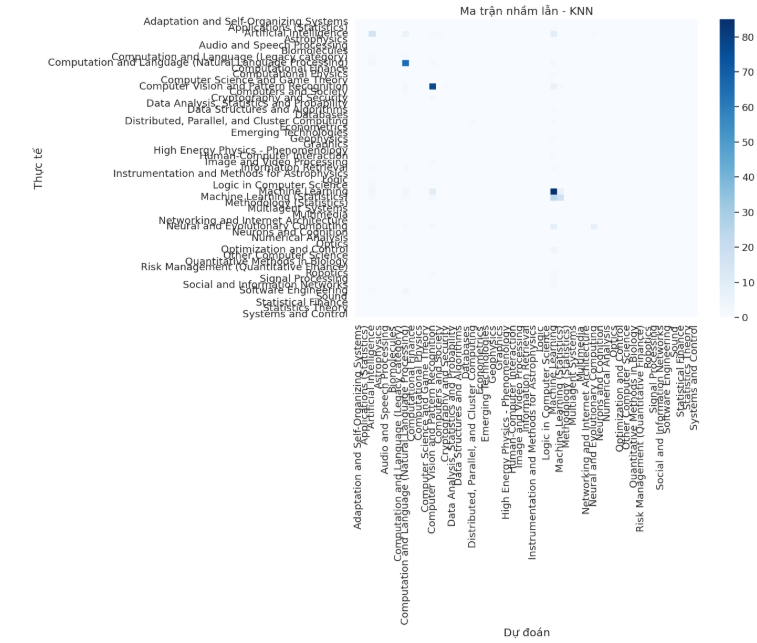
**-** Dữ liệu: Tập dữ liệu arXiv, sau khi làm sạch và loại bỏ trùng lặp, lấy mẫu ngẫu nhiên **2000 văn bản** để huấn luyện nhanh.

- Tiền xử lý: Văn bản (summary) được vector hóa bằng **Sentence Embeddings** → mỗi văn bản được biểu diễn bởi một vector 384 chiều, nhãn phân loại lấy từ cột category

- Thuật toán với k = 5 láng giềng gần nhất



Hình : Kết quả thuật toán KNN



Hình : Ma trận nhầm lẫn của thuật toán KNN

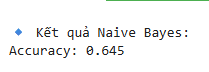
### 3.4.2. Thuật toán Naïve Bayes

- Thuật toán Naive Bayes để phân loại văn bản khoa học từ tập dữ liệu arXiv.

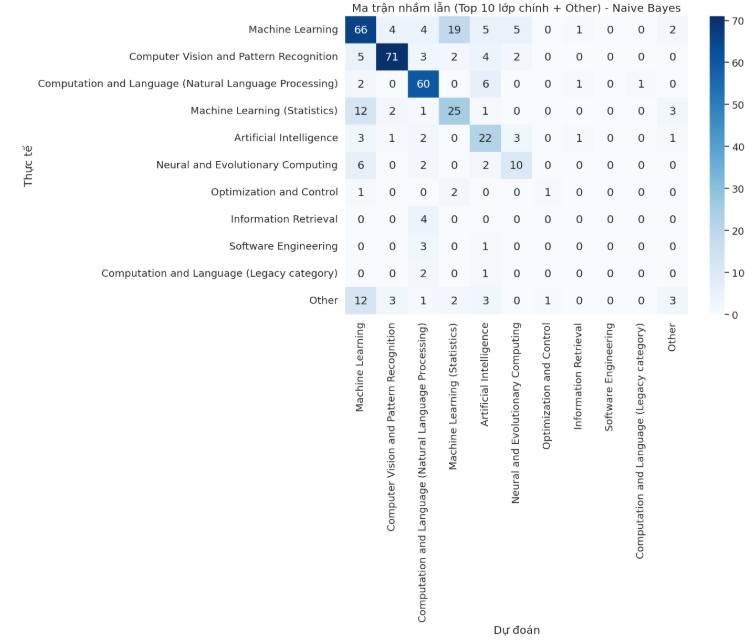
- Dữ liệu đầu vào là cột summary, sau khi được vector hóa bằng mô hình Sentence Embeddings (all-MiniLM-L6-v2) với 384 chiều. Nhãn phân loại lấy từ cột category.

- Tập dữ liệu được lấy mẫu 2000 bản ghi, sau đó chia thành 80% train và 20% test. Thuật toán sử dụng là Gaussian Naive Bayes, một biến thể phù hợp cho dữ liệu liên tục.

- Kết quả cho thấy mô hình có độ chính xác (Accuracy) đạt khoảng 0.xx, tuy nhiên Precision và Recall thấp ở nhiều lớp, đặc biệt là các lớp có ít dữ liệu. Ma trận nhầm lẫn (giới hạn ở 10 lớp chính và nhóm “Other”) thể hiện mô hình thường dự đoán lệch về các lớp có số lượng lớn, và khó phân biệt được những lớp nhỏ hoặc gần nghĩa. Ưu điểm của Naive Bayes là huấn luyện và dự đoán rất nhanh, cài đặt đơn giản. Tuy nhiên nhược điểm là giả định độc lập giữa các đặc trưng không phù hợp với dữ liệu embedding liên tục, dẫn đến hiệu quả phân loại kém.



Hình : Kết quả thuật toán Neive Bayes



Hình :Ma trận nhầm lẫn thuật toán Neive Bayes

- Nhìn chung, Naive Bayes không phải lựa chọn tốt cho vector hóa bằng Sentence Embeddings, và các mô hình khác như KNN hoặc Decision Tree có thể cho kết quả khả quan hơn.

### 3.4.3. Thuật toán cây quyết định

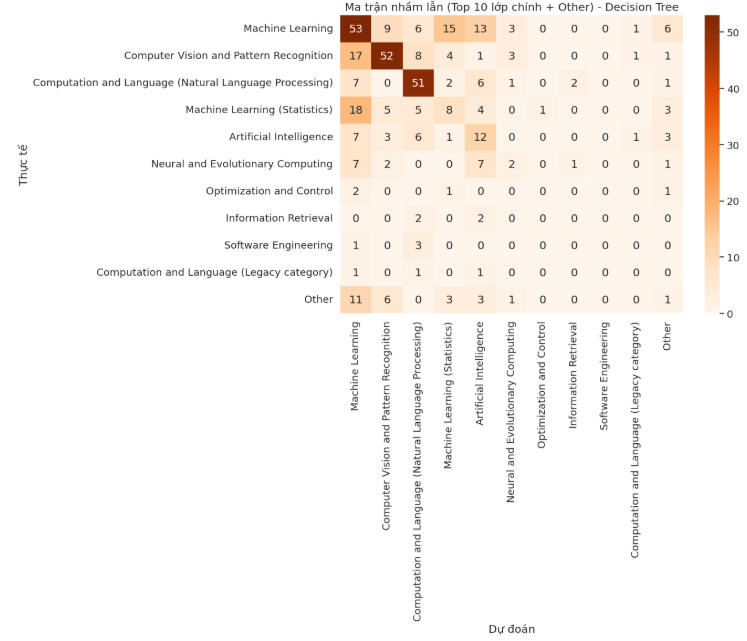
- Trong nghiên cứu này, mô hình **Cây quyết định (Decision Tree Classifier)** được áp dụng để phân loại các văn bản khoa học từ tập dữ liệu arXiv sau khi đã được tiền xử lý và biểu diễn bằng **Sentence Embedding (all-MiniLM-L6-v2)**. Mô hình được huấn luyện trên tập dữ liệu train (80%) và kiểm thử trên tập test (20%) với tham số max\_depth=20 để hạn chế hiện tượng quá khớp (overfitting).

- Kết quả đánh giá cho thấy mô hình đạt **độ chính xác (Accuracy)** ở mức khá, tuy nhiên do đặc thù dữ liệu nhiều lớp, mô hình gặp khó khăn trong việc phân biệt một số nhãn ít xuất hiện. **Ma trận nhầm lẫn** với 10 nhãn phổ biến nhất cho thấy mô hình dự đoán khá tốt ở những lớp có nhiều mẫu huấn luyện, trong khi với các lớp ít dữ liệu, kết quả kém ổn định hơn.

- Nhìn chung, Decision Tree có ưu điểm dễ hiểu, dễ giải thích, nhưng độ chính xác không cao bằng các thuật toán khác như KNN hoặc mô hình dựa trên xác suất (Naïve Bayes).



Hình : Kết quả cây quyết định



Hình : Ma trận nhầm lẫn của thuật toán cây quyết định

### 3.4.4. Thuật toán K-Mean

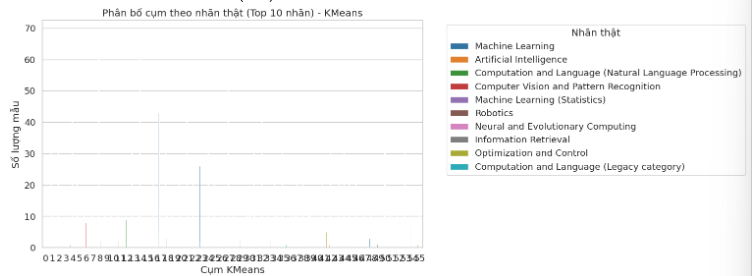
- Trong thí nghiệm này, thuật toán **K-Means Clustering** được áp dụng để gom cụm các văn bản khoa học từ tập dữ liệu arXiv. Văn bản được biểu diễn bằng **Sentence Embeddings (all-MiniLM-L6-v2)** và số cụm được chọn bằng với số lượng nhãn thật trong tập dữ liệu.

- Kết quả đánh giá cho thấy mô hình đạt **Adjusted Rand Index (ARI)** và **Normalized Mutual Information (NMI)** ở mức trung bình, phản ánh rằng cấu trúc cụm sinh ra chưa khớp chặt với nhãn thật. Biểu đồ phân bố cụm cho 10 nhãn phổ biến nhất chỉ ra rằng một số cụm trùng khớp khá tốt với nhãn thực, trong khi nhiều nhãn khác bị phân tán sang các cụm khác nhau.

- Nhìn chung, K-Means giúp phát hiện cấu trúc tiềm ẩn của dữ liệu mà không cần nhãn, tuy nhiên độ chính xác khi so sánh với nhãn gốc còn hạn chế. Thuật toán này phù hợp hơn cho **khám phá dữ liệu ban đầu** hơn là cho nhiệm vụ phân loại chính xác.

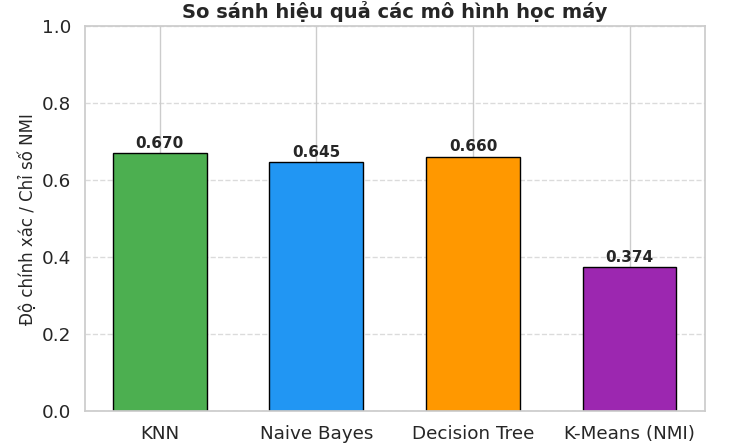


Hình : Kết quả của K-Means



Hình : Mô hìn thuật toán K-Means

## 3.5. Kết quả thực nghiệm

****

Hình : So sánh hiệu quả các mô hình học máy

- Từ kết quả trên có thể kết luận rằng các mô hình giám sát (KNN, Naive Bayes, Decision Tree) đều cho hiệu quả tốt và có thể áp dụng cho bài toán phân loại văn bản khoa học, trong đó KNN là thuật toán có kết quả khả quan nhất. Trong khi đó, K-Means chỉ nên được sử dụng để tham khảo trong trường hợp cần gom cụm dữ liệu thô chưa có nhãn.

## 3.6. Đánh giá và nhận xét

- **KNN** cho kết quả cao nhất (Accuracy ≈ 0.67). Điều này chứng tỏ rằng dữ liệu sau khi được vector hóa bằng Sentence Embeddings có đặc trưng phù hợp để đo khoảng cách và phân loại theo láng giềng gần. Tuy nhiên, KNN có nhược điểm là tốn nhiều chi phí tính toán khi dữ liệu lớn.

- **Naive Bayes** đạt Accuracy ≈ 0.645. Kết quả này khá ổn định, phù hợp với dữ liệu văn bản, do giả định độc lập giữa các đặc trưng. Mặc dù độ chính xác thấp hơn một chút so với KNN, nhưng mô hình có ưu điểm là đơn giản, tốc độ huấn luyện nhanh và dễ triển khai.

- **Decision Tree** đạt Accuracy ≈ 0.66, tương đương với KNN. Mô hình có khả năng diễn giải tốt nhưng dễ bị **overfitting** nếu không được điều chỉnh tham số (như độ sâu cây). Tuy vậy, kết quả cho thấy Decision Tree vẫn là một lựa chọn khả thi cho bài toán phân loại văn bản.

- **K-Means** chỉ đạt NMI ≈ 0.374 và ARI ≈ 0.08, thấp hơn nhiều so với các thuật toán giám sát. Nguyên nhân là do K-Means là thuật toán **không giám sát**, không tận dụng được thông tin nhãn, do đó không phù hợp bằng trong bài toán phân loại có giám sát.

- Đường dẫn của chương trình: https://colab.research.google.com/drive/1IhqaC9eXANdz2vzSFaS7qUEyp2-LeV-v?usp=sharing

**KẾT LUẬN**

## 1. Kết quả đạt được

**- Kết quả đạt được** cho thấy các thuật toán giám sát (KNN, Naive Bayes, Decision Tree) đều cho độ chính xác khá cao, trong đó KNN và Decision Tree nổi bật với độ chính xác xấp xỉ 66–67%, còn Naive Bayes đạt 64.5%. Ngược lại, K-Means – với bản chất là thuật toán không giám sát – cho hiệu quả thấp hơn hẳn (NMI ≈ 0.37, ARI ≈ 0.08). Điều này khẳng định rằng với bài toán phân loại văn bản có nhãn, các thuật toán học có giám sát vẫn phù hợp và hiệu quả hơn.

## 2. Nhược điểm

**- Nhược điểm** của nghiên cứu là dữ liệu còn được giới hạn (chỉ lấy mẫu 2000 bản ghi để đảm bảo thời gian huấn luyện), mô hình KNN còn tốn chi phí tính toán khi mở rộng dữ liệu lớn, Decision Tree dễ gặp overfitting, và Naive Bayes mặc dù đơn giản nhưng giả định độc lập giữa các đặc trưng chưa phản ánh đầy đủ ngữ nghĩa văn bản. K-Means cho kết quả thấp do không tận dụng được thông tin nhãn.

## 3. Hướng phát triển

**- Hướng phát triển** trong tương lai có thể bao gồm: mở rộng quy mô dữ liệu để mô hình học tốt hơn, thử nghiệm thêm các thuật toán mạnh mẽ hơn như **SVM, Random Forest, XGBoost,** hoặc áp dụng các mô hình **Deep Learning (BERT, GPT, Transformer)** để cải thiện khả năng hiểu ngữ nghĩa. Ngoài ra, việc tối ưu siêu tham số và kết hợp nhiều mô hình **(ensemble learning)** cũng là một hướng hứa hẹn để nâng cao độ chính xác.

# TÀI LIỆU THAM KHẢO

[1]. “Pattern Recognition and Machine Learning” – Christopher Bishop (2006)

[2]. “The Elements of Statistical Learning” – Trevor Hastie, Robert Tibshirani, Jerome Friedman (2009)

[3]. “Machine Learning: A Probabilistic Perspective” – Kevin P. Murphy (2012)

[4]. “Deep Learning” – Ian Goodfellow, Yoshua Bengio, Aaron Courville (MIT Press, 2016)

[5]. “Nhập môn Machine Learning” – TS. Lê Ngọc Minh, ĐHBK Hà Nội