

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI PHÂN HIỆU TẠI TP. HỒ CHÍ MINH**

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**

**BÁO CÁO BÀI TẬP LỚN**

**HỌC PHẦN: HỌC MÁY CƠ BẢN**

ĐỀ TÀI: XÂY DỰNG ỨNG DỤNG DI ĐỘNG STYLO CHO CỬA HÀNG THỜI TRANG TÍCH HỢP TÌM KIẾM VÀ GỢI Ý SẢN PHẨM THÔNG MINH

|  |
| --- |
| Giảng viên hướng dẫn: TRẦN THỊ DUNG |
| Sinh viên thực hiện: ĐẶNG THỊ KIM THẢO - 6351071066 |
| LÊ HOÀNG PHÚC - 6351071057 |
| VÕ THÀNH HOÀNG PHÚC - 6351071056 |
| Lớp: CÔNG NGHỆ THÔNG TIN  Khoá: 63 |

Tp. Hồ Chí Minh, năm 2025

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI PHÂN HIỆU TẠI TP. HỒ CHÍ MINH**



**BỘ MÔN CÔNG NGHỆ THÔNG TIN**

**BÁO CÁO BÀI TẬP LỚN**

HỌC PHẦN: HỌC MÁY CƠ BẢN

ĐỀ TÀI: XÂY DỰNG ỨNG DỤNG DI ĐỘNG STYLO CHO CỬA HÀNG THỜI TRANG TÍCH HỢP TÌM KIẾM VÀ GỢI Ý SẢN PHẨM THÔNG MINH

|  |
| --- |
| Giảng viên hướng dẫn: TRẦN THỊ DUNG |
| Sinh viên thực hiện: ĐẶNG THỊ KIM THẢO - 6351071066 |
| LÊ HOÀNG PHÚC - 6351071057 |
| VÕ THÀNH HOÀNG PHÚC - 6351071056 |
| Lớp: CÔNG NGHỆ THÔNG TIN |
| Khoá: 63 |

Tp. Hồ Chí Minh, năm 2025

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI **CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**

**PHÂN HIỆU TẠI THÀNH PHỐ HỒ CHÍ MINH** Độc lập – Tự do – Hạnh phúc



# NHIỆM VỤ THIẾT KẾ BÁO CÁO ĐỒ ÁN

BỘ MÔN: **CÔNG NGHỆ THÔNG TIN**

\*\*\*

**Mã sinh viên: Họ tên SV:**

6351071066 Đặng Thị Kim Thảo

6351071057 Lê Hoàng Phúc

6351071056 Võ Thành Hoàng Phúc

**Khóa:** 63 **Lớp:** Công Nghệ Thông Tin

## Tên đề tài báo cáo đồ án

Hệ thống AI hỗ trợ gợi ý sản phẩm và dự đoán khả năng mua hàng cho cửa hàng thời trang trực tuyến

## Mục đích, yêu cầu

* 1. **Mục đích:**
* Hệ thống gợi ý sản phẩm theo hành vi người dùng: Sử dụng hai thuật toán Random Forest và Gradient Boosting theo hướng hồi quy (Regression):
* Phân tích hành vi tương tác (view, add-to-cart, transaction).
* Dự đoán mức độ yêu thích nhằm ước lượng mức độ quan tâm của khách hàng đối với sản phẩm.
* Từ đó xếp hạng và gợi ý sản phẩm cá nhân hóa cho từng khách hàng.
* Tìm kiếm và gợi ý sản phẩm bằng hình ảnh:
* CNN / EfficientNet / ResNet để trích xuất embedding từ từ ảnh sản phẩm.
* FAISS index để xây dựng hệ thống tìm kiếm ảnh hiệu năng cao.
* Cosine Similarity để đo mức độ giống nhau giữa các sản phẩm.

## Yêu cầu:

* Cài đặt và triển khai mô hình Random Forest, Gradient Boosting và Image Embedding, FAISS trên Google Colab.
* Thực hiện chuẩn hóa dữ liệu, xây dựng pipeline huấn luyện và đánh giá mô hình.
* Xây dựng hệ thống tìm kiếm sản phẩm dựa trên hình ảnh.
* Đánh giá, trực quan hóa và so sánh kết quả dự đoán giữa hai thuật toán Random Forest và Gradient Boosting.

## Nội dung và phạm vi đề tài

* 1. **Nội dung:**
* Tìm hiểu lý thuyết nền tảng về hai thuật toán Random Forest và Gradient Boosting.
* Phân tích sự khác biệt giữa Random Forest và Gradient Boosting.
* Xây dựng bộ dữ liệu mô phỏng hành vi khách hàng trong mua sắm trực tuyến.
* Huấn luyện các mô hình gợi ý sản phẩm và dự đoán mua hàng sử dụng Random Forest và Gradient Boosting.
* Đánh giá, so sánh hiệu năng và phân tích kết quả.
* Thảo luận các khía cạnh kỹ thuật và tiềm năng ứng dụng trong thương mại điện tử.
  1. **Phạm vi đề tài:**
* Dữ liệu sử dụng có quy mô nhỏ đến trung bình, phù hợp với giới hạn tài nguyên của Google Colab và tính chất thử nghiệm của đồ án.
* Hệ thống chủ yếu tập trung vào xây dựng mô hình AI, chưa triển khai thành sản phẩm thực tế hay tích hợp Web/App.
* Sử dụng FAISS bản CPU.
* Sử dụng mô hình embedding có sẵn (transfer learning).

## Công nghệ, công cụ và ngôn ngữ lập trình

* 1. **Công nghệ:** FastAPI, TensorFlow, FAISS, ASP.NET
  2. **Công cụ:** Google Colab, VS Code, Visual Studio, Postman, GitHub
  3. **Ngôn ngữ lập trình:** Python, C#

## Các kết quả chính dự kiến sẽ đạt được và ứng dụng

* 1. **Kết quả dự kiến:**
* Xây dựng hệ thống gợi ý sản phẩm dựa trên hành vi với độ chính xác cao.
* Dự đoán được mức độ yêu thích (like\_score) bằng mô hình hồi quy.
* Hoàn thiện hệ thống tìm kiếm ảnh sản phẩm bằng embedding và FAISS.
* So sánh được hiệu quả giữa Random Forest và Gradient Boosting.
* Hoàn thiện hệ thống pipeline xử lý dữ liệu, huấn luyện và đánh giá mô hình AI trong thương mại điện tử.
  1. **Ứng dụng:**
* Phân tích hành vi khách hàng để đánh giá khả năng mua hàng trong tương lai.
* Tối ưu chiến lược marketing, cá nhân hóa khuyến mãi và chăm sóc khách hàng.
* Gợi ý sản phẩm phù hợp, tăng tỷ lệ chuyển đổi và doanh thu.
* Tìm kiếm sản phẩm thông qua hình ảnh.
* Làm nền tảng cho các hệ thống thông minh trong thương mại điện tử như:
* Gợi ý sản phẩm thời trang.
* Dự đoán nhu cầu mua sắm.
* Xếp hạng khách hàng tiềm năng.
* Cá nhân hóa trải nghiệm người dùng.

## Giáo viên và cán bộ hướng dẫn

Họ tên: Ths. Trần Thị Dung

Đơn vị công tác: Bộ môn Công Nghệ Thông Tin – Trường Đại học Giao thông Vận tải phân hiệu tại TP. Hồ Chí Minh.

Điện thoại: Email:

|  |  |
| --- | --- |
| **Ngày 22 tháng 11 năm 2025**  **Trưởng BM Công nghệ Thông tin** | **Đã giao nhiệm vụ TKBCDA**  **Giáo viên hướng dẫn** |

|  |  |
| --- | --- |
| **ThS. Trần Phong Nhã** | **ThS. Trần Thị Dung** |

Đã nhận nhiệm vụ TKBCDA

Sinh viên: Ký tên:

VÕ THÀNH HOÀNG PHÚC LÊ HOÀNG PHÚC

ĐẶNG THỊ KIM THẢO

Điện thoại: Email:

[6351071056@st.utc2.edu.vn](mailto:6351071056@st.utc2.edu.vn) [6351071057@st.utc2.edu.vn](mailto:6351071057@st.utc2.edu.vn) [6351071066@st.utc2.edu.vn](mailto:6351071066@st.utc2.edu.vn)

# LỜI CẢM ƠN

Qua thời gian học tập và rèn luyện tại Trường Đại học Giao thông Vận tải phân hiệu tại TP. Hồ Chí Minh, đến nay chúng em đã được trang bị những kỹ năng, kiến thức cơ bản để có thể hoàn thành được bài tập cuối kì do giảng viên giao.

Chúng em cảm ơn tập thể các thầy cô giáo Bộ môn Công Nghệ Thông Tin và các thầy cô thỉnh giảng đã giảng dạy, quan tâm và không ngần ngại dành thời gian để chỉ bài và giải đáp những thắc mắc của chúng em trong những tiết học và cả những lúc ngoài giờ.

Và chúng em cảm ơn cô Thạc sĩ Trần Thị Dung đã luôn quan tâm nhiệt tình hướng dẫn, giúp đỡ chúng em trong quá trình triển khai và thực hiện bài tập cuối kì. Cô cũng luôn nhắc nhỡ, giúp đỡ mỗi khi chúng em gặp khó khăn, nhờ vậy mà chúng em đã hoàn thành bài tập cuối kì của nhóm mình đúng thời hạn được giao. Nếu không có sự hướng dẫn của cô thì có lẽ chúng em đã khó có thể thực hiện được bài tập đúng theo mong muốn của mình.

Chúng em đã bỏ ra nhiều thời gian để tìm hiểu và trang bị thêm kiến thức nhằm phục vụ cho việc thực hiện ý tưởng, nhưng chắc chắn rằng chúng em sẽ không thể tránh khỏi những sai sót không đáng có vì kiến thức còn hạn chế. Chúng em hi vọng rằng sẽ nhận được những lời góp ý quý báu của cô để có thể hoàn thiện ý tưởng của nhóm một cách tốt nhất có thể.

***TP. Hồ Chí Minh, ngày 22 tháng 11 năm 2025***

**Sinh viên thực hiện**

Đặng Thị Kim Thảo

Lê Hoàng Phúc

Võ Thành Hoàng Phúc

# NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

...............................................................................................................................................

...............................................................................................................................................

...............................................................................................................................................

...............................................................................................................................................

...............................................................................................................................................

...............................................................................................................................................

...............................................................................................................................................

...............................................................................................................................................

...............................................................................................................................................

...............................................................................................................................................

...............................................................................................................................................

...............................................................................................................................................

...............................................................................................................................................

...............................................................................................................................................

***Tp. Hồ Chí Minh, ngày 22 tháng 11 năm 2025***

**Giáo viên hướng dẫn**

**ThS. Trần Thị Dung**

# MỤC LỤC

# DANH MỤC BẢNG BIỂU

[Bảng 3.1 Các đặc trưng của tập dữ liệu 1 13](#_Toc214711419)

[Bảng 3.2 Các đặc trưng của tập dữ liệu 2 14](#_Toc214711420)

[Bảng 4.1. Độ chính xác của các mô hình sau khi huấn luyện 29](#_Toc214711421)

# 

# DANH MỤC HÌNH ẢNH

[Hình 2.1. Quá trình thực hiện bài toán AI 3](#_Toc214711483)

[Hình 2.2. Quy trình hoạt động của máy học 4](#_Toc214711484)

[Hình 2.3. Sự khác biệt giữa AI, Machine Learning, Deep Learning 4](#_Toc214711485)

[Hình 2.4. Ngôn ngữ lập trình Python 5](#_Toc214711486)

[Hình 2.5. Ngôn ngữ lập trình Python 6](#_Toc214711487)

[Hình 2.6. Google Colab 6](#_Toc214711488)

[Hình 2.7. Ví dụ thuật toán Random Forest 8](#_Toc214711489)

[Hình 2.8. Ví dụ thuật toán Gradient Boosting 9](#_Toc214711490)

[Hình 2.9. Ví dụ về mạng tích chập CNN 10](#_Toc214711491)

[Hình 2.10. Ví dụ về Vector Embedding 11](#_Toc214711492)

[Hình 2.11. Quy trình hoạt động của hệ thống tìm kiếm dựa trên Embedding 12](#_Toc214711493)

[Hình 3.1 Mô tả thuộc tính timestamp 14](#_Toc214711494)

[Hình 3.2 Mô tả thuộc tính visitorid 15](#_Toc214711495)

[Hình 3.3 Mô tả thuộc tính event 16](#_Toc214711496)

[Hình 3.4 Mô tả thuộc tính itemid 16](#_Toc214711497)

[Hình 3.5 Mô tả thuộc tính transactionid 17](#_Toc214711498)

[Hình 3.6 Mô tả thuộc tính id 18](#_Toc214711499)

[Hình 3.7 Mô tả thuộc tính year 19](#_Toc214711500)

[Hình 3.8 Mô tả thuộc tính gender 20](#_Toc214711501)

[Hình 3.9 Mô tả thuộc tính masterCategory 20](#_Toc214711502)

[Hình 3.10 Mô tả thuộc tính subCategory 21](#_Toc214711503)

[Hình 3.11 Mô tả thuộc tính articleType 21](#_Toc214711504)

[Hình 3.12 Mô tả thuộc tính baseColour 22](#_Toc214711505)

[Hình 3.13 Mô tả thuộc tính season 22](#_Toc214711506)

[Hình 3.14 Mô tả thuộc tính usage 23](#_Toc214711507)

[Hình 3.15 Mô tả thuộc tính productDisplayName 23](#_Toc214711508)

[Hình 4.1 Thông tin tổng quan ban đầu 24](#_Toc214711509)

[Hình 4.2 Dữ liệu còn lại sau khi sử lí trùng lặp 24](#_Toc214711510)

[Hình 4.3 Dữ liệu còn lại sau khi lọc 25](#_Toc214711511)

[Hình 4.4 Dữ liệu còn lại khi lọc bỏ chuỗi sự kiện không hợp lệ 25](#_Toc214711512)

[Hình 4.5 Dữ liệu sau khi xử lí Outliers 26](#_Toc214711513)

[Hình 4.6 Lịch sử tương tác giữa user và item 26](#_Toc214711514)

[Hình 4.7 Mô tả hành vi người dùng 27](#_Toc214711515)

[Hình 4.8 Phân tích hoạt động của user 27](#_Toc214711516)

[Hình 4.9 Phân bố biến mục tiêu 28](#_Toc214711517)

[Hình 4.10 Các user hoạt động nhiều nhất 28](#_Toc214711518)

[Hình 4.11 Các feature quan trọng 29](#_Toc214711519)

[Hình 4.12 Kết quả Random Forest 30](#_Toc214711520)

[Hình 4.13 Kết quả Gradient Boosting Regressor 31](#_Toc214711521)

[Hình 4.14 So sánh 2 mô hình 31](#_Toc214711522)

[Hình 4.15 Biểu đồ 2 mô hình 32](#_Toc214711523)

# BẢNG PHÂN CHIA CÔNG VIỆC

|  |  |  |
| --- | --- | --- |
| **Tên thành viên** | **Phân công công việc** | **Đánh giá** |
| Võ Thành Hoàng Phúc | Tiền xử lý dữ liệu  Tích hợp API cho mô hình. | 7.5 |
| Đặng Thị Kim Thảo | Nghiên cứu và chuẩn bị dữ liệu.  Triển khai mô hình CNN cho chức năng tìm kiếm và gợi ý sản phẩm bằng hình ảnh. | 7.5 |
| Lê Hoàng Phúc | Triển khai mô hình Random Forest và Gradient Boosting cho chức năng gợi ý sản phẩm và dự đoán mua hàng dựa trên hành vi. | 7.5 |

# CHƯƠNG 1. MỞ ĐẦU

### Giới thiệu chung về đề tài

Trong bối cảnh thương mại điện tử phát triển mạnh mẽ, các cửa hàng thời trang trực tuyến không chỉ cần đa dạng hóa sản phẩm mà còn phải chú trọng nâng cao trải nghiệm cá nhân hóa cho khách hàng. Việc hiểu rõ nhu cầu, thị hiếu và hành vi mua sắm của từng người dùng đã trở thành yếu tố quan trọng giúp doanh nghiệp tăng tỷ lệ chuyển đổi và duy trì khách hàng lâu dài.

Trí tuệ nhân tạo (AI) và học máy (Machine Learning) hiện đóng vai trò then chốt trong việc phân tích dữ liệu hành vi và tự động đưa ra gợi ý phù hợp. Hai hướng tiếp cận phổ biến trong các hệ thống thương mại điện tử bao gồm:

* Gợi ý sản phẩm và dự đoán khả năng mua hàng dựa trên hành vi tương tác, sử dụng các mô hình học máy để dự đoán mức độ yêu thích của người dùng đối với từng sản phẩm.
* Tìm kiếm và gợi ý sản phẩm dựa trên hình ảnh, giúp người dùng tìm sản phẩm tương tự thông qua ảnh chụp hoặc hình mẫu.

Xuất phát từ nhu cầu thực tiễn đó, đề tài "Hệ thống AI hỗ trợ gợi ý sản phẩm và dự đoán khả năng mua hàng cho cửa hàng thời trang trực tuyến" được thực hiện với mục tiêu xây dựng mô hình gợi ý sản phẩm từ dữ liệu hành vi người dùng và mô hình tìm kiếm sản phẩm dựa trên hình ảnh, từ đó minh họa vai trò và hiệu quả của AI trong thương mại điện tử.

### Mục tiêu nghiên cứu

* Xây dựng hệ thống gợi ý sản phẩm và dự đoán khả năng mua hàng theo hành vi khách hàng:
* Áp dụng Random Forest Regressor và Gradient Boosting Regressor theo hướng hồi quy để dự đoán like\_score (điểm yêu thích).
* Mô hình hóa hành vi người dùng qua chuỗi sự kiện: xem sản phẩm, thêm vào giỏ hàng, thực hiện giao dịch.
* Xếp hạng và đề xuất các sản phẩm phù hợp nhất theo từng người dùng.
* Xây dựng hệ thống tìm kiếm sản phẩm dựa trên hình ảnh:
* Sử dụng CNN (EfficientNet/ResNet) để trích xuất vector embedding từ ảnh sản phẩm.
* Áp dụng FAISS và cosine similarity để xây dựng hệ thống tìm kiếm ảnh hiệu năng cao.
* Cho phép người dùng tải ảnh và nhận gợi ý sản phẩm tương tự.
* Thực hiện quá trình tiền xử lý dữ liệu, huấn luyện, kiểm thử và so sánh kết quả theo quy trình chuẩn.
* Đề xuất hướng ứng dụng thực tế của hệ thống trong môi trường thương mại điện tử.

### Phạm vi nghiên cứu

* Hệ thống tập trung vào hai nhóm mô hình chính:
* Hồi quy gợi ý sản phẩm: Random Forest Regressor, Gradient Boosting Regressor.
* Tìm kiếm bằng hình ảnh: CNN feature extractor + FAISS.
* Phần gợi ý sản phẩm chỉ giới thiệu ở mức mô phỏng với hai thuật toán: Random Forest và Gradient Boosting, chưa tích hợp thành hệ thống gợi ý thực tế.
* Dữ liệu sử dụng có quy mô nhỏ đến trung bình, chủ yếu mang tính mô phỏng và thử nghiệm.
* Đề tài chỉ triển khai trong môi trường Google Colab, không xây dựng giao diện hoặc hệ thống hoàn chỉnh trên nền tảng web/app.
* Các vấn đề liên quan đến triển khai thực tế như tối ưu hiệu năng hệ thống, cơ sở dữ liệu người dùng lớn hoặc bảo mật thông tin không nằm trong phạm vi nghiên cứu.

### Phương pháp nghiên cứu

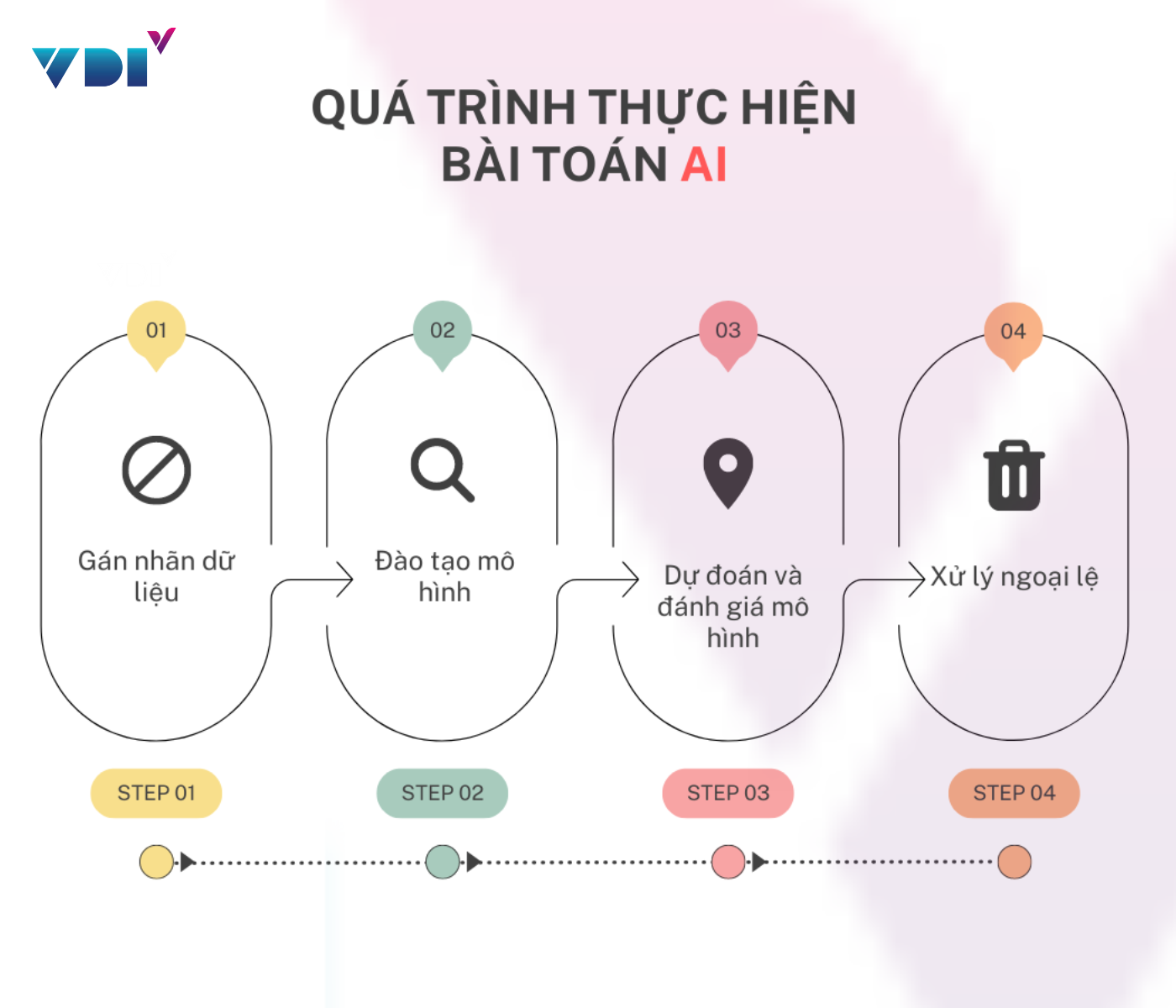
* Nghiên cứu tài liệu: Tìm hiểu lý thuyết về Random Forest, Gradient Boosting, CNN, Embedding, FAISS thông qua giáo trình và tài liệu chính thức.
* Phân tích – tổng hợp: So sánh đặc điểm, ưu nhược điểm và cách ứng dụng của các mô hình trong bài toán thực tế.
* Thực nghiệm mô hình: Xây dựng dữ liệu mô phỏng, thực hiện tiền xử lý, huấn luyện và đánh giá mô hình bằng Python/TensorFlow trong môi trường Google Colab.
* Đánh giá mô hình: Dựa trên R², MAE, RMSE, cosine similarity và tốc độ truy vấn.
* Xây dựng API và web demo:
* Backend API bằng FastAPI
* Web demo bằng ASP.NET C#

# CHƯƠNG 2. CƠ SỞ LÝ THUYẾT.

### Tổng quan về trí tuệ nhân tạo (AI)

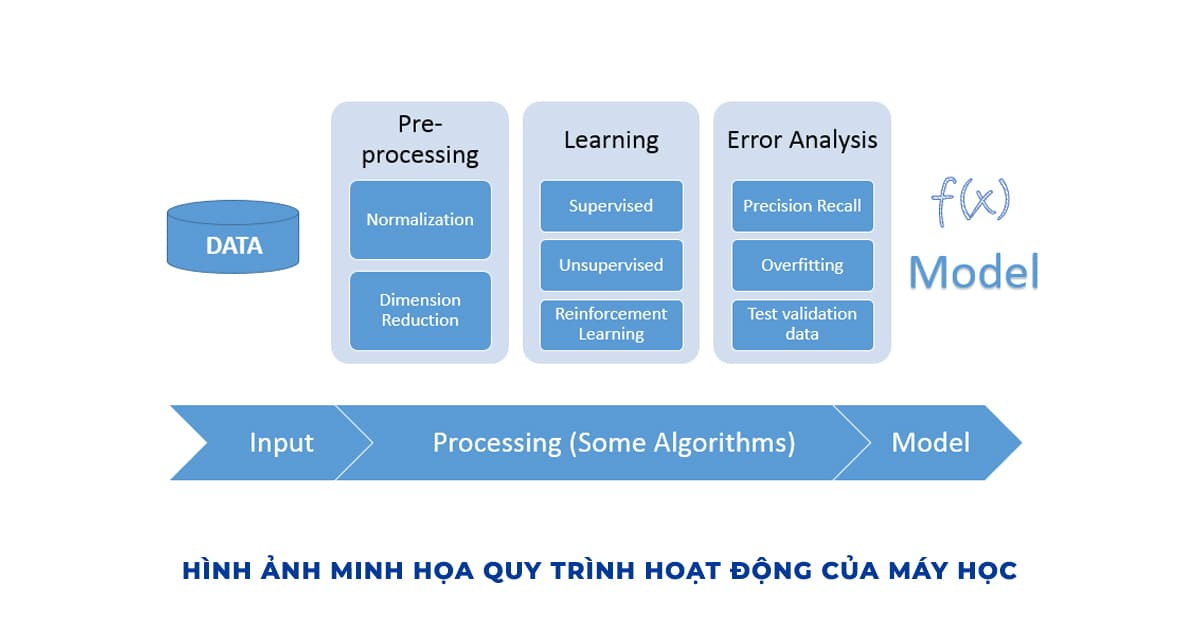
#### 2.1.1 Trí tuệ nhân tạo là gì?

Trí tuệ nhân tạo (AI) là công nghệ cho phép máy tính và máy móc mô phỏng quá trình học tập, hiểu biết, giải quyết vấn đề, ra quyết định, sáng tạo và tự chủ của con người. Các tổ chức hiện đại thu thập khối lượng dữ liệu cực lớn từ nhiều nguồn khác nhau như cảm biến thông minh, nội dung do con người tạo ra, công cụ giám sát và bản ghi hệ thống. Trí tuệ nhân tạo phân tích và sử dụng dữ liệu để hỗ trợ hoạt động kinh doanh một cách hiệu quả.



Hình 2.1. Quá trình thực hiện bài toán AI

#### 2.1.2 Học máy là gì?

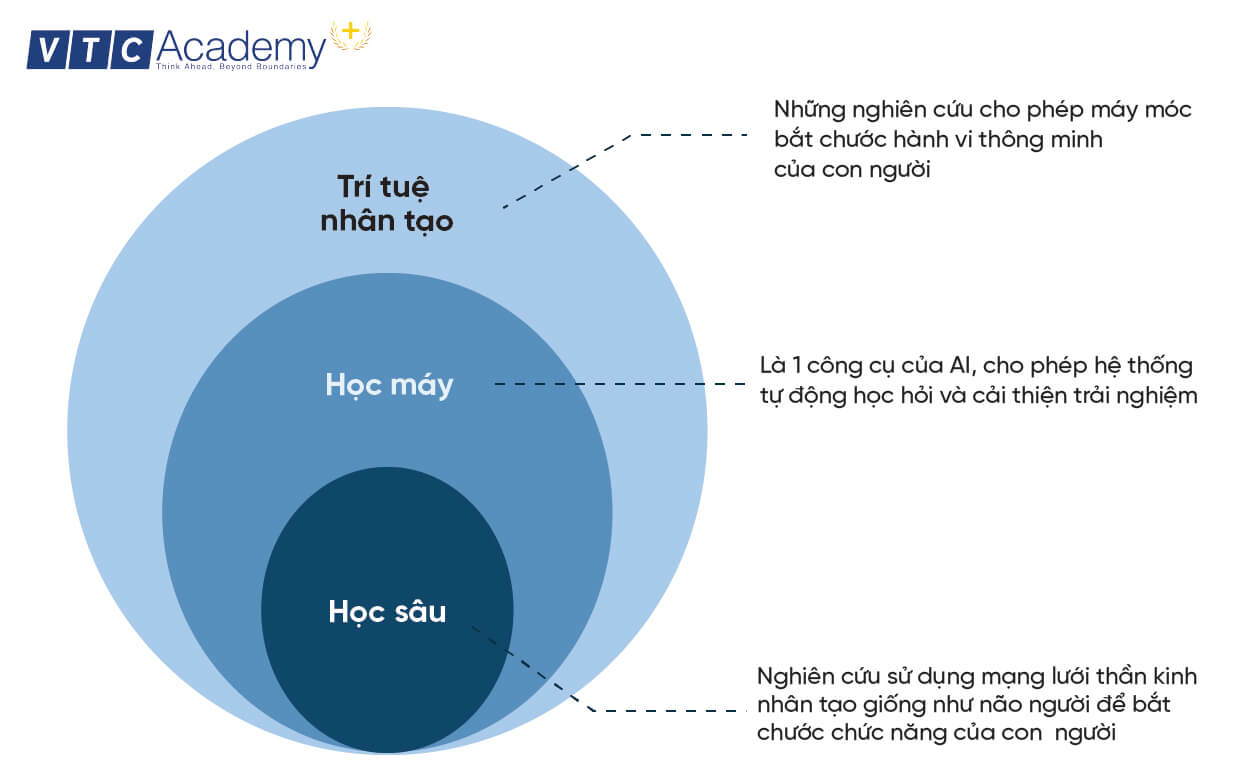
Học máy (Machine Learning)là một nhánh của trí tuệ nhân tạo cho phép các thuật toán khám phá các mẫu ẩn trong bộ dữ liệu để từ đó đưa ra dự đoán về dữ liệu mới. Học máy truyền thống kết hợp dữ liệu với các công cụ thống kê để dự đoán kết quả đầu ra, mang lại những hiểu biết sâu sắc có thể áp dụng được. Máy học được ứng dụng rộng rãi: nhận diện ảnh, giọng nói, đề xuất sản phẩm, phát hiện gian lận… Ví dụ: Facebook dựa vào lịch sử thích, bình luận trên các bài viết hay video, từ đó sẽ hiểu được sở thích của người dùng và đề xuất những nội dung liên quan, cũng như quảng cáo về các nội dung đó. 

Hình 2.2. Quy trình hoạt động của máy học

#### 2.1.3 Học sâu là gì?

Học sâu (Deep Learning) là một tập hợp con của học máy, tập trung vào việc xây dựng và huấn luyện mạng nơ-ron nhiều lớp, được gọi là mạng nơ-ron sâu (DNN – Deep neural networks) để chúng có thể tự động học, hiểu dữ liệu, mô phỏng khả năng ra quyết định phức tạp của bộ não con người.

Mô hình học sâu có thể nhận diện nhiều hình mẫu phức tạp trong hình ảnh, văn bản, âm thanh và các dữ liệu khác để tạo ra thông tin chuyên sâu và dự đoán chính xác. Bạn có thể sử dụng các phương pháp học sâu để tự động hóa các tác vụ thường đòi hỏi trí tuệ con người, chẳng hạn như phân loại hình ảnh hoặc chép lời một tập tin âm thanh.



Hình 2.3. Sự khác biệt giữa AI, Machine Learning, Deep Learning

### Tổng quan về ngôn ngữ và thư viện lập trình

#### 2.2.1 Ngôn ngữ Python

Python là một ngôn ngữ lập trình bậc cao, thông dịch, và đa mục đích, được tạo ra bởi Guido van Rossum và ra mắt lần đầu vào năm 1991. Python nổi tiếng với cú pháp đơn giản, dễ học, và dễ đọc, giúp lập trình viên phát triển phần mềm một cách nhanh chóng và hiệu quả. Python được thiết kế với triết lý "Code rõ ràng hơn là code phức tạp", khiến nó trở thành lựa chọn lý tưởng cho người mới bắt đầu cũng như các lập trình viên giàu kinh nghiệm.

Hiện nay chúng ta biết đến Python là một ngôn ngữ lập trình mạnh mẽ được sử dụng phổ biến trong lĩnh vực AI, ML và Khoa học dữ liệu. Ngoài ra Python cũng được ứng dụng trong các lĩnh vực khác như: Web, Game, phần mềm…



Hình 2.4. Ngôn ngữ lập trình Python

#### 2.2.2 Thư viện TensorFlow

TensorFlow là một thư viện mã nguồn mở do Google phát triển, được sử dụng rộng rãi trong lĩnh vực Machine Learning và Deep Learning. TensorFlow hỗ trợ xây dựng, huấn luyện và triển khai các mô hình học sâu trên nhiều nền tảng như CPU, GPU và TPU.

TensorFlow hoạt động dựa trên đồ thị tính toán (computational graph) và sử dụng tensor làm cấu trúc dữ liệu chính. Thư viện tích hợp API cấp cao Keras, giúp việc xây dựng mô hình mạng nơ-ron trở nên đơn giản và trực quan.

**Ưu điểm:** hiệu năng cao, hỗ trợ GPU, dễ triển khai mô hình CNN, hệ sinh thái phong phú.

**Hạn chế:** yêu cầu tài nguyên mạnh khi huấn luyện mô hình lớn.



Hình 2.5. Ngôn ngữ lập trình Python

### Các công cụ hỗ trợ

#### 2.3.1 Google Colab

Google Colab là một dịch vụ miễn phí từ Google cho phép bạn viết và chia sẻ mã Python. Người dùng có thể lưu trữ và chạy Notebook trên đám mây thông qua trình duyệt mà không cần kích hoạt cấu hình bộ đệm phức tạp cho máy tính cá nhân.



Hình 2.6. Google Colab

Nền tảng cũng cung cấp GPU và TPU để huấn luyện mô hình học và thực hiện các yêu cầu tính toán lớn. Điều này có tác dụng mở rộng sức mạnh xử lý trong điện toán và giảm thời gian chờ đợi khi thực hiện các nhiệm vụ tính toán nặng nề.

Google Colab được phát triển bởi nhóm Google Research nhằm hỗ trợ quá trình nghiên cứu và phát triển trong lĩnh vực học máy tính và khoa học dữ liệu. Công nghệ xuất phát từ nền tảng Jupyter Notebook phổ biến. Sau đó, Colab được kết hợp thêm khả năng tính toán trên tiện ích đám mây vô cùng mạnh mẽ của Google.

Từ đó, quá trình phát triển Google Colab tập trung vào mục tiêu cung cấp một môi trường lập trình linh hoạt và mạnh mẽ. Nền tảng hỗ trợ đa dạng công nghệ tính toán như Python, CUDA, Tensorflow. Kèm theo đó có nhiều thư viện khoa học máy tính và khoa học dữ liệu khác. Đồng thời, dịch cụ còn cung cấp GPU và TPU để tăng cường sức mạnh tính toán.

**\* Các tính năng chính của Google Colab**

* Sử dụng Jupyter Notebooks trực tuyến:Google Colab cho phép tạo và chạy các Jupyter Notebooks trực tuyến mà không cần cài đặt môi trường phát triển phức tạp trên máy tính cá nhân. Giao diện sử dụng tương tự như Jupyter Notebook truyền thống với các cell cho phép thực thi mã Python hoặc viết markdown để tạo nội dung hướng dẫn.
* Khả năng chia sẻ và cộng tác:Người dùng có thể chia sẻ notebook với những người khác để cùng làm việc trên cùng một notebook, tạo điều kiện thuận lợi cho việc học tập và làm việc nhóm. Các tính năng như bình luận và chế độ chỉnh sửa đồng thời giúp tăng tính tương tác và hiệu quả của quá trình cộng tác.
* Dùng GPU và TPU miễn phí:Google Colab cung cấp truy cập miễn phí đến GPU và TPU, đặc biệt hữu ích cho các tác vụ tính toán nặng về mặt số học, đặc biệt là trong lĩnh vực học máy và deep learning. Việc sử dụng các card GPU hoặc TPU có sẵn giúp tăng tốc độ xử lý và huấn luyện mô hình so với việc sử dụng CPU thông thường.
* Lưu trữ dữ liệu trên Google Drive và tích hợp Google Cloud:Người dùng có thể truy cập và lưu trữ dữ liệu trực tiếp từ Google Drive, tạo điều kiện thuận lợi cho việc làm việc với tập tin dữ liệu lớn. Tích hợp với Google Cloud Platform (GCP) cũng cho phép sử dụng các dịch vụ như BigQuery, Cloud Storage, và các API khác từ GCP trong quá trình làm việc

**\* Ứng dụng của Google Colab**

- Học máy và khoa học dữ liệu:Google Colab là một công cụ mạnh mẽ cho việc học máy và nghiên cứu khoa học dữ liệu. Với khả năng sử dụng miễn phí GPU và TPU, người dùng có thể xây dựng, huấn luyện và kiểm định các mô hình máy học một cách hiệu quả. Google Colab cung cấp môi trường Jupyter Notebook trực tuyến, cho phép viết và chạy code Python, đồng thời cung cấp các thư viện phổ biến như TensorFlow, Keras, và scikit-learn.

- Phát triển ứng dụng AI và ML:Với sức mạnh của Colab's GPU và TPU, các nhà phát triển có thể tận dụng để phát triển ứng dụng trí tuệ nhân tạo (AI) và máy học (ML). Điều này bao gồm việc xây dựng và đánh giá các mô hình học máy, tạo ứng dụng dự đoán, xử lý ngôn ngữ tự nhiên, nhận diện hình ảnh và nhiều ứng dụng AI khác.

- Nghiên cứu và phân tích dữ liệu:Các nhà nghiên cứu và chuyên gia phân tích dữ liệu thường sử dụng Google Colab để thực hiện phân tích số liệu, xử lý dữ liệu lớn, và thực hiện các thí nghiệm khoa học. Colab cung cấp khả năng tích hợp dữ liệu từ Google Drive hoặc các nguồn khác, cho phép họ thực hiện các phân tích phức tạp mà không cần tải về dữ liệu về máy cục bộ.

Giáo dục và đào tạo:Google Colab cũng được sử dụng rộng rãi trong lĩnh vực giáo dục và đào tạo. Nó cung cấp môi trường lập trình Python trực tuyến mà không yêu cầu cài đặt, giúp sinh viên và giáo viên dễ dàng tiếp cận và chia sẻ các notebook học tập. Điều này làm tăng tính tương tác và khả năng học tập cộng đồng trong việc chia sẻ kiến thức và dự án.

#### 2.3.2 FastAPI

FastAPI là một framework hiện đại dùng để xây dựng ứng dụng Web API trong Python. Được phát triển dựa trên chuẩn ASGI (Asynchronous Server Gateway Interface), FastAPI cung cấp tốc độ xử lý rất cao nhờ cơ chế bất đồng bộ (async/await) của Python, giúp nó trở thành lựa chọn hàng đầu cho các hệ thống AI cần xử lý số lượng lớn truy vấn.

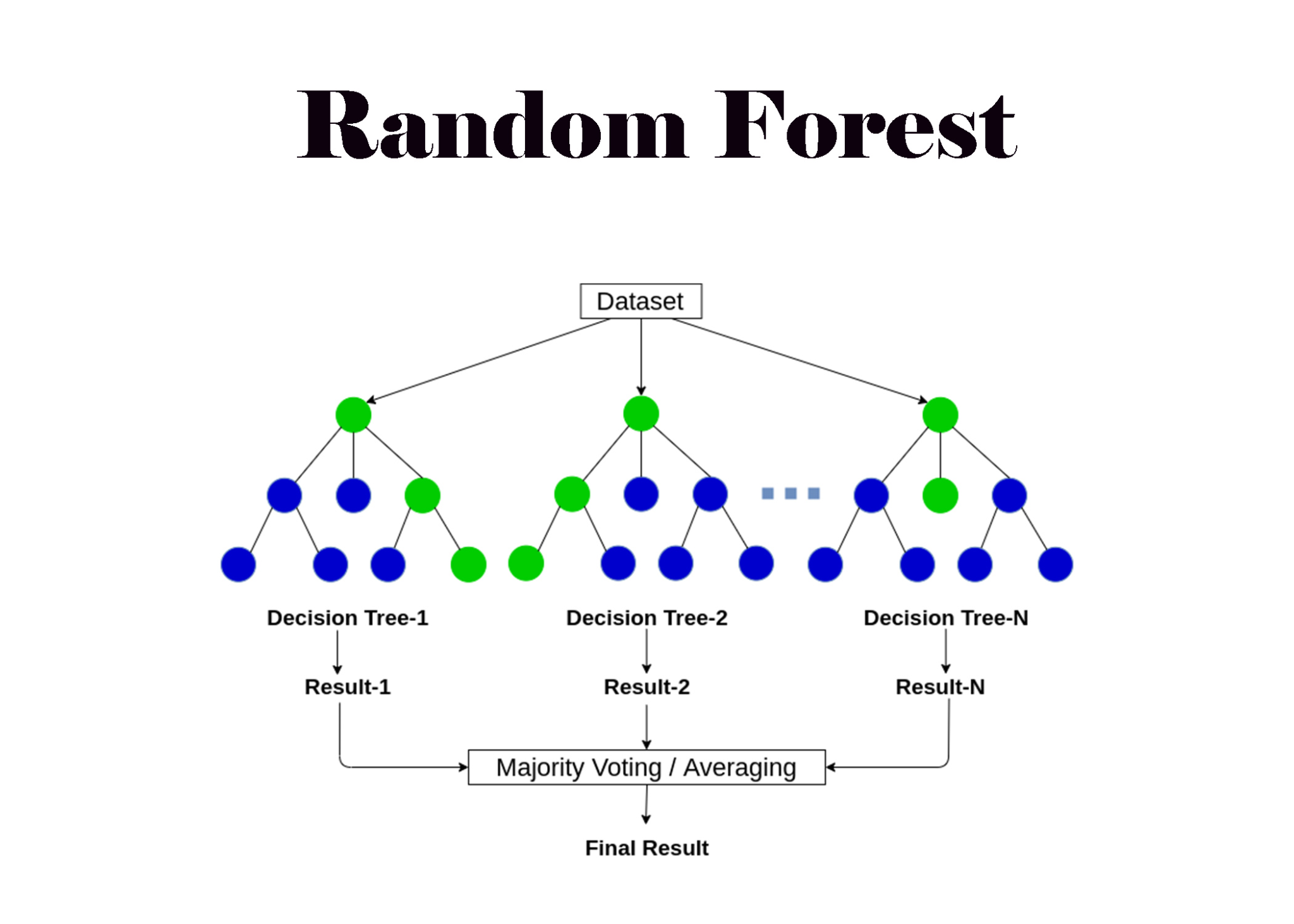
FastAPI tích hợp tự động việc kiểm tra kiểu dữ liệu (data validation) thông qua thư viện Pydantic và tự sinh tài liệu API bằng Swagger UI và ReDoc. Điều này hỗ trợ mạnh mẽ cho quá trình phát triển, kiểm thử và triển khai ứng dụng AI.

### Các thuật toán và mô hình áp trong đề tài

#### 2.4.1 Random Forest

Random Forest là một thuật toán học máy mạnh mẽ thuộc nhóm thuật toán học giám sát, được sử dụng để giải quyết các bài toán phân loại và hồi quy. Nó là một sự mở rộng của cây quyết định, nhưng có một số cải tiến giúp tăng độ chính xác và giảm hiện tượng overfitting.

Những cái cây trong rừng ngẫu nhiên là những cây quyết định nhưng nó sẽ khác nhau vì các cây được tạo ra bằng yếu tố random. Kết quả dự đoán cuối cùng của rừng ngẫu nhiên sẽ là tổng hợp kết quả dự đoán từ các cây trong rừng theo 2 cách là: bỏ phiếu hoặc là trung bình.



Hình 2.7. Ví dụ thuật toán Random Forest

**Ưu điểm:**

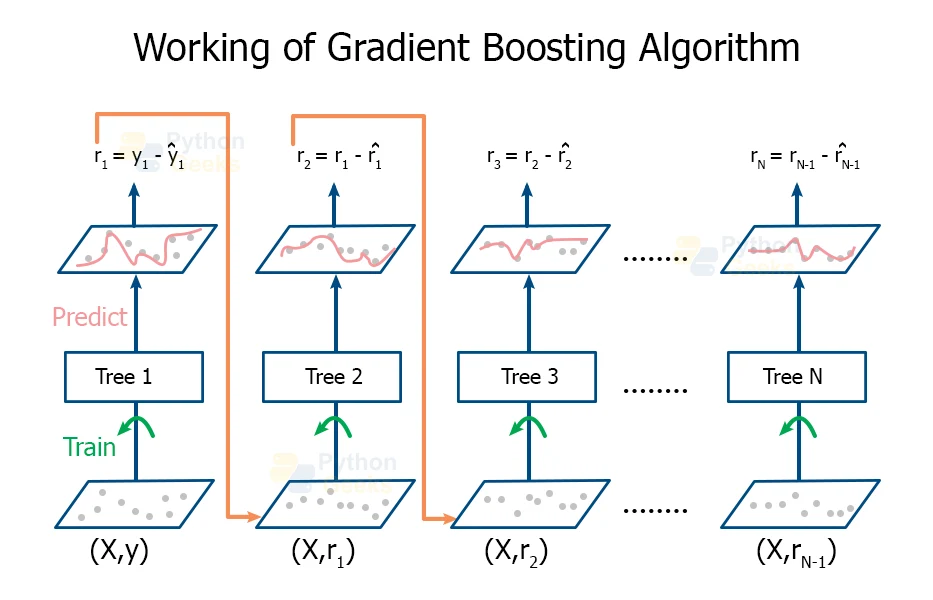
* Giảm overfitting so với Decision Tree.
* Hiệu năng tốt với nhiều loại dữ liệu.

**Nhược điểm:**

* Cần nhiều tài nguyên để tính toán.

#### 2.4.2 Gradient Boosting

Gradient Boosting là một thuật toán mạnh mẽ dựa trên ý tưởng tối ưu gradient. Nó xây dựng một mô hình dự đoán bằng cách kết hợp nhiều mô hình yếu (thường là cây quyết định) qua từng bước. Ở mỗi bước, thuật toán cố gắng giảm thiểu lỗi dự đoán bằng cách tối ưu hóa một hàm mất mát. Gradient Boosting phổ biến trong nhiều lĩnh vực như tài chính, y tế, và phân tích dữ liệu lớn.



Hình 2.8. Ví dụ thuật toán Gradient Boosting

**Ưu điểm:**

* Hiệu suất cao trên cả dữ liệu tuyến tính và phi tuyến tính.
* Có thể xử lý dữ liệu mất mát tốt bằng cách tùy chỉnh hàm mất mát.
* Hỗ trợ tối ưu hóa linh hoạt với nhiều dạng bài toán.

**Nhược điểm:**

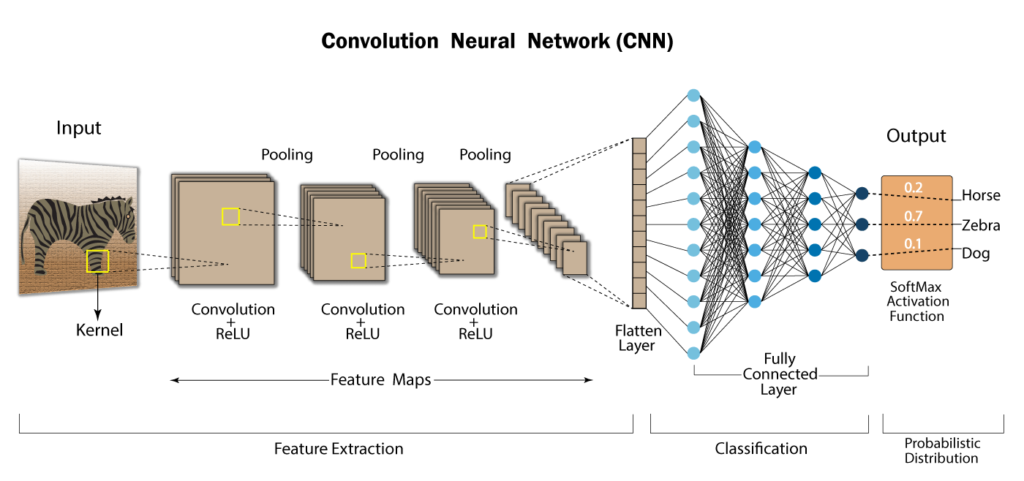
* Tốn tài nguyên tính toán và thời gian huấn luyện.
* Dễ bị overfitting nếu không tinh chỉnh tốt.

#### 2.4.3 Mạng nơ-ron tích chập (CNN)

Mạng nơ-ron tích chập – CNN là một trong những kiến trúc Deep Learning phổ biến nhất, được thiết kế chuyên biệt cho xử lý ảnh, video và dữ liệu dạng lưới (grid). CNN hoạt động dựa trên việc sử dụng các lớp convolution để tự động học đặc trưng (feature) từ dữ liệu hình ảnh.

Cấu trúc cơ bản của CNN:

* Lớp tích chập (Convolutional Layer): Đây là lớp quan trọng nhất trong CNN. Lớp này thực hiện phép tích chập giữa đầu vào và các bộ lọc để trích xuất các đặc trưng. Mỗi bộ lọc có thể phát hiện một loại đặc trưng cụ thể như cạnh ngang, cạnh dọc hoặc các cấu trúc phức tạp hơn ở các lớp sâu hơn. Kết quả của lớp tích chập là một tập hợp các bản đồ đặc trưng.
* Lớp kích hoạt (Activation Layer): Lớp này áp dụng một hàm phi tuyến, thường là ReLU, lên kết quả của lớp tích chập. ReLU chuyển đổi tất cả giá trị âm thành 0 trong khi giữ nguyên các giá trị dương. Hàm ReLU giúp mô hình học được các mối quan hệ phức tạp và phi tuyến trong dữ liệu, đồng thời giải quyết vấn đề gradient biến mất khi huấn luyện mạng sâu.
* Lớp gộp (Pooling Layer): Lớp này giảm kích thước không gian của bản đồ đặc trưng, giảm số lượng tham số và tính toán trong mạng. Max Pooling là phương pháp phổ biến nhất, lấy giá trị lớn nhất trong một cửa sổ. Pooling cũng giúp mô hình ít nhạy cảm hơn với vị trí chính xác của đặc trưng, tăng khả năng tổng quát hóa.
* Lớp kết nối đầy đủ (Fully-Connected Layer): Ở cuối CNN, các bản đồ đặc trưng được làm phẳng thành một vector và kết nối với một hoặc nhiều lớp kết nối đầy đủ. Các lớp này hoạt động giống như mạng nơ-ron truyền thống, kết hợp mọi đặc trưng đã học được để đưa ra quyết định cuối cùng, như phân loại đối tượng trong ảnh.



Hình 2.9. Ví dụ về mạng tích chập CNN

Quy trình hoạt động của CNN bắt đầu bằng việc áp dụng các bộ lọc tích chập để tạo ra bản đồ đặc trưng, sau đó kết quả đi qua hàm kích hoạt để học các quan hệ phi tuyến. Tiếp đến, mô hình sử dụng pooling (thường là Max Pooling) để giảm kích thước và tăng tính bền vững với nhiễu.

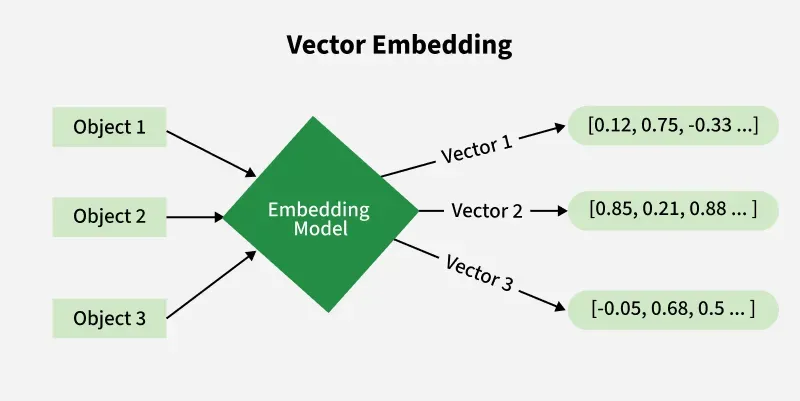
Ở cuối mạng, dữ liệu được làm phẳng thành vector và đưa qua các lớp kết nối đầy đủ để suy luận và phân loại. Toàn bộ quá trình học diễn ra nhờ thuật toán lan truyền ngược (Backpropagation), giúp CNN liên tục điều chỉnh trọng số để giảm sai số và cải thiện khả năng nhận diện mẫu.

#### 2.4.4 Embedding (Biểu diễn vector)

Embedding là một phương pháp biểu diễn dữ liệu (hình ảnh, văn bản...) dưới dạng vector nhiều chiều trong không gian liên tục. Với ảnh, embedding là đầu ra của CNN sau khi loại bỏ tầng phân loại. Mỗi embedding chứa các thông tin phản ánh đặc trưng nội dung của ảnh.

Ví dụ, một ảnh áo thun sau khi qua CNN có thể được biểu diễn thành một vector 1280 chiều:

[0.12, 0.75, -0.33, ...]

****

Hình 2.10. Ví dụ về Vector Embedding

#### 2.4.5 FAISS

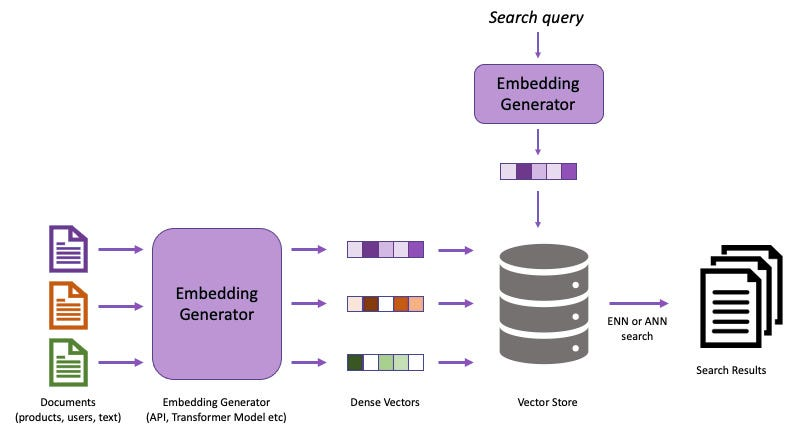
FAISS là thư viện tối ưu hóa do Facebook AI Research phát triển, chuyên dùng để tìm kiếm similarity (độ tương tự) giữa các vector embedding trong tập dữ liệu lớn. FAISS cho phép tìm kiếm K nearest neighbors (KNN) cực kỳ nhanh nhờ các cấu trúc index đặc biệt và tối ưu hóa hiệu năng CPU/GPU.

**Ưu điểm:**

* Tìm kiếm vector rất nhanh, phù hợp cho hàng nghìn đến hàng triệu embedding
* Hỗ trợ nhiều loại index: Flat (chính xác cao), IVF (nhanh hơn với dữ liệu lớn)
* Tích hợp tốt với cosine similarity hoặc L2 distance

**Vai trò của FAISS trong đề tài**

* Lưu trữ toàn bộ embedding của ảnh sản phẩm
* Khi người dùng tải ảnh lên:
* Ảnh → CNN → Embedding
* Embedding → FAISS → trả về danh sách sản phẩm giống nhất
* Tạo thành module tìm kiếm sản phẩm tương tự theo hình ảnh

****

Hình 2.11. Quy trình hoạt động của hệ thống tìm kiếm dựa trên Embedding

# CHƯƠNG 3. GIỚI THIỆU BỘ DỮ LIỆU

### Tổng quan

#### 3.1.1 Bộ dữ liệu 1

Tên bộ dữ liệu: Retailrocket recommender system dataset

Nguồn: Kaggle

Các đặc trưng của tập dữ liệu

|  |  |  |
| --- | --- | --- |
| **Tên đặc trưng** | **Kiểu dữ liệu** | **Mô Tả** |
| Timestamp | int | Thời điểm sự kiện xảy ra, dưới dạng Unix timestamp. Cần chuyển về dạng datetime để phân tích theo ngày/giờ. |
| Visitorid | int | ID người dùng. Một user có thể có nhiều hành vi trong suốt phiên làm việc. |
| Event | object | Loại hành vi. Các giá trị thường thấy:  + view : xem sản phẩm  + addtocart: thêm vào giỏ hàng  + transaction: mua hàng |
| Itemid | Float | ID sản phẩm mà người dùng tương tác (xem / thêm giỏ / mua). |
| Transactionid | int | ID giao dịch. Chỉ có giá trị khi event = transaction. Các event khác để NaN. |

Bảng 3.1 Các đặc trưng của tập dữ liệu 1

#### 3.1.2 Bộ dữ liệu 2

Tên bộ dữ liệu: Fashion Product Images

Nguồn: Kaggle

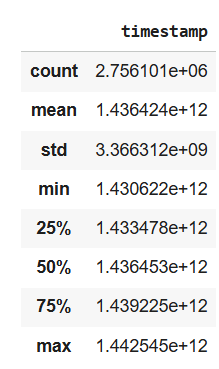
Các đặc trưng của tập dữ liệu

|  |  |  |
| --- | --- | --- |
| **Tên đặc trưng** | **Kiểu dữ liệu** | **Mô tả** |
| id | int | ID sản phẩm duy nhất. |
| gender | Object | Giới tính của đối tượng sử dụng sản phẩm. |
| masterCategory | Object | Danh mục sản phẩm cấp cao. |
| subCategory | Object | Danh mục sản phẩm phụ. |
| articleType | Object | Loại mặt hàng cụ thể. |
| baseColour | Object | Màu cơ bản của sản phẩm. |
| season | Object | Mùa được thiết kế cho sản phẩm. |
| year | int | Năm sản phẩm được phát hành hoặc liệt kê. |
| usage | Object | Mục đích sử dụng của sản phẩm. |
| productDisplayName | Object | Tên đầy đủ, dễ đọc của sản phẩm. |

Bảng 3.2 Các đặc trưng của tập dữ liệu 2

### Đặc điểm của bộ dữ liệu 1

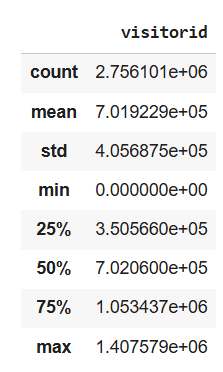
#### 3.2.1 Mô tả thuộc tính Timestamp

****

Hình 3.1 Mô tả thuộc tính timestamp

* Số lượng dữ liệu (count): Có 2,756,101 mẫu dữ liệu trong thuộc tính.
* Giá trị trung bình (mean): Giá trị trung bình của timestamp là 1.436424 \* 1012.
* Độ lệch chuẩn (std): Độ lệch chuẩn là 3.366312 \* 109, cho thấy sự biến động đáng kể về thời gian giữa các sự kiện được ghi nhận.
* Giá trị nhỏ nhất (min): Dấu thời gian nhỏ nhất là 1.430622 \* 1012.
* Tứ phân vị thứ nhất (25%): 25% các dấu thời gian nhỏ hơn hoặc bằng 1.433478 \* 1012.
* Trung vị (50%): 50% các dấu thời gian nhỏ hơn hoặc bằng 1.436453 \* 1012. Đây là giátrị giữa khi sắp xếp dữ liệu theo thứ tự tăng dần.
* Tứ phân vị thứ ba (75%): 75% các dấu thời gian nhỏ hơn hoặc bằng 1.439225 \* 1012.
* Giá trị lớn nhất (max): Giá trị lớn nhất (muộn nhất) của timestamp là 1.442545 \* 1012.

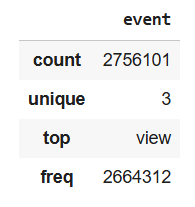
#### 3.2.2 Mô tả thuộc tính visitorid

**

Hình 3.2 Mô tả thuộc tính visitorid

* Số lượng dữ liệu (count): Có 2,756,101 mẫu dữ liệu trong thuộc tính.
* Giá trị trung bình (mean): Giá trị trung bình của ID khách truy cập là 701,922.9.
* Độ lệch chuẩn (std): Độ lệch chuẩn là 405,687.5, cho thấy sự phân bố rộng của các ID khách truy cập.
* Giá trị nhỏ nhất (min): ID khách truy cập nhỏ nhất là 0.
* Tứ phân vị thứ nhất (25%): 25% giá trị ID khách truy cập nhỏ hơn hoặc bằng 350,566.
* Trung vị (50%): 50% giá trị ID khách truy cập nhỏ hơn hoặc bằng 702,060.
* Tứ phân vị thứ ba (75%): 75% giá trị ID khách truy cập nhỏ hơn hoặc bằng 1,053,437.
* Giá trị lớn nhất (max): Giá trị lớn nhất của ID khách truy cập là 1,407,579.

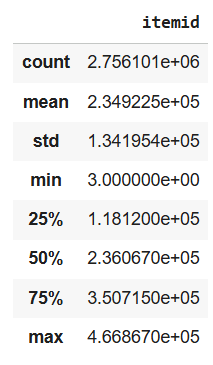
#### 3.2.3 Mô tả thuộc tính event

****

Hình 3.3 Mô tả thuộc tính event

* Tổng số lượng dữ liệu có trong thuộc tính: 2756101 mẫu
* Số lượng giá trị duy nhất có thuộc tính: 3
* Giá trị xuất hiện phổ biến nhất trong thuộc tính: view
* Số lần giá trị phổ biến xuất hiện: 2664312.

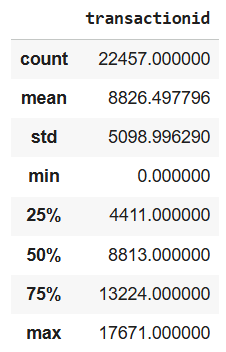
#### 3.2.4 Mô tả thuộc tính itemid

**

Hình 3.4 Mô tả thuộc tính itemid

* Số lượng dữ liệu (count): Có 2,756,101 mẫu dữ liệu trong thuộc tính
* Giá trị trung bình (mean): Giá trị trung bình của ID mặt hàng là 234,922.5.
* Độ lệch chuẩn (std): Độ lệch chuẩn là 134,195.4, cho thấy sự biến động lớn trong các giá trị ID mặt hàng, phù hợp với một thuộc tính định danh.
* Giá trị nhỏ nhất (min): ID mặt hàng nhỏ nhất là 3.
* Tứ phân vị thứ nhất (25%) 25% giá trị ID mặt hàng nhỏ hơn hoặc bằng 118,120
* Trung vị (50%): 50% giá trị ID mặt hàng nhỏ hơn hoặc bằng 236,067
* Tứ phân vị thứ ba (75%): 75% giá trị ID mặt hàng nhỏ hơn hoặc bằng 350,715
* Giá trị lớn nhất (max): Giá trị lớn nhất của ID mặt hàng là 466,867

#### 3.2.5 Mô tả thuộc tính transactionid

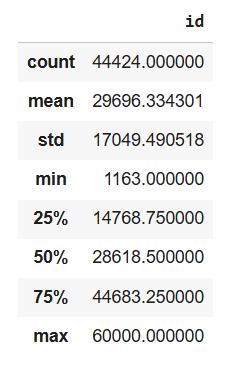
**

Hình 3.5 Mô tả thuộc tính transactionid

* Số lượng dữ liệu (count): Có 22,457 mẫu dữ liệu có ID giao dịch
* Giá trị trung bình (mean): Giá trị trung bình của ID giao dịch là $8,826.498$.
* Độ lệch chuẩn (std): Độ lệch chuẩn là 5,098.996.
* Giá trị nhỏ nhất (min): ID giao dịch nhỏ nhất là 0.
* Tứ phân vị thứ nhất (25%): 25% giá trị ID giao dịch nhỏ hơn hoặc bằng 4,411.
* Trung vị (50%): 50% giá trị ID giao dịch nhỏ hơn hoặc bằng 8,813.
* Tứ phân vị thứ ba (75%): 75% giá trị ID giao dịch nhỏ hơn hoặc bằng 13,224.
* Giá trị lớn nhất (max): Giá trị lớn nhất của ID giao dịch là 17,671.

### Đặc điểm của bộ dữ liệu 2

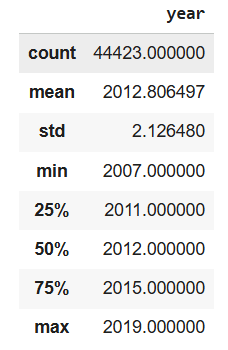
#### 3.3.1 Mô tả thuộc tính id



Hình 3.6 Mô tả thuộc tính id

* Số lượng dữ liệu (count): Có 44,424 mẫu dữ liệu trong thuộc tính.
* Giá trị trung bình (mean): Giá trị trung bình của ID là 29,696.33
* Độ lệch chuẩn (std): Độ lệch chuẩn là 17,049.49, cho thấy độ biến động rất lớn trong các giá trị ID. Điều này là hoàn toàn bình thường và được mong đợi đối với một thuộc tính định danh (ID) có phạm vi rộng
* Giá trị nhỏ nhất (min): ID nhỏ nhất là 1,163.00
* Tứ phân vị thứ nhất (25%): 25% giá trị ID nhỏ hơn hoặc bằng 14,768.75
* Trung vị (50%): 50% giá trị ID nhỏ hơn hoặc bằng 28,618.50. Đây là giá trị nằm chính giữa khi sắp xếp dữ liệu theo thứ tự tăng dần
* Tứ phân vị thứ ba (75%): 75% giá trị ID nhỏ hơn hoặc bằng 44,683.25
* Giá trị lớn nhất (max): Giá trị lớn nhất của ID là 60,000.00

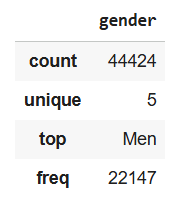
#### 3.3.2 Mô tả thuộc tính year

****

Hình 3.7 Mô tả thuộc tính year

* Số lượng dữ liệu (count): Có 44,423 mẫu dữ liệu trong thuộc tính
* Giá trị trung bình (mean): Giá trị trung bình của year là 2012.81
* Độ lệch chuẩn (std): Độ lệch chuẩn là 2.13, cho thấy độ biến động rất thấp. Các giá trị năm rất tập trung và đồng nhất xung quanh giá trị trung bình
* Giá trị nhỏ nhất (min): year nhỏ nhất là 2007
* Tứ phân vị thứ nhất (25%): 25% giá trị year nhỏ hơn hoặc bằng 2011
* Trung vị (50%): 50% giá trị year nhỏ hơn hoặc bằng 2012.00. Đây là năm nằm giữa khi sắp xếp dữ liệu
* Tứ phân vị thứ ba (75%): 75% giá trị year nhỏ hơn hoặc bằng 2015.00
* Giá trị lớn nhất (max): Giá trị lớn nhất của year là 2019.00

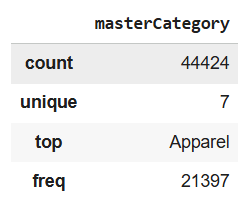
#### 3.3.3 Mô tả thuộc tính gender

****

Hình 3.8 Mô tả thuộc tính gender

* Tổng số mẫu dữ liệu có trong thuộc tính: 44,424 mẫu
* Số lượng giá trị duy nhất của thuộc tính: 5
* Giá trị xuất hiện phổ biến nhất trong thuộc tính: Men
* Số lần giá trị phổ biến nhất xuất hiện: 22,147 lần

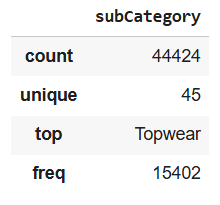
#### 3.3.4 Mô tả thuộc tính *masterCategory*

**

Hình 3.9 Mô tả thuộc tính masterCategory

* Tổng số mẫu dữ liệu có trong thuộc tính: 44,424 mẫu
* Số lượng giá trị duy nhất của thuộc tính: 7
* Giá trị xuất hiện phổ biến nhất trong thuộc tính: Apparel
* Số lần giá trị phổ biến nhất xuất hiện: 21397 lần

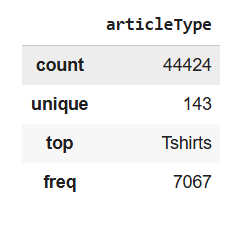
#### 3.3.5 Mô tả thuộc tính subCategory

**

Hình 3.10 Mô tả thuộc tính subCategory

* Tổng số mẫu dữ liệu có trong thuộc tính: 44,424 mẫu
* Số lượng giá trị duy nhất của thuộc tính: 45
* Giá trị xuất hiện phổ biến nhất trong thuộc tính: Topwear
* Số lần giá trị phổ biến nhất xuất hiện: 15402 lần

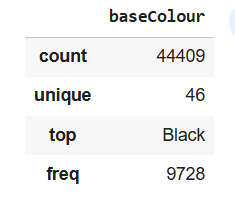
#### 3.3.6 Mô tả thuộc tính articleType

**

Hình 3.11 Mô tả thuộc tính articleType

* Tổng số mẫu dữ liệu có trong thuộc tính: 44,424 mẫu
* Số lượng giá trị duy nhất của thuộc tính: 143
* Giá trị xuất hiện phổ biến nhất trong thuộc tính: Tshirts
* Số lần giá trị phổ biến nhất xuất hiện: 7076 lần

#### 3.3.7 Mô tả thuộc tính baseColour



Hình 3.12 Mô tả thuộc tính baseColour

* Tổng số mẫu dữ liệu có trong thuộc tính: 44,424 mẫu
* Số lượng giá trị duy nhất của thuộc tính: 46
* Giá trị xuất hiện phổ biến nhất trong thuộc tính: Black
* Số lần giá trị phổ biến nhất xuất hiện: 9728 lần

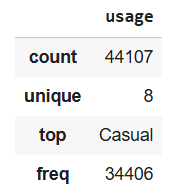
#### 3.3.8 Mô tả thuộc tính season



Hình 3.13 Mô tả thuộc tính season

* Tổng số mẫu dữ liệu có trong thuộc tính: 44,424 mẫu
* Số lượng giá trị duy nhất của thuộc tính: 4
* Giá trị xuất hiện phổ biến nhất trong thuộc tính: Summer
* Số lần giá trị phổ biến nhất xuất hiện: 21472 lần

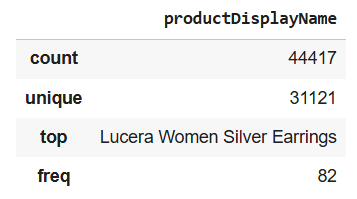
#### 3.3.9 Mô tả thuộc tính usage

**

Hình 3.14 Mô tả thuộc tính usage

* Tổng số mẫu dữ liệu có trong thuộc tính: 44,424 mẫu
* Số lượng giá trị duy nhất của thuộc tính: 8
* Giá trị xuất hiện phổ biến nhất trong thuộc tính: Casual
* Số lần giá trị phổ biến nhất xuất hiện: 34406 lần

#### 3.3.10 Mô tả thuộc tính productDisplayName

**

Hình 3.15 Mô tả thuộc tính productDisplayName

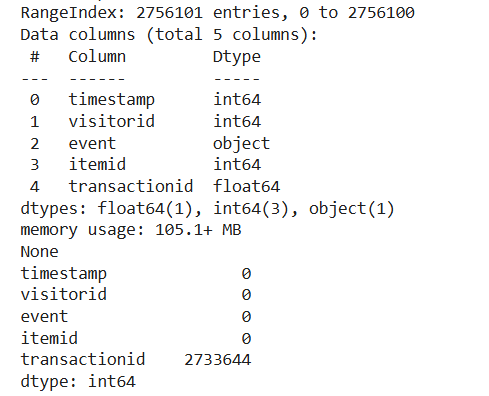
* Tổng số mẫu dữ liệu có trong thuộc tính: 44,424 mẫu
* Số lượng giá trị duy nhất của thuộc tính: 31121
* Giá trị xuất hiện phổ biến nhất trong thuộc tính: Lucera Women Silver Earrings
* Số lần giá trị phổ biến nhất xuất hiện: 82 lần

# CHƯƠNG 4. THỰC NGHIỆM VÀ KẾT QUẢ THU ĐƯỢC

### 4.1 Cài đặt môi trường thực nghiệm

#### 4.1.1 Các bước tiền xử lí cơ bản

Thông tin tổng quan dữ liệu

****

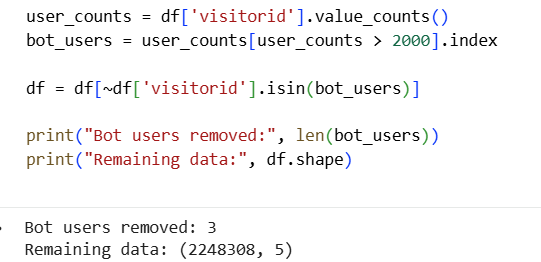
Hình 4.1 Thông tin tổng quan ban đầu

Loại bỏ dữ liệu trùng lặp

****

Hình 4.2 Dữ liệu còn lại sau khi sử lí trùng lặp

Lọc bỏ tài khoản nghi ngờ là bot



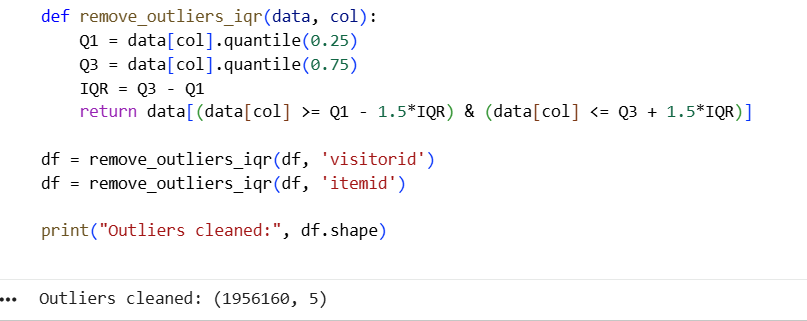
Hình 4.3 Dữ liệu còn lại sau khi lọc

Loại bỏ chuỗi sự kiện không hợp lệ



Hình 4.4 Dữ liệu còn lại khi lọc bỏ chuỗi sự kiện không hợp lệ

Xử lí Outliers

****

Hình 4.5 Dữ liệu sau khi xử lí Outliers

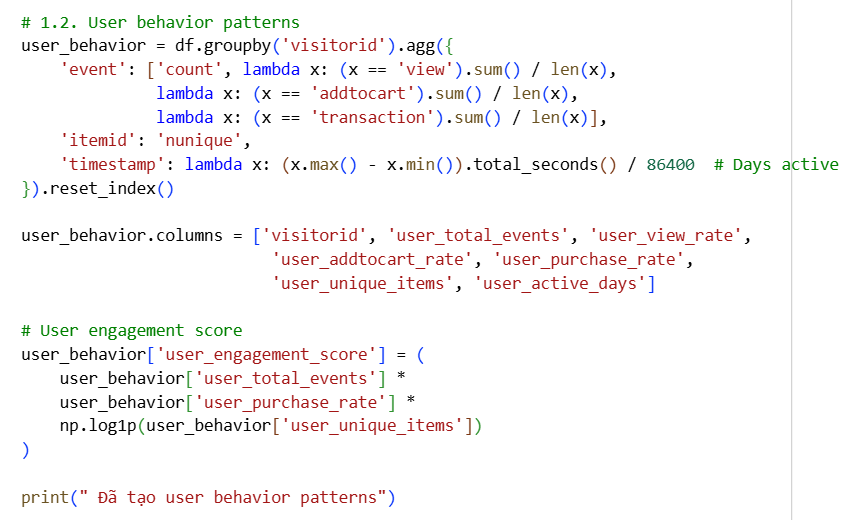
#### 4.1.2 Các bước trong tiền xử lí nâng cao

Lịch sử tương tác giữa user và item

****

Hình 4.6 Lịch sử tương tác giữa user và item

Mô hình hành vi của người dùng

****

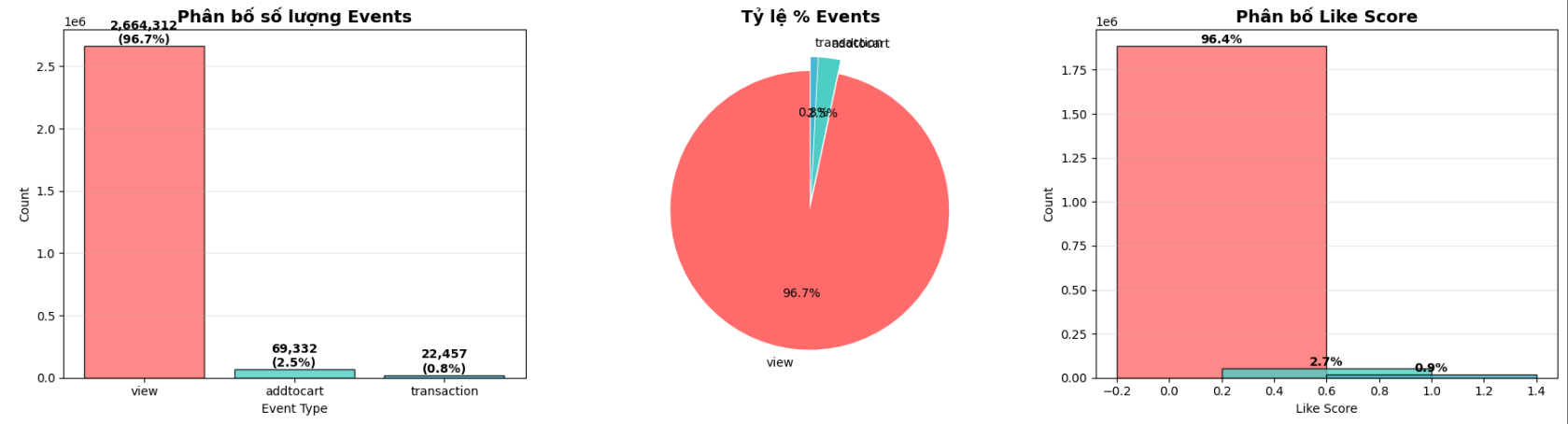
Hình 4.7 Mô tả hành vi người dùng

Phân tích phiên hoạt động của user

****

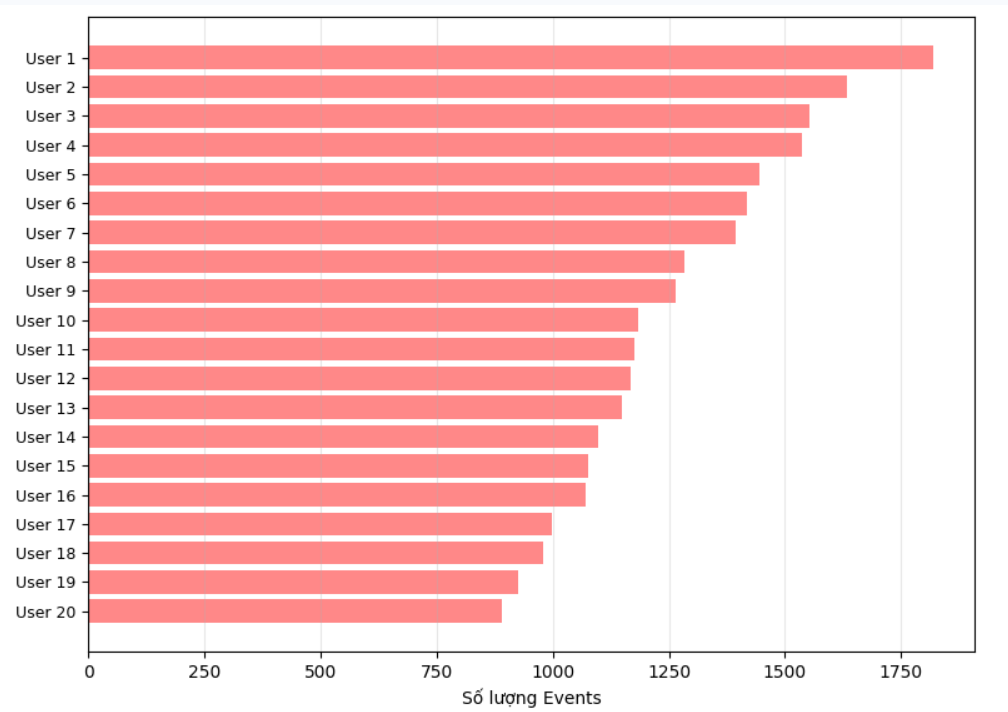
Hình 4.8 Phân tích hoạt động của user

Biểu đồ phân bố biến mục tiêu

****

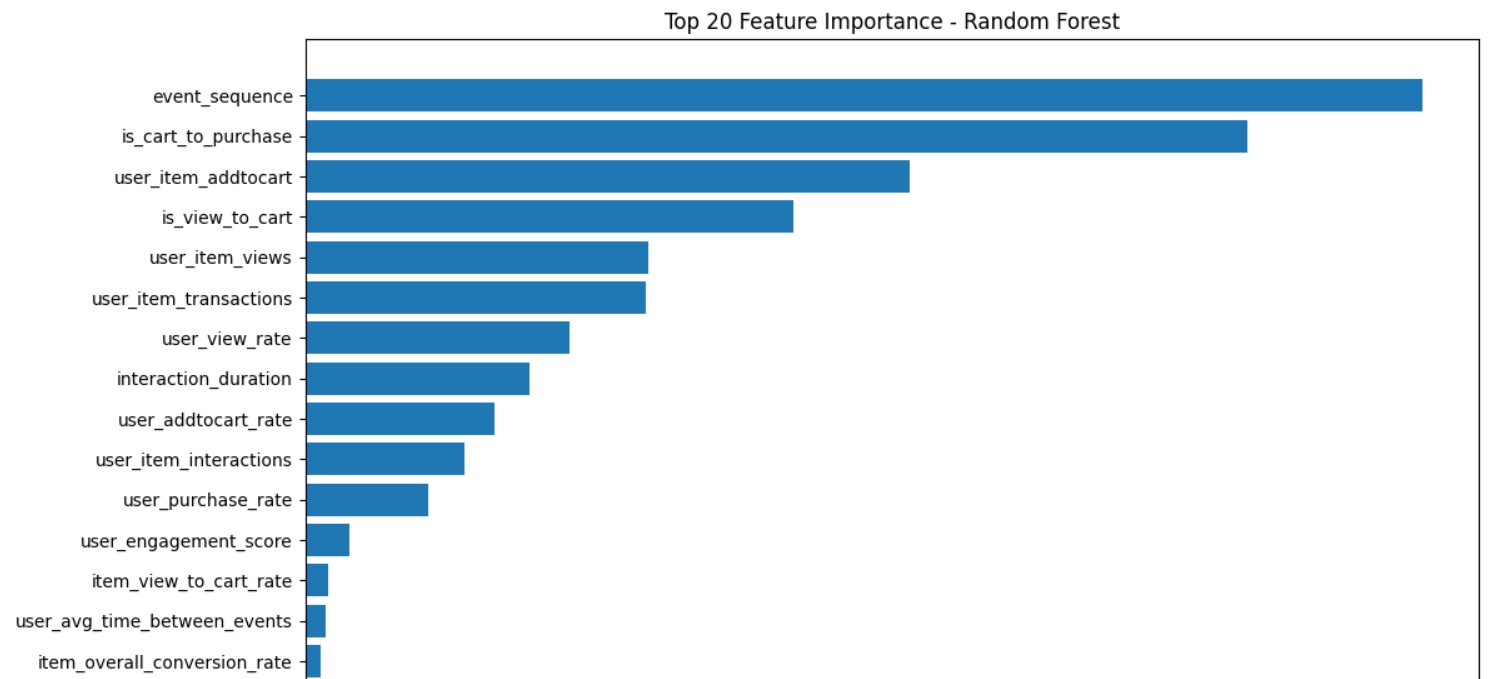
Hình 4.9 Phân bố biến mục tiêu

Biểu đồ phân bố các user hoạt động nhiều nhất

****

Hình 4.10 Các user hoạt động nhiều nhất

Biểu đồ phân bố các feature quan trọng

****

Hình 4.11 Các feature quan trọng

### 4.2 Kết quả đạt được

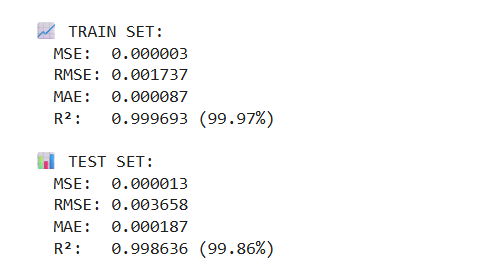
Kết quả tổng hợp của từng mô hình được trình bày dưới đây.

|  |  |
| --- | --- |
| **Mô hình** | **Độ chính xác** |
| Random Forest Regressor | 99,99% |
| Gradient Boosting Regressor | 99,99% |

Bảng 4.1. Độ chính xác của các mô hình sau khi huấn luyện

### 4.3 Đánh giá các mô hình

#### 4.3.1 Random Forest Regressor



Hình 4.12 Kết quả Random Forest

Random Forest đạt:

* Train R² = 0.99
* Test R² = 0.99
* Test RMSE = 0.0036
* Test MAE = 0.00018

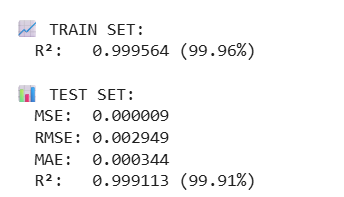
Đây là kết quả rất cao, cho thấy mô hình học rất tốt quan hệ phi tuyến giữa các thuộc tính. Sai số dự đoán thấp và R² gần tuyệt đối cho thấy mô hình giải thích gần như toàn bộ phương sai của dữ liệu.

Một số nhận xét:

* Sai số trên tập kiểm tra chỉ tăng nhẹ so với tập huấn luyện, chứng tỏ mô hình không bị overfitting nghiêm trọng nhờ cơ chế bagging.
* Điểm mạnh của Random Forest là khả năng xử lý dữ liệu lớn và nhiều biến với độ ổn định cao.
* OOB Score đạt 0.99 khẳng định sự nhất quán của mô hình.

Random Forest phù hợp cho bài toán dự đoán các chỉ số thương mại điện tử vì dữ liệu có tính phi tuyến và chứa nhiều nhiễu.

#### 4.3.2 Gradient Boosting Regressor



Hình 4.13 Kết quả Gradient Boosting Regressor

Gradient Boosting là mô hình hoạt động tốt nhất trong nhóm hồi quy của bộ dữ liệu này. Kết quả:

* Train R² = 0.99
* Test R² = 0.99
* Test RMSE = 0.0029
* Test MAE = 0.000344

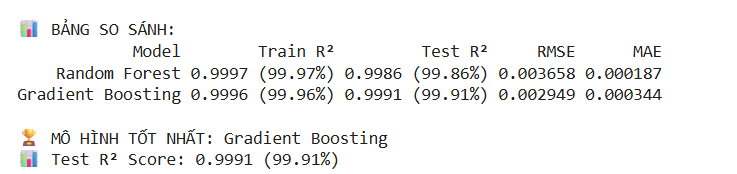
Điểm đáng chú ý:

* Gradient Boosting học sâu vào phần dư (residual), giúp mô hình tối ưu hóa các sai lệch nhỏ mà Random Forest khó bắt được.
* Mức độ sai số thấp và R² cao cho thấy mô hình dự đoán rất tốt trên cả giá trị lớn và nhỏ.
* Với GPU, thời gian huấn luyện được rút ngắn đáng kể, giúp mô hình phù hợp trong bối cảnh dữ liệu lớn.

Kết quả chứng minh Gradient Boosting là mô hình hồi quy mạnh nhất trong thí nghiệm này.

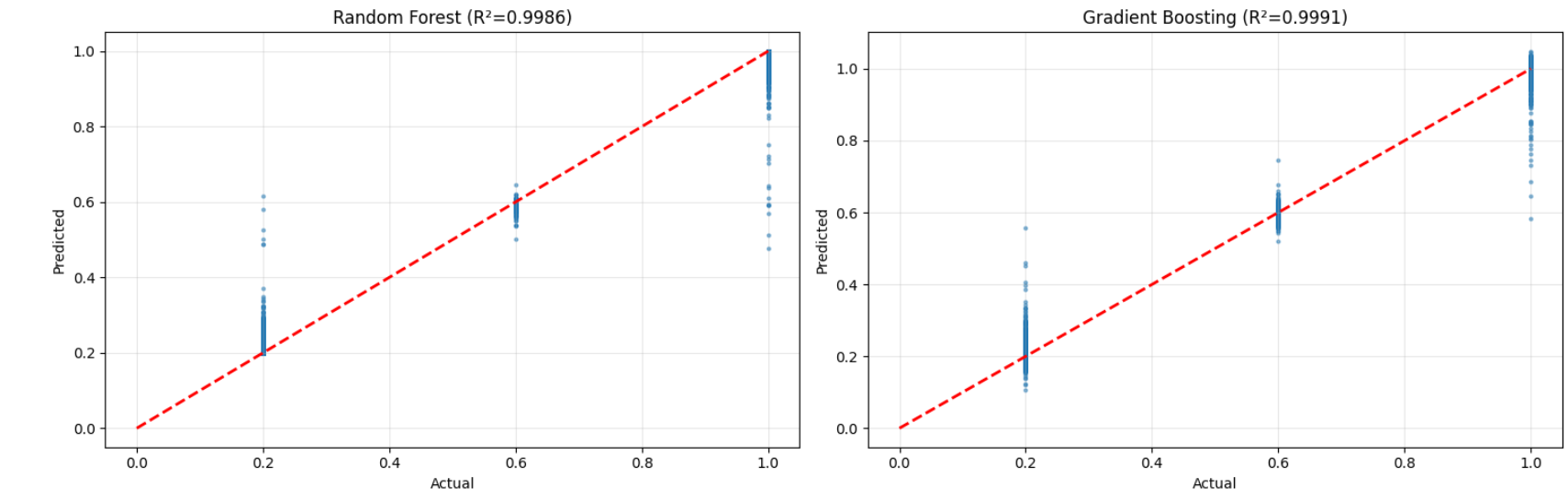
#### 4.3.3 So sánh 2 mô hình

Các thông số so sánh



Hình 4.14 So sánh 2 mô hình

Biểu đồ của 2 mô hình

**

Hình 4.15 Biểu đồ 2 mô hình

# KẾT LUẬN

### 1. Kết quả đạt được

* Đề tài "Hệ thống AI hỗ trợ gợi ý sản phẩm và dự đoán khả năng mua hàng cho cửa hàng thời trang trực tuyến" đã hoàn thành đầy đủ các mục tiêu đặt ra, đồng thời chứng minh được tính khả thi và hiệu quả của việc ứng dụng trí tuệ nhân tạo trong thương mại điện tử.
* Đầu tiên, hệ thống đã xây dựng thành công mô hình gợi ý sản phẩm dựa trên hành vi người dùng bằng hai thuật toán Random Forest Regressor và Gradient Boosting Regressor. Các mô hình hồi quy này dự đoán *like\_score* (mức độ quan tâm của khách hàng đối với sản phẩm) với độ chính xác cao. Kết quả đánh giá thông qua các thước đo chuẩn như R², MAE và RMSE cho thấy mô hình hoạt động ổn định và phù hợp cho bài toán gợi ý sản phẩm trong thực tế.
* Bên cạnh đó, đề tài cũng triển khai thành công hệ thống tìm kiếm và gợi ý sản phẩm dựa trên hình ảnh. Cụ thể, mô hình CNN (EfficientNet/ResNet) được sử dụng để trích xuất embedding ảnh, sau đó lưu trữ và truy vấn bằng FAISS nhằm tìm ra những sản phẩm có độ tương đồng cao nhất. Kết quả cho thấy hệ thống hoạt động nhanh, chính xác và đáp ứng tốt nhu cầu tìm kiếm trực quan theo hình ảnh.
* Tổng thể, đề tài đã xây dựng được một pipeline hoàn chỉnh gồm: tiền xử lý dữ liệu, huấn luyện mô hình, trích xuất embedding, xây dựng vector database FAISS và triển khai API bằng FastAPI. Đây là một hệ thống AI đa chức năng với khả năng ứng dụng cao trong thương mại điện tử.

### 2. Hạn chế và cải tiến

Mặc dù đạt được nhiều kết quả tích cực, đề tài vẫn tồn tại một số hạn chế:

* Quy mô dữ liệu còn nhỏ, chủ yếu mang tính thử nghiệm. Dữ liệu ít làm giảm khả năng tổng quát hóa của mô hình khi áp dụng vào môi trường thực tế, nơi số lượng sản phẩm và hành vi người dùng rất lớn.
* Quy trình tiền xử lý dữ liệu hành vi vẫn có thể tối ưu hơn. Một số giá trị nhiễu, ngoại lai hay sự mất cân bằng dữ liệu có thể ảnh hưởng đến hiệu suất dự đoán.
* Chưa tích hợp thành hệ thống thương mại điện tử hoàn chỉnh. Các mô hình mới được triển khai thông qua API và web demo đơn giản, chưa được gắn với cơ sở dữ liệu thực tế và luồng người dùng thật.
* Hệ thống tìm kiếm ảnh mới chỉ sử dụng CNN + FAISS, chưa áp dụng các mô hình thị giác hiện đại hơn như CLIP, ViT hay các phương pháp đa phương thức (multimodal).

### 3. Kiến nghị

Để nâng cao chất lượng và khả năng ứng dụng thực tế, một số kiến nghị bao gồm:

* Phát triển giao diện người dùng (UI): Cần triển khai một giao diện người dùng thân thiện và trực quan, giúp người dùng dễ dàng nhận được những sản phẩm gợi ý phù hợp nhất với nhu cầu của họ. Giao diện này cần phải được tối ưu hóa để giúp người dùng dễ dàng tương tác và nhận được thông tin chính xác trong thời gian ngắn nhất.
* Mở rộng dữ liệu: Thu thập nhiều dữ liệu hơn từ cửa hàng thời trang thực tế, bao gồm log hành vi, hình ảnh sản phẩm đa dạng, đánh giá của khách hàng,… giúp mô hình học được nhiều đặc trưng phức tạp hơn.
* Cải thiện mô hình gợi ý: Thử nghiệm các thuật toán hiện đại hơn như XGBoost, LightGBM hoặc các mô hình Deep Learning như Neural Collaborative Filtering (NCF), Recommender Systems dựa trên embedding người dùng – sản phẩm.
* Nâng cấp hệ thống tìm kiếm ảnh: Áp dụng các mô hình thị giác mạnh hơn như CLIP, Vision Transformer (ViT) để cải thiện độ chính xác và khả năng hiểu ngữ nghĩa của hình ảnh.
* Tối ưu hóa triển khai mô hình: Sử dụng Docker, GPU server hoặc dịch vụ cloud để nâng tốc độ suy luận và khả năng mở rộng. Ngoài ra, cần bổ sung pipeline cập nhật mô hình theo thời gian thực khi có dữ liệu mới.

# TÀI LIỆU THAM KHẢO

[1]. Nguyen, T. T., & Nguyen, M. T. (2020), *Deep Learning-Based E-commerce Product Recommendation Systems*, Springer.

[2]. Aggarwal (2016), *Recommender Systems: The Textbook*, Springer.

[3]. AICandy (2023). *Thuật toán Random Forest – Giải thích chi tiết và ứng dụng*. Available at: <https://aicandy.vn/thuat-toan-random-forest-giai-thich-chi-tiet-va-ung-dung/>

[4]. OpenAI (2024). *Embeddings Guide.* Available at: <https://platform.openai.com/docs/guides/embeddings>

[5]. W3Schools (2024). *Python & C# Tutorials.* Retrieved from <https://www.w3schools.com/python/>

[6]. RetailRocket (2017). *eCommerce Dataset.* Available at: <https://www.kaggle.com/datasets/retailrocket/ecommerce-dataset>

[7]. Agrawal, P. (2019). *Fashion Product Images (Small)*. Available at:, <https://www.kaggle.com/datasets/paramaggarwal/fashion-product-images-small>