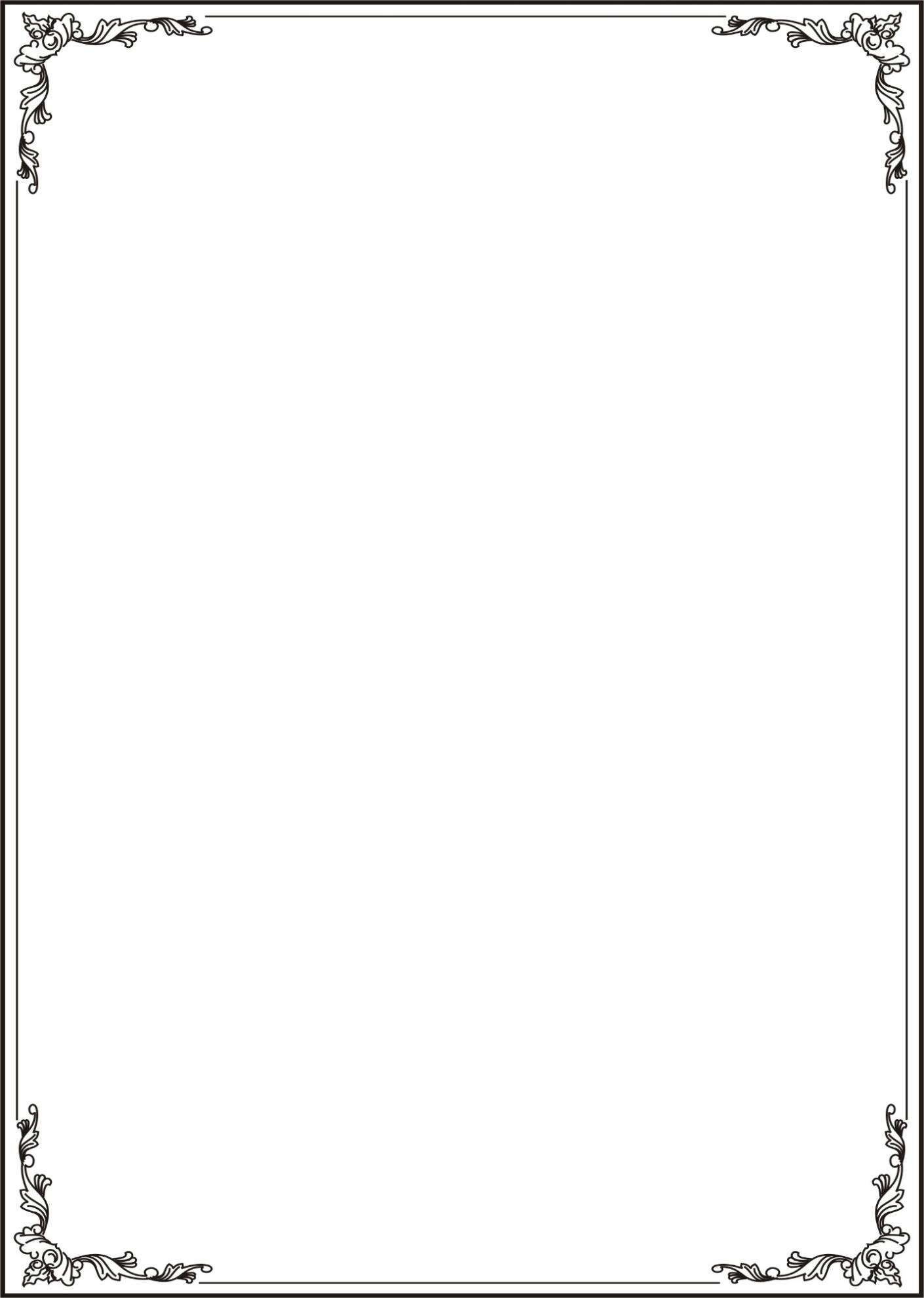
****

**KHOA CÔNG NGHỆ THÔNG TIN 1**

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**



**Phương pháp luận và nghiên cứu khoa học**

**ĐỀ TÀI: XÂY DỰNG ỨNG DỤNG ĐẶT ĐỒ ĂN VỚI CẢI TIẾN TỐI ƯU HÓA TRẢI NGHIỆM NGƯỜI DÙNG**

Sinh viên thực hiện : Đặng Minh Đức

Mã sinh viên : B21DCCN236

Nhóm môn học : 16

Giảng viên hướng dẫn : Nguyễn Thị Kim Chi

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | |  |  | |  |  | |  |  | |  |

Hà Nội

# LỜI CẢM ƠN

Trước hết, tôi xin bày tỏ lòng biết ơn sâu sắc đến cô Nguyễn Thị Kim Chi, giảng viên bộ môn Phương pháp luận và Nghiên cứu khoa học, trường Học viện Công nghệ Bưu chính Viễn thông. Nhờ sự hướng dẫn tận tình, những gợi ý quý báu và kiến thức chuyên môn sâu rộng của cô, tôi đã hoàn thành đề tài một cách tốt nhất có thể.

Trong suốt quá trình thực hiện đề tài, cô không chỉ cung cấp cho tôi các phương pháp nghiên cứu khoa học hiệu quả, mà còn khuyến khích tư duy sáng tạo và chủ động tìm tòi giải pháp. Với những hướng dẫn của cô đã giúp tôi hoàn thiện công việc một cách trọn vẹn.

Bên cạnh đó, tôi cũng xin gửi lời cảm ơn đến Học viện Công nghệ Bưu chính Viễn thông đã tạo điều kiện về tài liệu, môi trường học tập và cơ hội tiếp cận với các kiến thức nền tảng về khoa học và công nghệ, giúp tôi áp dụng hiệu quả vào nghiên cứu và phát triển sản phẩm.

Cuối cùng, tôi xin cảm ơn gia đình và bạn bè, những người đã luôn ủng hộ và động viên tôi trong suốt quá trình thực hiện đề tài này.

# MỤC LỤC

[LỜI CẢM ƠN ii](#_Toc182866860)

[DANH SÁCH CÁC HÌNH VẼ v](#_Toc182866861)

[DANH SÁCH CÁC TỪ VIẾT TẮT vi](#_Toc182866862)

[MỞ ĐẦU 1](#_Toc182866863)

[1.Giới thiệu 1](#_Toc182866864)

[2. Lý do chọn đề tài 1](#_Toc182866865)

[2.1. Xu hướng phát triển của ứng dụng đặt đồ ăn 1](#_Toc182866866)

[2.2. Nhu cầu cá nhân hóa trải nghiệm người dùng 1](#_Toc182866867)

[2.3. Ảnh hưởng của điều kiện thời tiết đến thói quen ăn uống 2](#_Toc182866868)

[2.4. Hỗ trợ nhà hàng tối ưu hóa doanh thu 2](#_Toc182866869)

[2.5. Ý nghĩa thực tiễn 2](#_Toc182866870)

[3. Mục đích nghiên cứu 2](#_Toc182866871)

[4. Đối tượng và phạm vi nghiên cứu 3](#_Toc182866872)

[4.1. Đối tượng nghiên cứu 3](#_Toc182866873)

[4.2. Phạm vi nghiên cứu 3](#_Toc182866874)

[Chương 1. Tổng quan về ứng dụng 4](#_Toc182866875)

[1.1 Tổng quan về ứng dụng đặt đồ ăn 4](#_Toc182866876)

[1.1.1. Khái niệm và vai trò của ứng dụng đặt đồ ăn 4](#_Toc182866877)

[1.1.2. Cấu trúc tổng quát của ứng dụng đặt đồ ăn 4](#_Toc182866878)

[1.2. Tổng quan về trải nghiệm người dùng và đề xuất sản phẩm 5](#_Toc182866879)

[1.3. Collaborative Filtering và ứng dụng trong đề xuất sản phẩm 6](#_Toc182866880)

[1.3.1. Phương pháp lọc cộng tác User-based và Item-based 6](#_Toc182866881)

[1.4. Tổng quan về tích hợp API thời tiết 10](#_Toc182866882)

[Chương 2. Xây dựng ứng dụng 12](#_Toc182866883)

[2.1. Thiết kế hệ thống 12](#_Toc182866884)

[2.1.1. Kiến trúc hệ thống 12](#_Toc182866885)

[2.1.2. Công nghệ sử dụng 12](#_Toc182866886)

[2.1.3. Quy trình hoạt động của hệ thống 13](#_Toc182866887)

[2.1.4. Bảo mật. 14](#_Toc182866888)

[2.2. Tích hợp Collaborative Filtering 14](#_Toc182866889)

[2.2.1. Xây dựng bộ dữ liệu 14](#_Toc182866890)

[2.2.2. Huấn luyện Collaborative Filtering 17](#_Toc182866891)

[2.2.3 Đánh giá mô hình 19](#_Toc182866892)

[2.3. Tích hợp API thời tiết 19](#_Toc182866893)

[2.3.1. API thời tiết và quy trình tích hợp 20](#_Toc182866894)

[1.3.2. Ứng dụng thời tiết trong đề xuất sản phẩm 21](#_Toc182866895)

[2.4. Tối ưu hóa trải nghiệm nhà hàng 23](#_Toc182866896)

[2.4.1 Thiết kế API cho thống kê doanh thu 23](#_Toc182866897)

[2.4.2. Thiết kế giao diện thống kê doanh thu 23](#_Toc182866898)

[Chương 3. Đánh giá và kết quả 25](#_Toc182866899)

[3.1. Các trường hợp đánh giá 25](#_Toc182866900)

[3.2. Đánh giá mô hình Collaborative Filtering 26](#_Toc182866901)

[3.3. Đánh giá hiệu quả tích hợp API thời tiết 26](#_Toc182866902)

[3.4 Đánh giá giao diện đề xuất sản phẩm theo sở thích và thời tiết 27](#_Toc182866903)

[3.5. Đánh giá giao diện thống kê doanh thu 27](#_Toc182866904)

[KẾT LUẬN 29](#_Toc182866905)

[1. Kết quả đạt được 29](#_Toc182866906)

[1.1. Về mặt lý thuyết 29](#_Toc182866907)

[1.2. Về mặt thực tiễn 29](#_Toc182866908)

[2. Hạn chế 29](#_Toc182866909)

[3. Hướng phát triển 30](#_Toc182866910)

[Tài liệu tham khảo 31](#_Toc182866911)

# DANH SÁCH CÁC HÌNH VẼ

[Hình 1.1 Mô tả về CF 7](#_Toc182868109)

[Hình 2.1 Dữ liệu đánh giá món ăn của người dùng 14](#_Toc182868110)

[Hình 2.2 Dữ liệu về người dùng 15](#_Toc182868111)

[Hình 2.3 Dữ liệu về món ăn 15](#_Toc182868112)

[Hình 2.4 Trích id của các cột từ các bảng 15](#_Toc182868113)

[Hình 2.5 Loại bỏ cột 16](#_Toc182868114)

[Hình 2.6 Bảng đầy đủ để xây dựng ma trận 16](#_Toc182868115)

[Hình 2.7 Ma trận 17](#_Toc182868116)

[Hình 2.8 Huấn luyện mô hình (Item-Based) 18](#_Toc182868117)

[Hình 2.9 Huấn luyện mô hình (User-Based) 18](#_Toc182868118)

[Hình 2.10 Đánh giá mô hình 19](#_Toc182868119)

[Hình 2.11 API Tích hợp OpenWeatherAPI 21](#_Toc182868120)

[Hình 2.12 Tính độ phù hợp 22](#_Toc182868121)

[Hình 2.13 Áp dụng 22](#_Toc182868122)

[Hình 3.1 Giao diện đề xuất 27](#_Toc182868123)

[Hình 3.2 Giao diện thống kê 28](#_Toc182868124)

# DANH SÁCH CÁC TỪ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| **Từ viết tắt** | **Tiếng Anh** | **Ý nghĩa** |
| API | |  | | --- | | Application Programming Interface |  |  | | --- | |  | | |  | | --- | | Giao diện lập trình ứng dụng |  |  | | --- | |  | |
| UI | |  | | --- | | User Interface |  |  | | --- | |  | | |  | | --- | | Giao diện người dùng |  |  | | --- | |  | |
| UX | User Experience | Trải nghiệm người dùng |
| CF | Collaborative Filtering | Lọc cộng tác |
| AI | Artifical Intelligence | Trí tuệ nhân tạo |
| ML | Machin Learning | Học máy |
| HTTP | Hypertext Transfer Protocol | Giao thức truyền tải siêu văn bản |
| JSON | Javascript Notation Object | Cấu trúc đối tượng trong Javascript |
| RMSE | Root Mean Squared Error | Căn trung bình bình phương sai số |

# MỞ ĐẦU

## Giới thiệu

Cùng với sự phát triển vượt bậc của công nghệ và sự thay đổi trong thói quen tiêu dùng, các ứng dụng đặt đồ ăn trực tuyến đã trở thành công cụ thiết yếu, mang lại sự tiện lợi và tiết kiệm thời gian cho hàng triệu người dùng trên toàn thế giới. Những ứng dụng này không chỉ đóng vai trò là cầu nối giữa khách hàng và nhà hàng mà còn tạo nên một hệ sinh thái phong phú giúp các nhà hàng tối ưu hóa hoạt động kinh doanh.

Tuy nhiên, thị trường ứng dụng đặt đồ ăn hiện nay đang đối mặt với nhiều thách thức. Khách hàng ngày càng có yêu cầu cao hơn về tính cá nhân hóa và sự phù hợp trong gợi ý món ăn. Đồng thời, các nhà hàng cần có những công cụ hỗ trợ hiệu quả hơn để quản lý doanh thu và ra quyết định chiến lược. Việc ứng dụng các công nghệ tiên tiến như *Collaborative Filtering* và tích hợp các API về thời tiết sẽ mang lại giải pháp tối ưu, cải thiện trải nghiệm người dùng và nâng cao hiệu quả kinh doanh.

Đề tài này tập trung xây dựng một ứng dụng đặt đồ ăn với các tính năng mới, bao gồm gợi ý sản phẩm dựa trên sở thích của từng người dùng, tích hợp điều kiện thời tiết trong quá trình gợi ý món ăn, và cung cấp giao diện thống kê doanh thu trực quan cho các nhà hàng đối tác.

## 2. Lý do chọn đề tài

### 2.1. Xu hướng phát triển của ứng dụng đặt đồ ăn

Trong những năm gần đây, ứng dụng đặt đồ ăn trực tuyến đã phát triển mạnh mẽ với các tên tuổi lớn như GrabFood, Batôiin, ShopeeFood. Những ứng dụng này không chỉ cung cấp dịch vụ giao đồ ăn mà còn mang lại sự tiện lợi vượt trội trong đời sống hàng ngày. Tuy nhiên, cạnh tranh trong lĩnh vực này ngày càng khốc liệt, và để tạo được sự khác biệt, các ứng dụng cần phải không ngừng cải tiến và đổi mới. Một trong những hướng đi quan trọng là tập trung vào trải nghiệm người dùng cá nhân hóa, giúp đáp ứng chính xác hơn nhu cầu của từng khách hàng.

### 2.2. Nhu cầu cá nhân hóa trải nghiệm người dùng

Các nghiên cứu cho thấy, người dùng có xu hướng quay lại các nền tảng cung cấp trải nghiệm phù hợp với sở thích và hành vi cá nhân. Tuy nhiên, phần lớn các ứng dụng đặt đồ ăn hiện nay vẫn chỉ dựa vào các danh mục tĩnh hoặc đề xuất không chính xác. Việc áp dụng Collaborative Filtering có thể giúp giải quyết vấn đề này, thông qua phân tích dữ liệu và đưa ra các gợi ý dựa trên thói quen mua hàng của người dùng tương tự.

### 2.3. Ảnh hưởng của điều kiện thời tiết đến thói quen ăn uống

Thời tiết là một yếu tố ảnh hưởng lớn đến quyết định đặt món ăn của người dùng. Ví dụ, trong ngày mưa lạnh, các món ăn nóng như lẩu, mì có xu hướng được ưa chuộng hơn. Ngược lại, vào những ngày hè nóng bức, các món giải nhiệt như ktôi, trà sữa thường được đặt nhiều hơn. Tích hợp API thời tiết để điều chỉnh gợi ý sản phẩm theo thời gian thực không chỉ làm tăng sự hài lòng của người dùng mà còn tạo nên sự khác biệt so với các đối thủ.

### 2.4. Hỗ trợ nhà hàng tối ưu hóa doanh thu

Ngoài việc tối ưu hóa trải nghiệm cho khách hàng, việc cung cấp các công cụ hỗ trợ nhà hàng quản lý hiệu quả hơn cũng là một yếu tố quan trọng. Thống kê doanh thu bằng biểu đồ không chỉ giúp nhà hàng theo dõi tình hình kinh doanh mà còn đưa ra các dự báo hữu ích, hỗ trợ việc ra quyết định.

### 2.5. Ý nghĩa thực tiễn

Đề tài này không chỉ mang lại lợi ích cho người dùng cá nhân mà còn hỗ trợ nhà hàng cải thiện hoạt động kinh doanh, từ đó góp phần vào sự phát triển chung của ngành dịch vụ giao nhận thực phẩm trực tuyến. Với sự kết hợp giữa công nghệ tiên tiến và phân tích dữ liệu, đề tài có tiềm năng ứng dụng cao trong thực tế.

## 3. Mục đích nghiên cứu

Đề tài tập trung vào việc xây dựng một ứng dụng đặt đồ ăn trực tuyến với những cải tiến nhằm tối ưu hóa trải nghiệm người dùng và hỗ trợ nhà hàng kinh doanh hiệu quả. Mục đích cụ thể bao gồm:

* Tối ưu hóa trải nghiệm người dùng cá nhân hóa: Ứng dụng sử dụng Collaborative Filtering để phân tích hành vi người dùng, từ đó gợi ý các món ăn hoặc nhà hàng phù hợp với sở thích và thói quen của từng cá nhân.
* Khai thác dữ liệu thời tiết để tăng tính thông minh trong đề xuất: Tích hợp API thời tiết để đưa ra các gợi ý món ăn phù hợp với điều kiện thời tiết hiện tại, nâng cao sự hài lòng và tính tiện ích.
* Cải thiện công cụ hỗ trợ nhà hàng: Phát triển tính năng thống kê doanh thu trực quan qua biểu đồ, giúp nhà hàng theo dõi hiệu quả kinh doanh, nhận biết xu hướng đặt món, và đưa ra quyết định chiến lược.

## 4. Đối tượng và phạm vi nghiên cứu

### 4.1. Đối tượng nghiên cứu

Bao gồm:

* Người dùng: Là những cá nhân sử dụng ứng dụng đặt đồ ăn trực tuyến, bao gồm các nhóm người dùng phổ biến như sinh viên, nhân viên văn phòng, hoặc các hộ gia đình bận rộn.
* Dữ liệu: Dữ liệu về hành vi người dùng (như lịch sử đặt món, lượt đánh giá các món ăn), thông tin món ăn, và dữ liệu thời tiết theo thời gian thực.
* Nhà hàng: Các nhà hàng sử dụng ứng dụng để cung cấp dịch vụ và mong muốn tăng hiệu quả kinh doanh thông qua các công cụ hỗ trợ.

### 4.2. Phạm vi nghiên cứu

Phạm vi nội dung:

* Xây dựng một ứng dụng mẫu hỗ trợ đặt đồ ăn trực tuyến.
* Tích hợp các thuật toán gợi ý sản phẩm dựa trên Collaborative Filtering.
* Kết nối API thời tiết để gợi ý món ăn theo điều kiện thời tiết.
* Phát triển tính năng thống kê doanh thu qua biểu đồ cho nhà hàng.

Phạm vi kỹ thuật:

* Nghiên cứu và áp dụng các thuật toán Collaborative Filtering.
* Sử dụng công nghệ API thời tiết (như OpenWeather hoặc AccuWeather) để thu thập dữ liệu.
* Phát triển giao diện trực quan và trải nghiệm người dùng dựa trên Flutter.

Phạm vi ứng dụng thực tiễn:

* Ứng dụng thử nghiệm sẽ tập trung vào nhóm người dùng tại một thành phố hoặc khu vực cụ thể để kiểm tra tính hiệu quả.
* Dữ liệu được thu thập giới hạn trong khoảng thời gian từ 3-6 tháng.

# Chương 1. Tổng quan về ứng dụng

## Tổng quan về ứng dụng đặt đồ ăn

### 1.1.1. Khái niệm và vai trò của ứng dụng đặt đồ ăn

Ứng dụng đặt đồ ăn là một nền tảng công nghệ cho phép người dùng lựa chọn, đặt món ăn và nhận giao hàng từ các nhà hàng, quán ăn thông qua thiết bị di động hoặc website. Đây là một phần quan trọng trong ngành công nghệ dịch vụ ăn uống, kết hợp giữa thương mại điện tử và logistics.

Ứng dụng đặt đồ ăn là một giải pháp công nghệ số hóa việc đặt món, giúp kết nối người dùng với nhà hàng một cách nhanh chóng, tiện lợi. Ứng dụng thường cung cấp các tính năng như xtôi món, đặt món online , theo dõi đơn hàng, và đánh giá nhà hàng.

Vai trò:

1. Đối với người dùng:

* Tiết kiệm thời gian và công sức khi không cần đến tận nơi để mua món ăn.
* Cung cấp sự lựa chọn đa dạng về nhà hàng, món ăn phù hợp với sở thích cá nhân.
* Đtôi lại trải nghiệm tiện ích qua các tính năng như gợi ý món ăn, ưu đãi giảm giá, theo dõi thời gian giao hàng.

1. Đối với nhà hàng:

* Mở rộng thị trường tiếp cận đến lượng khách hàng lớn hơn, bao gồm cả những khách hàng không thể đến quán trực tiếp.
* Giảm tải chi phí vận hành bằng cách tích hợp với hệ thống giao hàng của bên thứ ba.
* Thu thập dữ liệu khách hàng để cải thiện dịch vụ và xây dựng chiến lược kinh doanh.

1. Đối với xã hội:

* Tăng cường sự linh hoạt trong việc cung cấp dịch vụ ăn uống, đặc biệt hữu ích trong các tình huống như đại dịch hoặc phong tỏa.
* Thúc đẩy sự phát triển của các công nghệ liên quan như thanh toán trực tuyến, AI trong đề xuất món ăn, và quản lý chuỗi cung ứng.

### 1.1.2. Cấu trúc tổng quát của ứng dụng đặt đồ ăn

Ứng dụng đặt đồ ăn thường bao gồm 3 thành phần chính:

1. Thành phần phía người dùng :

* Giao diện người dùng (UI): Thiết kế thân thiện, dễ sử dụng, hỗ trợ các chức năng như tìm kiếm nhà hàng, chọn món, và đặt hàng.
* Quản lý tài khoản: Bao gồm đăng nhập, đăng ký, quản lý thông tin cá nhân, và địa chỉ giao hàng.
* Đặt hàng và thanh toán: Tích hợp các phương thức thanh toán như thẻ tín dụng, ví điện tử, hoặc thanh toán khi nhận hàng.
* Theo dõi đơn hàng: Hiển thị trạng thái đơn hàng, thời gian dự kiến giao, và thông tin người giao hàng.

1. Thành phần phía nhà hàng:

* Quản lý menu: Hỗ trợ nhà hàng thêm, xóa, cập nhật món ăn và giá cả.
* Quản lý đánh giá: Cung cấp thông tin phản hồi từ khách hàng để cải thiện dịch vụ.
* Thống kê về doanh thu theo ngày, tháng, năm.

1. Thành phần phía người giao hàng:

* Quản lý đơn hàng: Tích hợp Google Map đề xuất các đơn hàng thuận lợi đối với người giao hàng. Hỗ trợ quản lý thông tin đơn hàng và giao tiếp với người đặt hàng.
* Quản lý đánh giá: Cung cấp thông tin phản hồi từ khách hàng để cải thiện dịch vụ.

1. Thành phần phía máy chủ:

* Quản lý dữ liệu: Xử lý thông tin từ người dùng, nhà hàng và hệ thống giao hàng.
* Quản lý dữ liệu với trang Admin: Quản lý dữ liệu với giao diện dễ học và sử dụng cho những người quản trị viên hệ thống.
* Hệ thống giao hàng: Kết nối với các dịch vụ vận chuyển để đảm bảo thời gian giao hàng nhanh chóng.
* Hệ thống đề xuất sản phẩm: Tích hợp trí Collaborative Filtering để gợi ý món ăn dựa trên thói quen của người dùng.
* Bảo mật: Đảm bảo an toàn dữ liệu người dùng và giao dịch tài chính thông qua các giao thức mã hóa.

## 1.2. Tổng quan về trải nghiệm người dùng và đề xuất sản phẩm

Trải nghiệm người dùng (UX) trong ứng dụng đặt đồ ăn:

* Khái niệm: Trải nghiệm người dùng tập trung vào cảm giác, sự hài lòng và sự tiện ích mà người dùng có được khi sử dụng ứng dụng.
* Yếu tố quan trọng:

1. Hiệu năng: Ứng dụng cần hoạt động nhanh, mượt mà, không bị gián đoạn trong các thao tác đặt hàng.
2. Sự trực quan: Thiết kế giao diện phải dễ hiểu, phù hợp với mọi lứa tuổi.
3. Cá nhân hóa: Hiển thị nội dung phù hợp với sở thích và lịch sử mua sắm của người dùng.
4. Tương tác đa kênh: Hỗ trợ đồng bộ giữa ứng dụng di động, website, và các nền tảng khác.

Đề xuất sản phẩm:

* Khái niệm: Đề xuất sản phẩm là việc gợi ý món ăn, nhà hàng hoặc các ưu đãi phù hợp với sở thích của từng cá nhân dựa trên dữ liệu thu thập được.
* Các kỹ thuật đề xuất sản phẩm:

1. Rule-based recommendation: Gợi ý dựa trên các quy tắc cố định, ví dụ như món ăn phổ biến nhất hoặc ưu đãi hiện hành.
2. Collaborative Filtering:
   * User-based: Dựa trên hành vi của những người dùng có sở thích tương tự.
   * Item-based: Dựa trên các món ăn có tính chất tương đồng như được đánh giá qua các lần đặt hàng.
3. Content-based recommendation: Gợi ý dựa trên đặc điểm món ăn mà người dùng đã chọn hoặc đánh giá tốt.
4. Hybrid recommendation: Kết hợp nhiều phương pháp để tăng độ chính xác và sự linh hoạt.

Vai trò của đề xuất sản phẩm trong trải nghiệm người dùng:

* Cải thiện tính cá nhân hóa, giúp người dùng dễ dàng tìm thấy món ăn yêu thích mà không cần mất nhiều thời gian tìm kiếm.
* Tăng doanh thu cho nhà hàng thông qua việc gợi ý các món ăn mới hoặc các combo khuyến mãi hấp dẫn.
* Giữ chân người dùng bằng cách tạo trải nghiệm thú vị và tiện ích hơn.

## 1.3. Collaborative Filtering và ứng dụng trong đề xuất sản phẩm

### 1.3.1. Phương pháp lọc cộng tác User-based và Item-based

Collaborative Filtering (CF) là một trong những kỹ thuật phổ biến nhất trong hệ thống gợi ý, sử dụng dữ liệu lịch sử của người dùng để đưa ra các đề xuất cá nhân hóa. CF hoạt động dựa trên ý tưởng rằng nếu người dùng có hành vi hoặc sở thích giống nhau, họ sẽ có khả năng quan tâm đến các sản phẩm tương tự.

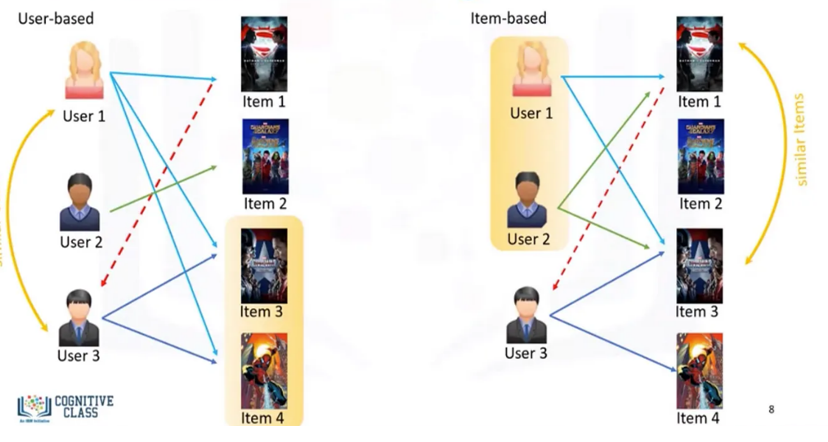
Phương pháp lọc cộng tác được chia thành hai loại chính:

1. User-based Collaborative Filtering:

* Dựa trên hành vi của người dùng. Nếu hai người dùng có lịch sử tương tác giống nhau (ví dụ: cùng đánh giá cao một món ăn), hệ thống sẽ đề xuất các sản phẩm mà một trong hai người chưa khám phá.
* Ví dụ: Người dùng A và B đều thích món "Pizza Pepperoni". Nếu A đã thử món "Burger BBQ" và đánh giá cao, thì hệ thống sẽ gợi ý món này cho B.
* Công thức phổ biến:  
  Hệ thống sử dụng các thuật toán như Cosine Similarity hoặc Pearson Correlation để đo độ tương đồng giữa các người dùng dựa trên ma trận đánh giá.

1. Item-based Collaborative Filtering:

* Dựa trên sự tương đồng giữa các sản phẩm. Nếu hai sản phẩm được đánh giá tích cực bởi cùng nhóm người dùng, chúng sẽ được coi là tương tự nhau.
* Ví dụ:
* Món "Mì Ý sốt bò bằm" và "Mì Ý sốt hải sản" được nhiều người dùng đánh giá cao cùng lúc. Hệ thống sẽ đề xuất món còn lại khi người dùng chọn một trong hai món này.
* Công thức phổ biến:  
  Hệ thống dựa vào ma trận tương tác giữa sản phẩm và người dùng, sau đó tính toán độ tương đồng giữa các sản phẩm bằng Jaccard Similarity hoặc Adjusted Cosine Similarity.



#### Hình 1.1 Mô tả về CF

***1.3.2.*** ***Minh họa quy trình Collaborative Filtering***

Trong Collaborative Filtering dựa trên người dùng, ta có một người dùng mục tiêu (active user), là người mà ta muốn tạo gợi ý. Hệ thống Collaborative Filtering sẽ tìm kiếm những người dùng có mức độ tương đồng cao với người dùng mục tiêu, tức là những người có mô hình đánh giá sản phẩm giống nhau. Mức độ tương đồng này được tính toán dựa trên lịch sử, sở thích, và lựa chọn mà người dùng thực hiện khi mua sắm, xtôi, hoặc sử dụng một sản phẩm nào đó.

Ví dụ: Nếu một người dùng A và người dùng B đều thích các món ăn X, Y, thì nếu A thích món ăn Z, hệ thống có thể gợi ý món ăn Z cho B.

1. Đối với User-based:

Cách hoạt động

* Bước 1: Giả sử ta có một ma trận người dùng - sản phẩm (user-Item matrix), biểu diễn điểm đánh giá của 4 người dùng cho 5 món ăn. Người dùng mục tiêu chỉ đánh giá 3 trong số 5 món ăn này.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| User/Món ăn | Món ăn A | Món ăn B | Món ăn C | Món ăn D | Món ăn E |
| User 1 | 5 | 3 | 0 | 0 | 1 |
| User 2 | 4 | 0 | 0 | 2 | 5 |
| User 3 | 0 | 2 | 4 | 5 | 0 |
| User 4 | 1 | 1 | 0 | 0 | 4 |

* Bước 2: Tìm mức độ tương đồng giữa người dùng mục tiêu và những người khác bằng cách sử dụng các kỹ thuật như:
  + Khoảng cách Euclidean (Euclidean Distance).
  + Hệ số tương quan Pearson (Pearson Correlation).
  + Độ tương tự Cosine (Cosine Similarity)..
  + Ví dụ tính toán dựa trên bảng trên ta được:

|  |  |  |  |
| --- | --- | --- | --- |
| User | | Độ Tương Đồng với User 1 | | --- |  |  | | --- | |  | |
| User 2 | 0.8 |
| User 3 | 0.7 |
| User 4 | 0.6 |

* Bước 3: Dự đoán điểm đánh giá cho các món ăn chưa thử. Sau khi có độ tương đồng, hệ thống sẽ tính toán điểm dự đoán cho Món Ăn C và Món Ăn D mà User 1 chưa đánh giá.
* Dự đoán cho Món Ăn C: Cộng điểm đánh giá của các người dùng tương tự, có trọng số theo độ tương đồng:
* Dự đoán cho Món Ăn D: Tương tự, ta tính toán cho Món Ăn D:
* Bước 4: Gợi ý món ăn
* Dựa trên các dự đoán này, hệ thống sẽ gợi ý cho User 1 các món ăn mà người này chưa thử, nhưng có thể sẽ thích, ví dụ như:
  + Món Ăn D (dự đoán điểm: 3.5)
  + Món Ăn C (dự đoán điểm: 2.8)
* Như vậy, User 1 có thể được gợi ý thử Món Ăn D vì dự đoán điểm đánh giá khá cao.

1. Đối với Item-based:

Gần như tương đồng với User-based khác ở bước 2 với ma trận tương đồng sẽ là giữa món ăn với món ăn:

|  |  |  |  |
| --- | --- | --- | --- |
| Món ăn | | Độ Tương Đồng với Món ăn 1 | | --- |  |  | | --- | |  | |
| Món ăn 1 | … |
| Món ăn 2 | … |
| Món ăn 3 | … |

***1.3.3. Ưu điểm và hạn chế của Collaborative Filtering***

Ưu điểm:

1. Không yêu cầu kiến thức về sản phẩm:

* Hệ thống không cần thông tin chi tiết về món ăn (ví dụ: thành phần hoặc cách chế biến). Thay vào đó, nó chỉ dựa vào dữ liệu tương tác như đánh giá, lịch sử đặt hàng.
* Điều này giúp CF dễ dàng mở rộng ra các lĩnh vực khác nhau, từ sách, phim , món ăn, ảnh đến thời trang và thực phẩm.

1. Khả năng cá nhân hóa cao:

* Đề xuất dựa trên hành vi thực tế của người dùng, giúp tăng khả năng cung cấp các gợi ý phù hợp.

1. Khả năng phát hiện sở thích tiềm ẩn:

* CF có thể gợi ý các món ăn mà người dùng chưa từng thử nhưng phù hợp với sở thích của họ.

Hạn chế:

1. Vấn đề Cold Start:

* Đối với người dùng mới (chưa có lịch sử tương tác) hoặc món ăn mới (chưa có đánh giá), hệ thống gặp khó khăn trong việc đưa ra gợi ý.
* Giải pháp: Kết hợp với Content-based Filtering hoặc thu thập thêm thông tin từ người dùng thông qua bảng khảo sát sở thích.
* Dữ liệu thưa thớt (Data Sparsity):
* Trong trường hợp có quá nhiều sản phẩm và người dùng, ma trận tương tác trở nên rất lớn nhưng lại chứa nhiều giá trị rỗng (người dùng không đánh giá hầu hết sản phẩm). Điều này làm giảm hiệu quả của thuật toán CF.

1. Vấn đề Popularity Bias:

* Hệ thống có xu hướng ưu tiên các món ăn phổ biến, khiến các món mới hoặc ít được đặt trở nên khó tiếp cận hơn.

1. Hiệu suất tính toán:

* Khi số lượng người dùng và món ăn tăng lên, thuật toán CF đòi hỏi tài nguyên tính toán lớn để tìm kiếm và so sánh độ tương đồng.

## 1.4. Tổng quan về tích hợp API thời tiết

API thời tiết cung cấp dữ liệu thời tiết theo thời gian thực từ các nguồn đáng tin cậy (như OpenWeatherMap, WeatherStack). API này thường bao gồm các thông tin như:Nhiệt độ, Độ ẩm**,**  thời tiết (mưa, nắng, gió), điều kiện khí hậu theo từng khu vực cụ thể.

Ứng dụng API thời tiết trong hệ thống đặt đồ ăn

Cá nhân hóa đề xuất món ăn: Dữ liệu thời tiết có thể được sử dụng để gợi ý món ăn phù hợp với điều kiện hiện tại.Ví dụ: Trời lạnh sẽ gợi ý các món ăn nóng như lẩu, súp và khi trời nóng sẽ gợi ý đồ uống giải khát hoặc món ăn nhẹ.

Tăng cường trải nghiệm người dùng: Thời tiết thường ảnh hưởng đến tâm trạng và lựa chọn của khách hàng. Bằng cách tích hợp thời tiết vào hệ thống, ứng dụng có thể cung cấp trải nghiệm "thấu hiểu" người dùng hơn.

Quy trình tích hợp API thời tiết

1. Lựa chọn nhà cung cấp API: Các nhà cung cấp phổ biến: OpenWeatherMap, WeatherStack, AccuWeather với tiêu chí lựa chọn bao gồm: Độ chính xác, khả năng mở rộng, chi phí, và tài liệu hỗ trợ.
2. Kết nối API: Tích hợp API thông qua các giao thức HTTP/HTTPS. Gửi yêu cầu (request) chứa thông tin địa điểm của người dùng (ví dụ: tọa độ GPS hoặc mã zip).
3. Xử lý dữ liệu: Hệ thống nhận phản hồi (response) từ API dưới dạng JSON/XML, sau đó trích xuất thông tin cần thiết (ví dụ: nhiệt độ, thời tiết hiện tại).
4. Áp dụng: Dựa vào dữ liệu thời tiết của từng khu vực đối với từng người dùng từ đó tích hợp với món ăn để đưa ra đề xuất đối với người dùng.

Ưu và nhược điểm của tích hợp API thời tiết

* Ưu điểm: Cập nhật theo thời gian thực đảm bảo dữ liệu luôn chính xác với tính cá nhân hóa từ đó sẽ gợi ý sản phẩm theo điều kiện thời tiết của từng khu vực.
* Nhược điểm:Phụ thuộc vào dịch vụ bên thứ ba khi hệ thống có thể gặp gián đoạn nếu API gặp lỗi hoặc bị giới hạn tài nguyên với yêu cầu phí cao cho các tính năng nâng cao hoặc dung lượng lớn với một số nhà cung cấp API.

# Chương 2. Xây dựng ứng dụng

## 2.1. Thiết kế hệ thống

Việc thiết kế một hệ thống cho ứng dụng đặt đồ ăn yêu cầu sự kết hợp giữa các yếu tố kỹ thuật để đảm bảo hiệu suất, khả năng mở rộng, và tối ưu hóa trải nghiệm người dùng. Trong chương này, tôi sẽ trình bày chi tiết về thiết kế hệ thống của ứng dụng đặt đồ ăn, sử dụng các công nghệ tiên tiến để đảm bảo hiệu quả và tính linh hoạt của hệ thống. Các công nghệ chính bao gồm Dart, Flutter, Python, Django Rest Framework, cùng với các thư viện và plugin như Flutter Google Maps, GetX, và các công cụ hỗ trợ khác.

### 2.1.1. Kiến trúc hệ thống

Hệ thống ứng dụng đặt đồ ăn được thiết kế theo mô hình Client-Server, trong đó:

* Client (Ứng dụng di động): Được phát triển bằng Flutter, sử dụng Dart làm ngôn ngữ lập trình chính. Ứng dụng hoạt động Android, giúp người dùng có thể dễ dàng truy cập vào dịch vụ đặt đồ ăn từ điện thoại di động của mình.
* Server (Backend): Dựa trên Python và Django Rest Framework để phát triển các API phục vụ cho việc xử lý dữ liệu, quản lý người dùng, xử lý đơn hàng và tích hợp với các mô hình học máy (machine learning). Python được sử dụng để huấn luyện và triển khai mô hình Collaborative Filtering, giúp đưa ra các gợi ý về món ăn cho người dùng dựa trên thói quen và sở thích.

Cả client và server sẽ giao tiếp thông qua RESTful API, giúp kết nối các phần tử trong hệ thống với nhau. Dữ liệu được lưu trữ trong cơ sở dữ liệu PostgreSQL, một lựa chọn phổ biến cho các ứng dụng web vì tính ổn định và khả năng mở rộng.

### 2.1.2. Công nghệ sử dụng

Flutter & Dart: Được chọn làm công nghệ chính cho phần client vì khả năng phát triển ứng dụng cross-platform nhanh chóng và dễ dàng. Flutter cung cấp một framework mạnh mẽ để xây dựng giao diện người dùng đẹp mắt, với khả năng tương thích cao trên các thiết bị di động. Dart là ngôn ngữ lập trình chính, mang lại hiệu suất cao và khả năng lập trình bất đồng bộ, điều này rất hữu ích khi xử lý các tác vụ như tải dữ liệu từ API hay hiển thị bản đồ.

Google Maps & Direction API: Google Maps được tích hợp để cung cấp bản đồ cho người dùng và các nhà hàng, cho phép người dùng tìm kiếm và xác định vị trí các nhà hàng xung quanh. Google Directions API giúp tính toán và hiển thị lộ trình di chuyển của người giao hàng từ nhà hàng đến người nhận.

GetX: Là một thư viện quản lý trạng thái cho Flutter, GetX được sử dụng để cải thiện hiệu suất và đơn giản hóa quá trình quản lý trạng thái của ứng dụng. Thư viện này giúp việc xây dựng các tính năng như điều hướng, xử lý sự kiện và quản lý trạng thái người dùng trở nên dễ dàng và hiệu quả.

Python & Django Rest Framework: Backend của hệ thống được phát triển với Python, một ngôn ngữ lập trình phổ biến trong lĩnh vực phát triển web và học máy. Django Rest Framework được sử dụng để xây dựng API, phục vụ việc quản lý dữ liệu người dùng, các đơn hàng, cũng như tương tác với các mô hình học máy. Python cũng được sử dụng để huấn luyện và triển khai mô hình Collaborative Filtering, một phương pháp học máy phổ biến trong việc đưa ra các gợi ý sản phẩm dựa trên hành vi của người dùng.

Collaborative Filtering: Đây là kỹ thuật học máy được sử dụng để cá nhân hóa các đề xuất món ăn cho người dùng dựa trên các tương tác trước đó của họ hoặc dựa trên sở thích của những người dùng tương tự. Các mô hình collaborative filtering có thể là user-based hoặc Item-based, tùy thuộc vào cách thức dữ liệu được xử lý và khai thác.

### 2.1.3. Quy trình hoạt động của hệ thống

Hệ thống của ứng dụng hoạt động qua ba bước chính:

* Tương táccủa người dùng: Người dùng tải ứng dụng, đăng nhập hoặc đăng ký tài khoản và có thể tìm kiếm các món ăn hoặc nhà hàng trong khu vực. Các thông tin như món ăn yêu thích, lịch sử đặt hàng và vị trí của người dùng sẽ được hệ thống ghi nhận và sử dụng để cải thiện trải nghiệm người dùng.
* Hệ thống đề xuất: Dựa trên các dữ liệu thu thập được từ người dùng (bao gồm các món ăn đã đặt trước, đánh giá món ăn) hệ thống sẽ sử dụng Collaborative Filtering để đề xuất các món ăn phù hợp. Mô hình này sẽ được huấn luyện từ dữ liệu lịch sử của người dùng và cập nhật theo thời gian để ngày càng chính xác hơn.
* Xử lý đơn hàng và giao hàng: Khi người dùng lựa chọn món ăn và đặt hàng, hệ thống sẽ gửi yêu cầu đến backend để xác nhận đơn hàng, tính toán phí vận chuyển dựa trên khoảng cách, và cung cấp lộ trình giao hàng thông qua Google Directions API, sau đó gửi yêu cầu đến người giao hàng. Sau khi đơn hàng được giao, thông tin về doanh thu sẽ được gửi về backend để lưu trữ và phân tích.

### 2.1.4. Bảo mật.

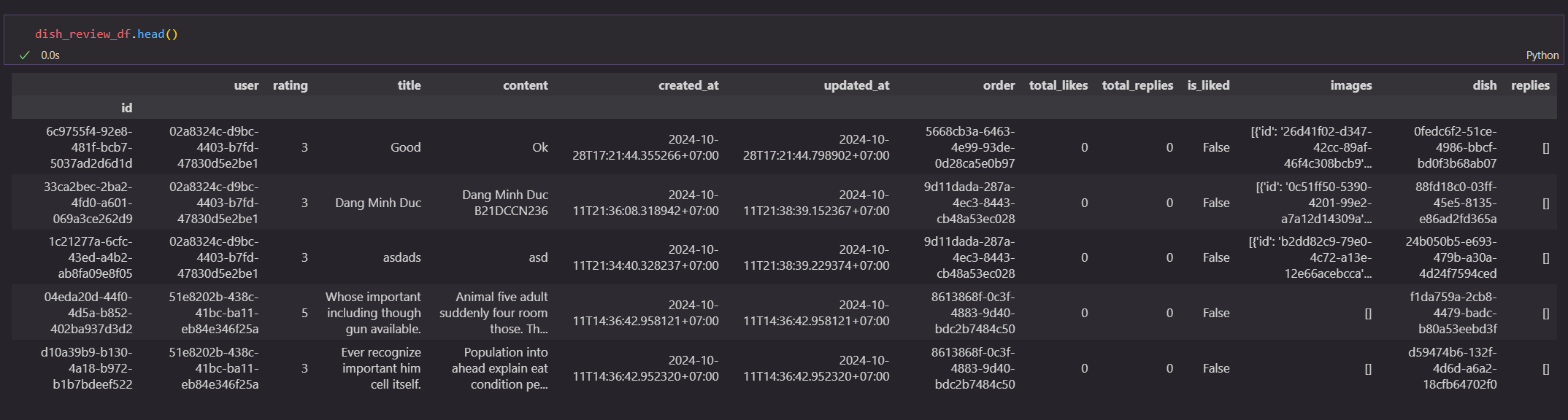
Bảo mật: Các API được bảo vệ với JWT Authentication để đảm bảo rằng dữ liệu của người dùng và giao dịch được bảo mật. Các biện pháp bảo vệ như mã hóa dữ liệu và sử dụng HTTPS cũng sẽ được triển khai để bảo vệ thông tin cá nhân. Với các phương thức bảo mật sẽ bảo mật được thông tin người dùng và thanh toán, tránh việc đánh cắp thông tin người dùng và can thiệp vào hoạt động thanh toán.

## 2.2. Tích hợp Collaborative Filtering

### 2.2.1. Xây dựng bộ dữ liệu

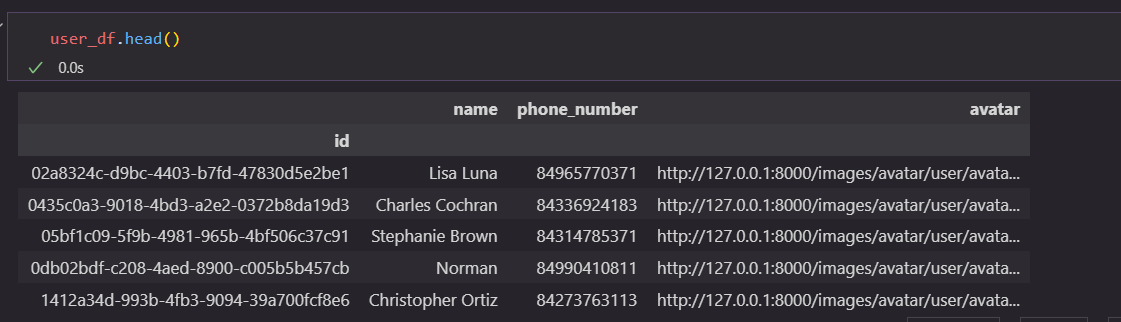
1. Bộ dữ liệu sẽ được chia thành các phần chính sau:

* Đánh giá của người dùng: Đây là các đánh giá món ăn từ người dùng, thường dưới dạng sao (ví dụ, 1 đến 5 sao) hoặc nhận xét văn bản. Các đánh giá này rất quan trọng vì chúng giúp mô hình hiểu được sự ưa thích của người dùng đối với các món ăn.



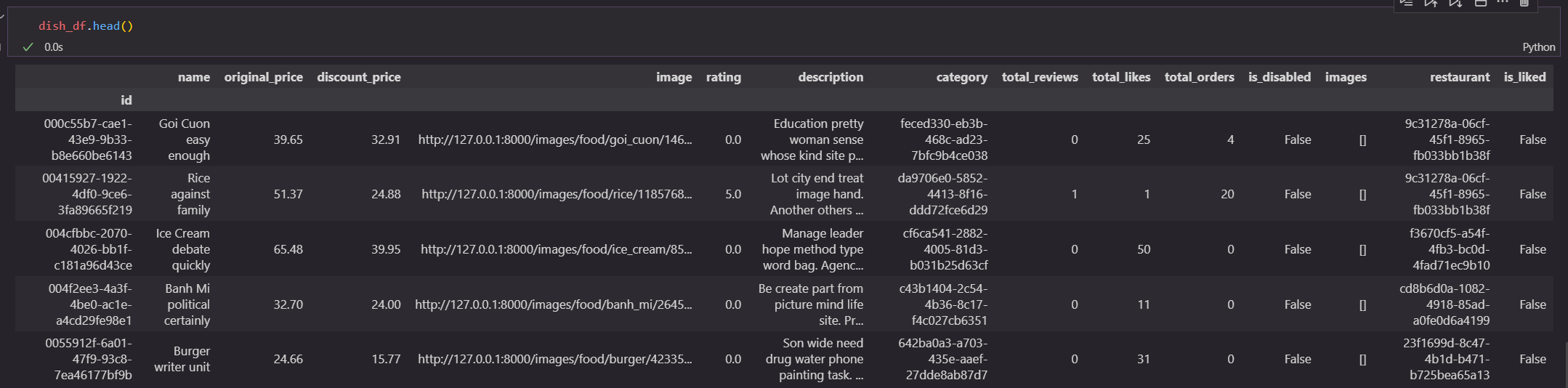
#### Hình 2.1 Dữ liệu đánh giá món ăn của người dùng

* Thông tin người dùng: Dữ liệu này có thể bao gồm độ tuổi, giới tính, và các yếu tố nhân khẩu học khác, giúp cải thiện độ chính xác của các đề xuất.



#### Hình 2.2 Dữ liệu về người dùng

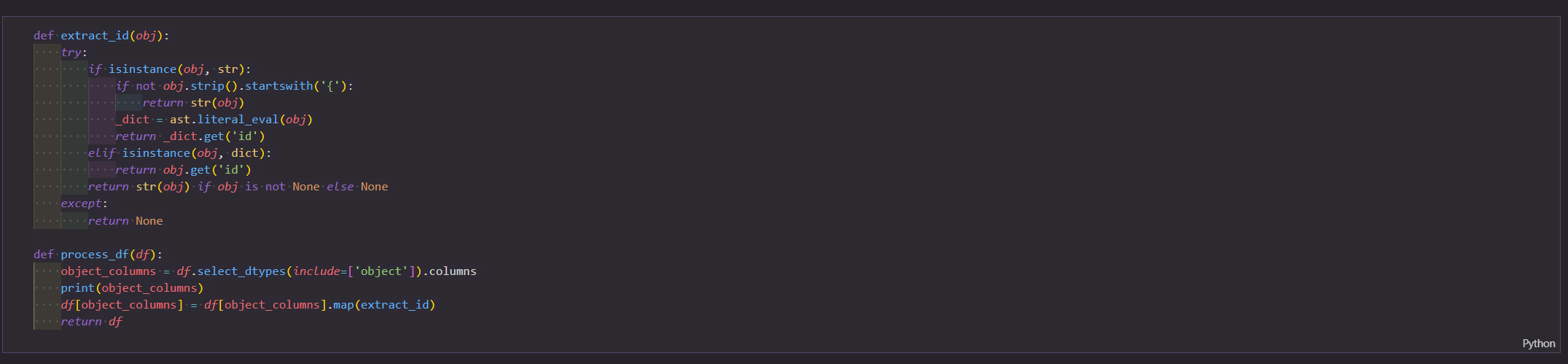
* **Thông tin món ăn**: Các đặc điểm của món ăn như loại món (món chính, món tráng miệng, đồ uống), mức giá, v.v.

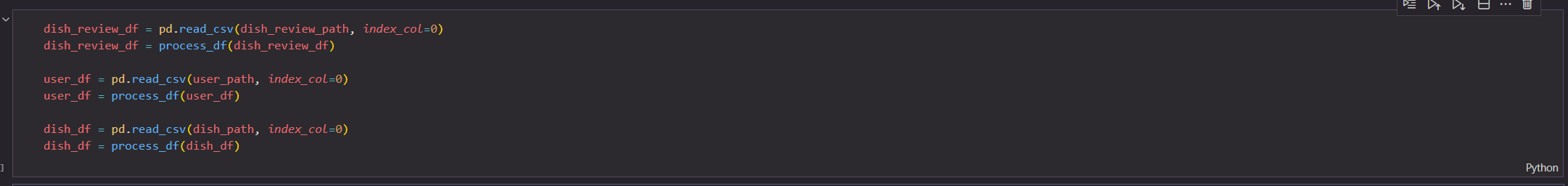


#### Hình 2.3 Dữ liệu về món ăn

Xử lý và làm sạch dữ liệu: Trước khi bắt đầu huấn luyện mô hình, cần thực hiện quá trình tiền xử lý và làm sạch dữ liệu. Quá trình này bao gồm:

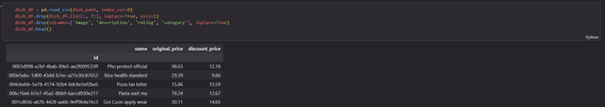
* Ở mỗi bảng trích giá trị từ các cột có dạng là từ điển.





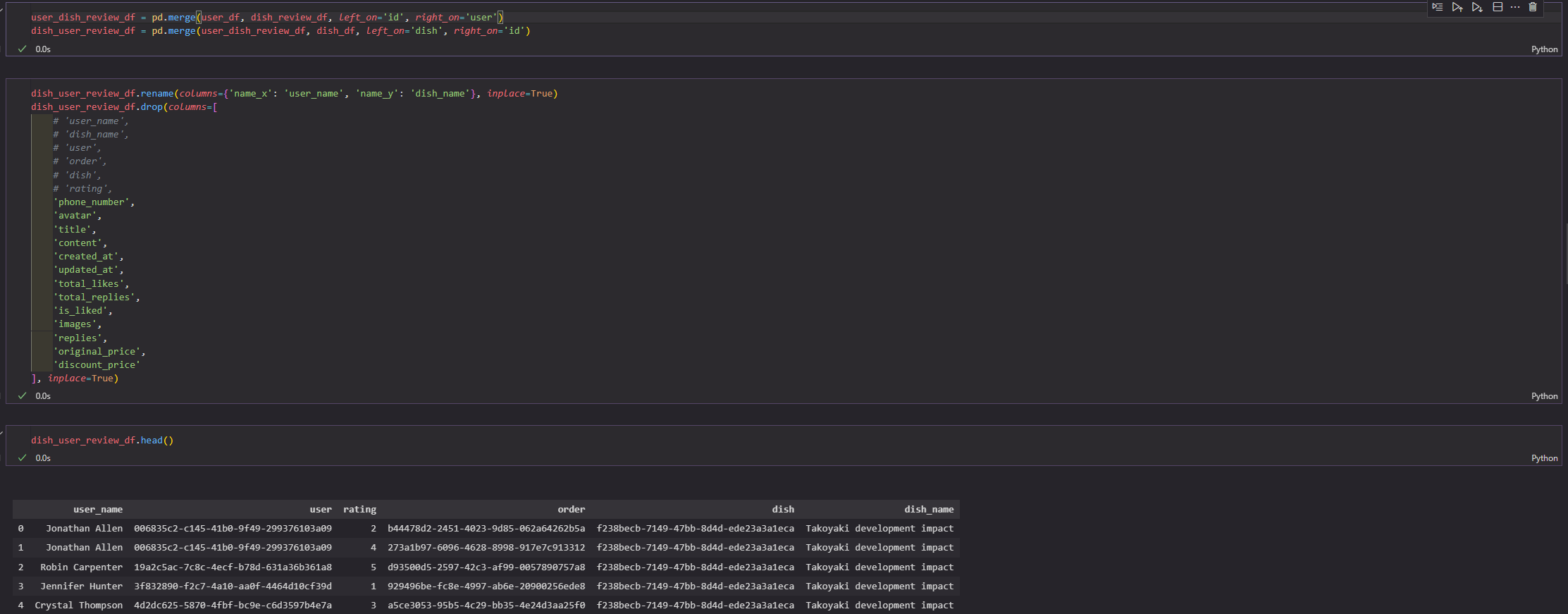
#### Hình 2.4 Trích id của các cột từ các bảng

* Loại bỏ những cột không cần thiết của bảng dish để giảm tải dữ liệu.



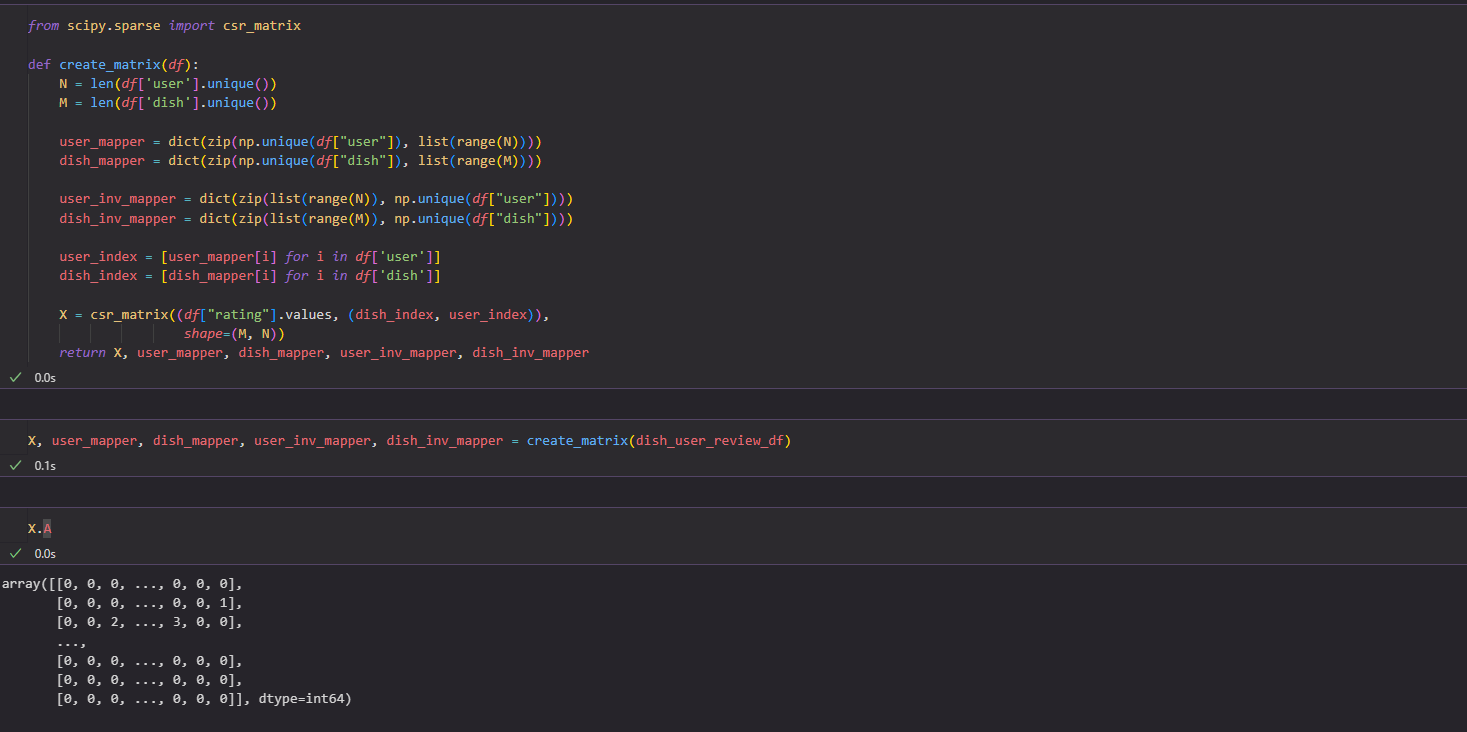
#### Hình 2.5 Loại bỏ cột

* Kết hợp dữ liệu từ nhiều nguồn của 3 bảng dish\_review\_df, dish\_df, user\_df.Loại bỏ các cột không cần thiết, đổi tên các cột để dễ hiểu hơn và đơn giản hóa cấu trúc dữ liệu. Từ kết quả trên, với mỗi dòng của bảng sẽ có đầy đủ thông tin về người đánh giá, món ăn được đánh, đánh giá từ những thông tin đó có thể xây dựng được ma trận user, dish.



#### Hình 2.6 Bảng đầy đủ để xây dựng ma trận

* Xây dựng ma trận user-dish: Là một trong những yếu tố quan trọng trong Collaborative Filtering. Mỗi hàng trong ma trận đại diện cho một người dùng, và mỗi cột đại diện cho một món ăn. Các giá trị trong ma trận này sẽ là các đánh giá của người dùng đối với món ăn (hoặc giá trị tương tác, ví dụ như số lần người dùng đã chọn món ăn).



#### Hình 2.7 Ma trận

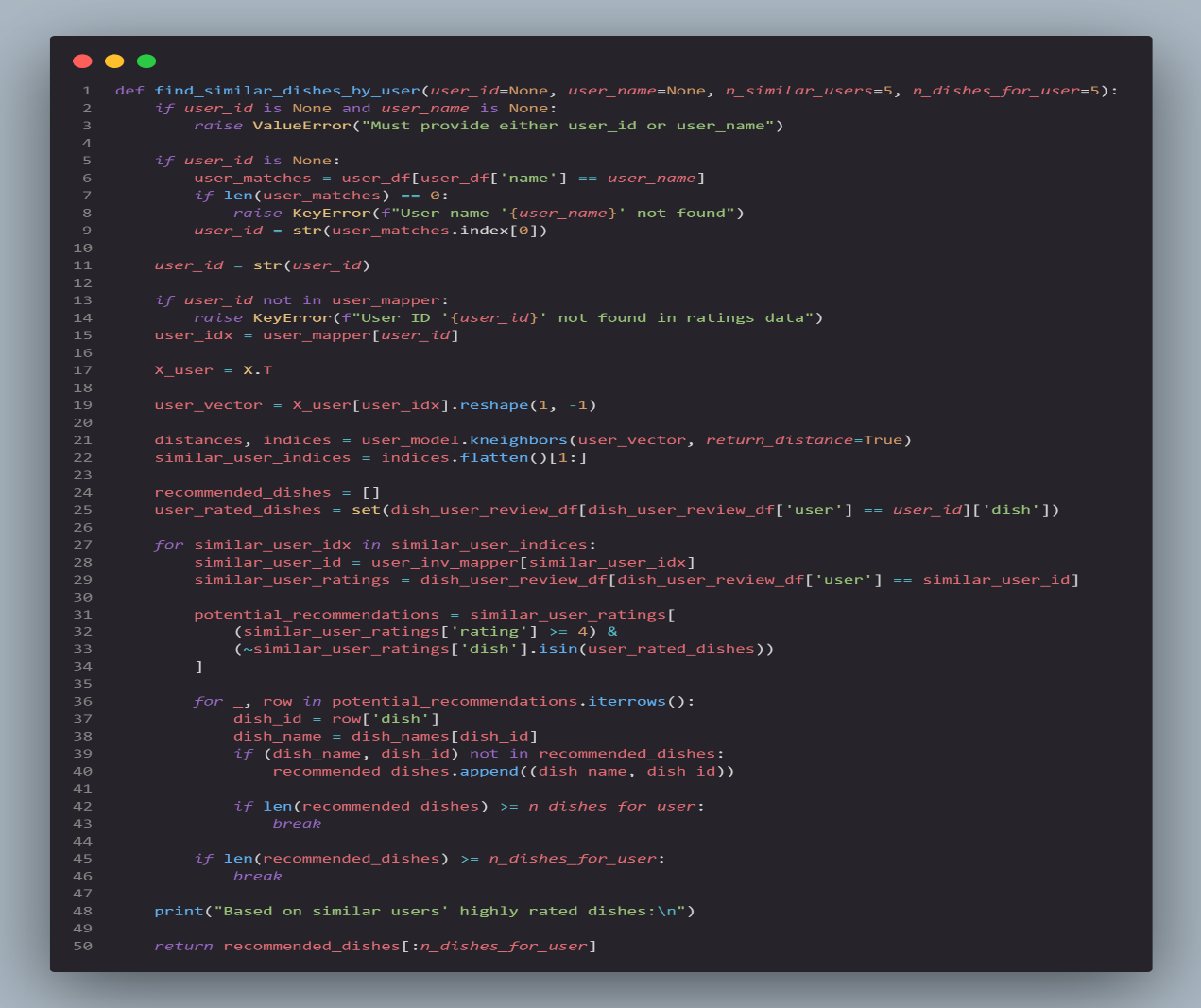
### 2.2.2. Huấn luyện Collaborative Filtering

Xây dựng mô hình Collaborative Filtering bằng cách sử dụng các thuật toán k-Nearest Neighbors (k-NN).

* k-Nearest Neighbors (k-NN): Thuật toán này tìm kiếm những người dùng (hoặc món ăn) có sự tương đồng nhất định trong ma trận tương tác. Khoảng cách tương đồng có thể được tính bằng các phương pháp Cosine similarity.



#### Hình 2.8 Huấn luyện mô hình (Item-Based)

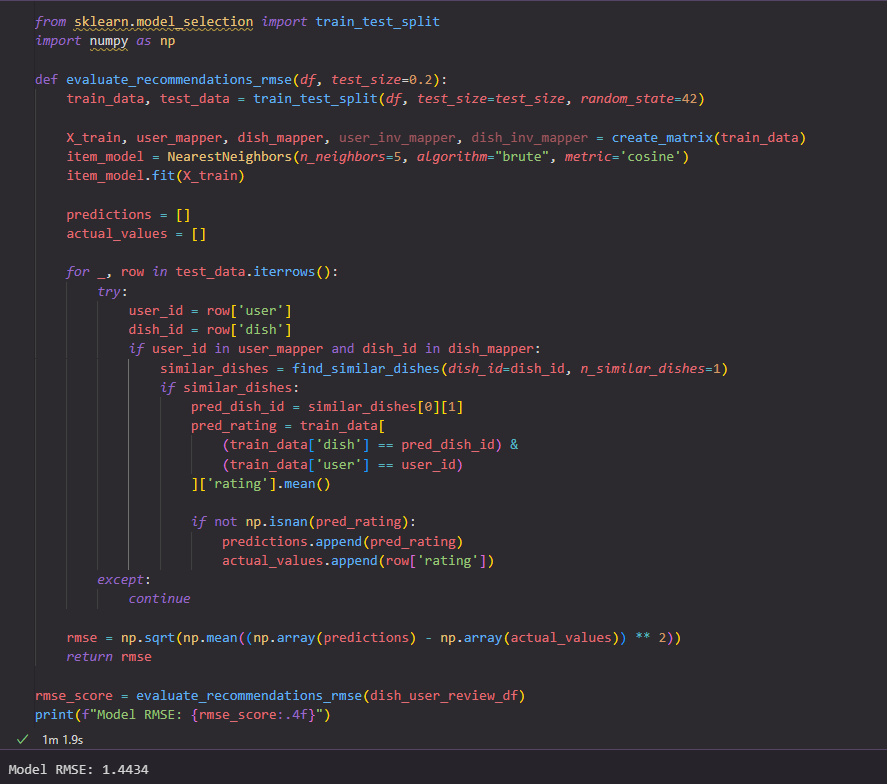


#### Hình 2.9 Huấn luyện mô hình (User-Based)

### 2.2.3 Đánh giá mô hình

Để đánh giá hiệu quả của mô hình Collaborative Filtering, Root Mean Square Error (RMSE) sẽ được sử dụng cho mô hình này. Phương pháp này sẽ đo lường độ sai lệch giữa các giá trị dự đoán và giá trị thực tế.

Kết quả đánh giá mô hình sau khi thực hiện đánh giá được điểm RMSE = 1.44 Trong hệ thống rating từ 1-5 sao, đây là mức điểm trung bình. Để hiểu rõ hơn nếu người dùng đánh giá món ăn 5 sao, mô hình có thể dự đoán trong khoảng 3.56-5 sao. Nguyên nhân chủ yếu là dữ liệu còn khá ít chưa và Collaborative Filtering đơn giản.



#### Hình 2.10 Đánh giá mô hình

## 2.3. Tích hợp API thời tiết

Thời tiết là một yếu tố quan trọng ảnh hưởng đến hành vi tiêu dùng, đặc biệt là trong lĩnh vực dịch vụ đặt đồ ăn. Nhiều nghiên cứu đã chỉ ra rằng điều kiện thời tiết có thể ảnh hưởng lớn đến loại thực phẩm mà người tiêu dùng chọn mua. Ví dụ, vào những ngày lạnh, người dùng có xu hướng đặt các món ăn nóng như súp hoặc đồ nướng, trong khi vào những ngày nóng, họ thường ưa chuộng các món ăn mát như salad hoặc trái cây tươi. Vì vậy, việc tích hợp API thời tiết vào ứng dụng Đặt đồ ăn không chỉ giúp cải thiện trải nghiệm người dùng mà còn tối ưu hóa hệ thống đề xuất sản phẩm.

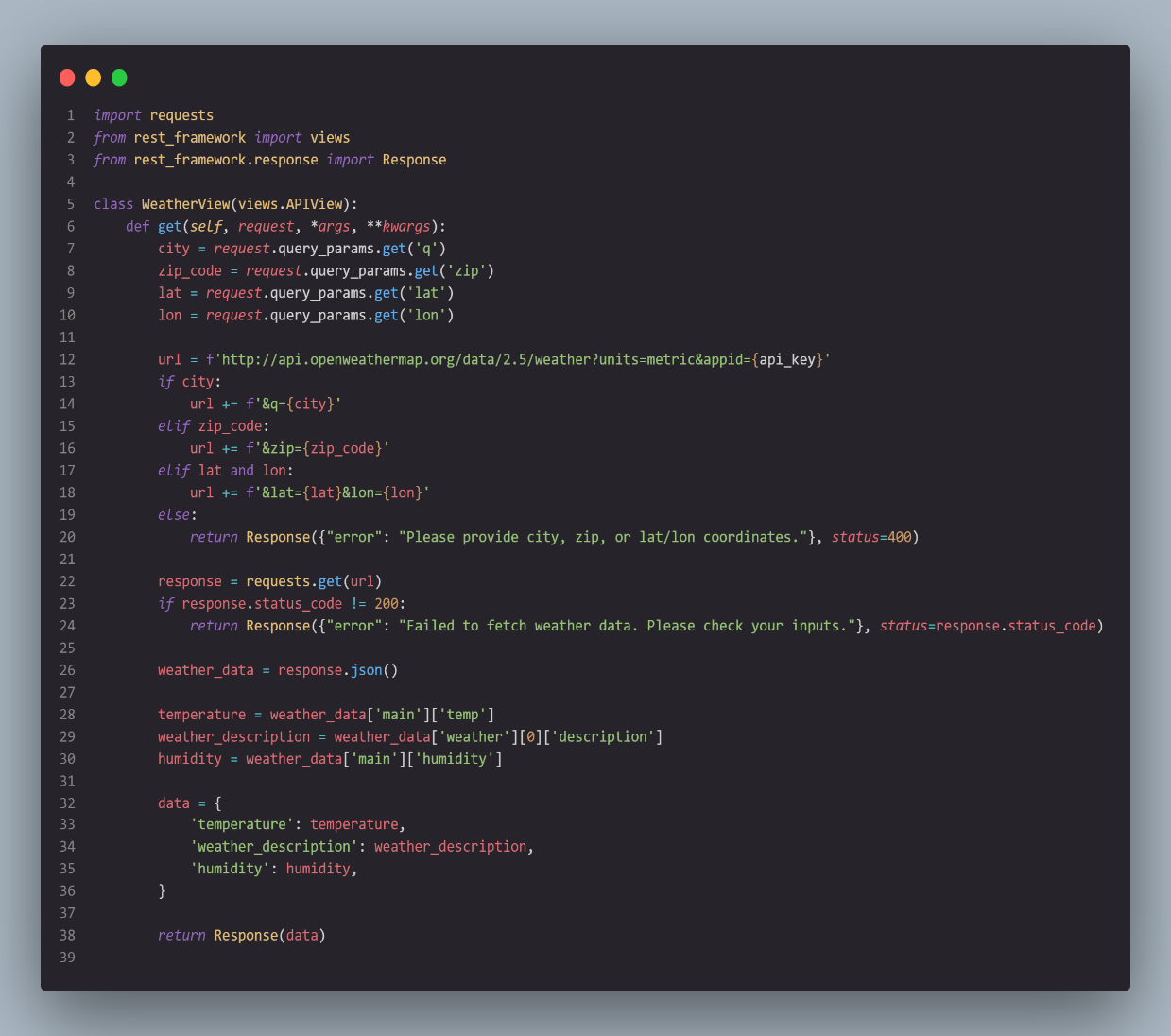
### 2.3.1. API thời tiết và quy trình tích hợp

Lựa chọn API thời tiết: Để tích hợp thông tin thời tiết vào ứng dụng, cần lựa chọn một dịch vụ API thời tiết phù hợp. Một số API thời tiết phổ biến hiện nay bao gồm OpenWeatherMap, Weatherstack, và AccuWeather. Mỗi API này cung cấp các dữ liệu thời tiết chi tiết, từ nhiệt độ, độ ẩm, đến các điều kiện thời tiết như mưa, tuyết, hay gió.

* OpenWeatherMap API: Cung cấp dữ liệu thời tiết chi tiết theo thời gian thực và dự báo trong 7 ngày, với khả năng tích hợp miễn phí cho các ứng dụng không yêu cầu quá nhiều yêu cầu API mỗi ngày. API này có thể cung cấp thông tin về nhiệt độ, độ ẩm, áp suất, cũng như thông tin chi tiết về trạng thái thời tiết.
* Dựa trên yêu cầu của ứng dụng và thuận tiện trong việc tích hợp, OpenWeatherMap API sẽ được chọn để tích hợp vì độ phổ biến, độ chính xác cao, và khả năng miễn phí cho các mức độ sử dụng cơ bản.

Quy trình tích hợp API thời tiết được chia thành các bước cơ bản sau:

1. **Đăng ký và lấy API Key**: Đầu tiên, cần đăng ký tài khoản trên OpenWeatherMap và lấy API key, là chuỗi mã hóa duy nhất giúp xác định và quản lý các yêu cầu đến API.
2. **Tích hợp API vào ứng dụng**:
   * **Backend (Python + Django Rest Framework)**: Backend của hệ thống sẽ xử lý các yêu cầu API từ ứng dụng di động. Khi nhận được yêu cầu về thời tiết, backend sẽ sử dụng API key để gửi yêu cầu đến OpenWeatherMap API thông qua HTTP request (GET).



#### Hình 2.11 API Tích hợp OpenWeatherAPI

* + **Frontend (Flutter + Dart)**: Ứng dụng Flutter sẽ gửi yêu cầu tới backend thông qua API vừa tích hợp trên, nhận kết quả từ backend và hiển thị thông tin thời tiết cho người dùng.

1. **Xử lý dữ liệu thời tiết**: Khi nhận được dữ liệu từ API thời tiết, backend sẽ xử lý thông tin (nhiệt độ, trạng thái thời tiết, độ ẩm) và trả về dữ liệu cho client.

### Ứng dụng thời tiết trong đề xuất sản phẩm

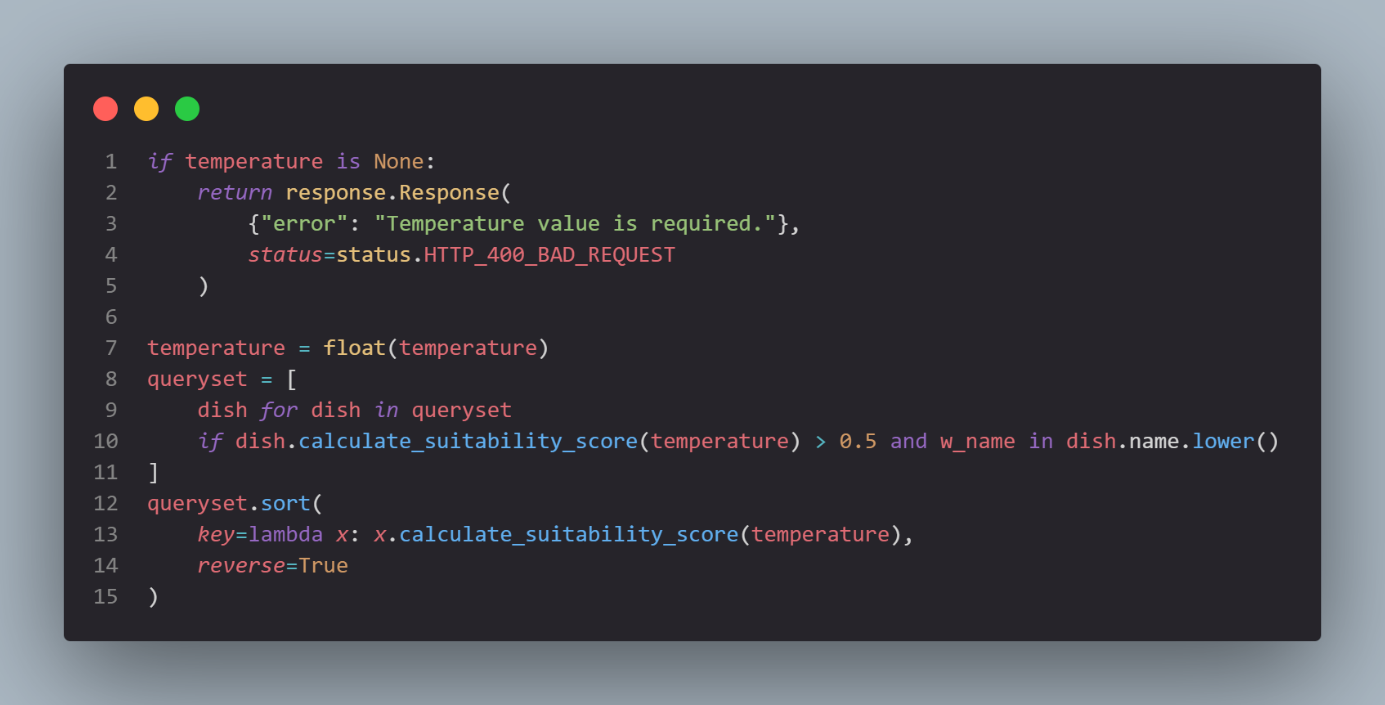
Điều kiện thời tiết có thể ảnh hưởng đến sự lựa chọn món ăn của người dùng. Ví dụ, trong những ngày mưa hoặc lạnh, người dùng có xu hướng chọn các món ăn nóng như súp, hầm, cháo. Ngược lại, vào những ngày nóng, các món ăn nhẹ, mát như salad, trái cây, hay đồ uống giải khát thường được ưa chuộng hơn.

Ứng dụng vào hệ thống đề xuất:Dựa trên thông tin thời tiết thu thập được từ API thời tiết, hệ thống có thể dựa vào dữ liệu về món ăn đã được phân loại sẵn dựa trên nhiệt độ kết hợp với các yếu tố thời tiết để đưa ra các đề xuất món ăn phù hợp. Cụ thể, nếu hệ thống nhận thấy rằng nhiệt độ ngoài trời thấp và có mưa, nó sẽ ưu tiên gợi ý các món ăn nóng. Ngược lại, khi thời tiết nóng, các món ăn nhẹ và mát sẽ được ưu tiên hơn.

Cơ chế đề xuất chủ yếu dựa vào thuộc tính nhiệt độ đã được phân loại sẵn của mỗi món ăn và từ nhiệt độ lấy được từ API sau đó sẽ tính toán ra một tham số gọi là điểm phù hợp với điểm phù hợp lớn hơn 0.5 (có thể chấp nhận được) từ điểm sẽ tìm được những món ăn phù hợp nhất đối với người dùng.



#### Hình 2.12 Tính độ phù hợp



#### Hình 2.13 Áp dụng

Cải tiến trải nghiệm người dùng với thời tiết: Việc tích hợp thời tiết vào hệ thống đề xuất có thể giúp cải thiện sự hài lòng của người dùng, vì họ sẽ nhận được các đề xuất phù hợp hơn với điều kiện thời gian thực. Hệ thống có thể hiển thị các món ăn theo thời tiết trong phần giao diện người dùng, giúp người dùng dễ dàng tìm thấy các món ăn phù hợp mà không cần phải tìm kiếm.

## 2.4. Tối ưu hóa trải nghiệm nhà hàng

Trải nghiệm của nhà hàng trong một ứng dụng đặt đồ ăn không chỉ được thể hiện qua giao diện người dùng và hiệu suất hệ thống mà còn thông qua khả năng phân tích và hiểu rõ các thông số hoạt động của nhà hàng. Một trong những yếu tố quan trọng trong việc tối ưu hóa trải nghiệm nhà hàng là khả năng cung cấp thông tin chi tiết về doanh thu, giúp các nhà dễ dàng theo dõi và theo dõi tình hình kinh doanh.

### 2.4.1 Thiết kế API cho thống kê doanh thu

API được thiết kế nhằm cung cấp các thống kê kinh doanh của nhà hàng qua các giai đoạn thời gian khác nhau (hàng ngày, hàng tháng, hàng năm). Dưới đây là tư tưởng của thiết kế này:

* Tùy biến cao: API cho phép lựa chọn giai đoạn thời gian linh hoạt (hàng ngày, hàng tháng, hàng năm) thông qua tham số time\_range. Có thể kết hợp các bộ lọc bổ sung như ngày cụ thể (day), tháng (month), năm (year).
* Hỗ trợ thời gian thực: API sử dụng múi giờ của nhà hàng để đảm bảo các thống kê được tính toán phù hợp với khung giờ hoạt động thực tế.
* Cung cấp dữ liệu chi tiết: Kết quả trả về bao gồm tổng doanh thu và tổng số lượng đơn hàng, mà còn chi tiết hóa theo từng giai đoạn thời gian (giờ, ngày, tháng). Các chỉ số bổ sung như tỷ lệ hủy đơn và giá trị trung bình mỗi đơn hàng giúp nhà hàng nắm bắt hiệu suất một cách toàn diện.

### 2.4.2. Thiết kế giao diện thống kê doanh thu

Giao diện Thống kê Doanh thu được thiết kế với mục tiêu giúp các nhà hàng dễ dàng theo dõi và phân tích doanh thu theo các khung thời gian khác nhau (ngày, tuần, tháng, năm) hoặc theo từng loại sản phẩm/nhóm món ăn. Những nguyên tắc sau được áp dụng:

* Trực quan hóa dữ liệu: Sử dụng biểu đồ đường để hiển thị doanh thu theo thời gian, giúp nhà hàng quan sát xu hướng kinh doanh. Các nhãn trục X hiển thị thông tin thời gian phù hợp với khung thời gian đã chọn (giờ, ngày, tháng, hoặc năm). Trục Y hiển thị giá trị doanh thu, được điều chỉnh động dựa trên dữ liệu.
* Thông tin tổng quan: Hiển thị tổng doanh thu, tổng số đơn hàng, số lượng đơn hàng bị hủy, và tỷ lệ hủy dưới dạng thẻ thông tin. Cung cấp thông tin về giá trị trung bình của mỗi đơn hàng.
* Tương tác nâng cao: Bộ lọc cho phép người dùng thay đổi khung thời gian theo ý muốn (daily, monthly, yearly).

# Chương 3. Đánh giá và kết quả

## 3.1. Các trường hợp đánh giá

Để đánh giá hiệu quả của các tính năng trong ứng dụng, các trường hợp thí nghiệm được xây dựng với mục đích kiểm tra từng chức năng của hệ thống, bao gồm:

1. Đánh giá với mô hình Collaborative Filtering:
   * Mục tiêu: Kiểm tra độ chính xác của các gợi ý món ăn được cung cấp cho người dùng.
   * Phương pháp: Chạy mô hình Collaborative Filtering với bộ dữ liệu người dùng và món ăn, kiểm tra khả năng gợi ý chính xác các món ăn mà người dùng có khả năng yêu thích.
   * Kết quả mong đợi: Tăng tỷ lệ người dùng chọn món ăn theo các đề xuất của hệ thống.
2. Đánh giá với API thời tiết:
   * Mục tiêu: Đánh giá sự hiệu quả của việc tích hợp API thời tiết trong việc đưa ra các gợi ý sản phẩm dựa trên thời tiết.
   * Phương pháp: Chạy thử nghiệm trong các điều kiện thời tiết khác nhau (nắng, mưa, lạnh) và kiểm tra các đề xuất món ăn của hệ thống.
   * Kết quả mong đợi: Cung cấp các gợi ý món ăn phù hợp với điều kiện thời tiết, như món nóng khi trời lạnh và món mát khi trời nóng.
3. Đánh giá với giao diện đề xuất sản phẩm theo sở thích và thời tiết:

* Mục tiêu: Kiểm tra mức độ thân thiện, dễ sử dụng và hiệu quả của giao diện trong việc cung cấp các gợi ý sản phẩm dựa trên sở thích cá nhân và điều kiện thời tiết.
* Phương pháp: Thực hiện khảo sát nhóm người dùng thử nghiệm (bao gồm người mới và người quen), thu thập phản hồi về trải nghiệm sử dụng, kiểm tra độ chính xác và khả năng tùy chỉnh các gợi ý. Đồng thời, đánh giá mức độ tương thích của giao diện trên các thiết bị khác nhau và so sánh hiệu quả khi đề xuất chỉ dựa trên sở thích cá nhân so với khi kết hợp thêm thông tin thời tiết.
* Kết quả mong đợi: Giao diện thân thiện, trực quan, dễ sử dụng và cung cấp các gợi ý phù hợp với sở thích và thời tiết. Hoạt động mượt mà trên mọi thiết bị, đảm bảo trải nghiệm nhất quán, đồng thời tích hợp thông tin thời tiết giúp nâng cao độ chính xác và sự hài lòng của người dùng.

1. Đánh giá với giao diện thống kê doanh thu:
   * Mục tiêu: Kiểm tra tính dễ sử dụng và hiệu quả của giao diện thống kê doanh thu đối với nhà hàng.
   * Phương pháp: Thử nghiệm với các loại dữ liệu khác nhau về doanh thu và đánh giá khả năng tương tác của người quản lý với giao diện thống kê.
   * Kết quả mong đợi: Giao diện phải dễ sử dụng, trực quan và cung cấp thông tin chính xác về doanh thu.

## 3.2. Đánh giá mô hình Collaborative Filtering

Độ chính xác của mô hình: sử dụng các chỉ số RMSE đã cho ra kết quả 1.4434 mức trung bình.

Đánh giá độ phủ của mô hình: Coverage đề cập đến khả năng của mô hình trong việc đưa ra các gợi ý cho nhiều người dùng và món ăn khác nhau. Mô hình đã có khả năng đưa ra các gợi ý cho tất cả các loại người dùng với sở thích khác nhau.

Tốc độ và hiệu quả của mô hình: Mô hình Collaborative Filtering cần được huấn luyện trên một bộ dữ liệu nhỏ và thời gian trả về gợi ý trong khoảng thời gian chấp nhận được.

Kết quả đánh giá cho thấy mô hình Collaborative Filtering chưa được tốt những cũng có thể chấp nhận được, với độ phủ đạt mức mong muốn, còn độ chính xác thì mức trung bình. Các gợi ý được đưa ra phù hợp với sở thích của người, và tốc độ phản hồi của mô hình có thể chấp nhận được để đảm bảo trải nghiệm người dùng không bị gián đoạn.

## 3.3. Đánh giá hiệu quả tích hợp API thời tiết

Độ chính xác của dữ liệu thời tiết: API thời tiết cung cấp dữ liệu chính xác về nhiệt độ, độ ẩm và tình trạng thời tiết tại khu vực người dùng. Chúng tôi đánh giá độ chính xác của dữ liệu bằng cách so sánh với các nguồn dữ liệu thời tiết khác.

Khả năng cập nhật thời gian thực: Một yếu tố quan trọng trong việc tích hợp API thời tiết là khả năng cập nhật thời gian thực. API cần có khả năng cung cấp thông tin thời tiết liên tục và kịp thời để đề xuất món ăn phù hợp với điều kiện hiện tại.

Tác động của API thời tiết đến gợi ý món ăn: Chúng tôi đã thử nghiệm hệ thống trong các điều kiện thời tiết khác nhau (nắng, mưa, lạnh) và đánh giá mức độ phù hợp của các gợi ý món ăn. Các kết quả cho thấy hệ thống đã cung cấp các món ăn phù hợp như món súp nóng trong mùa đông, hoặc món nước trái cây lạnh trong mùa hè.

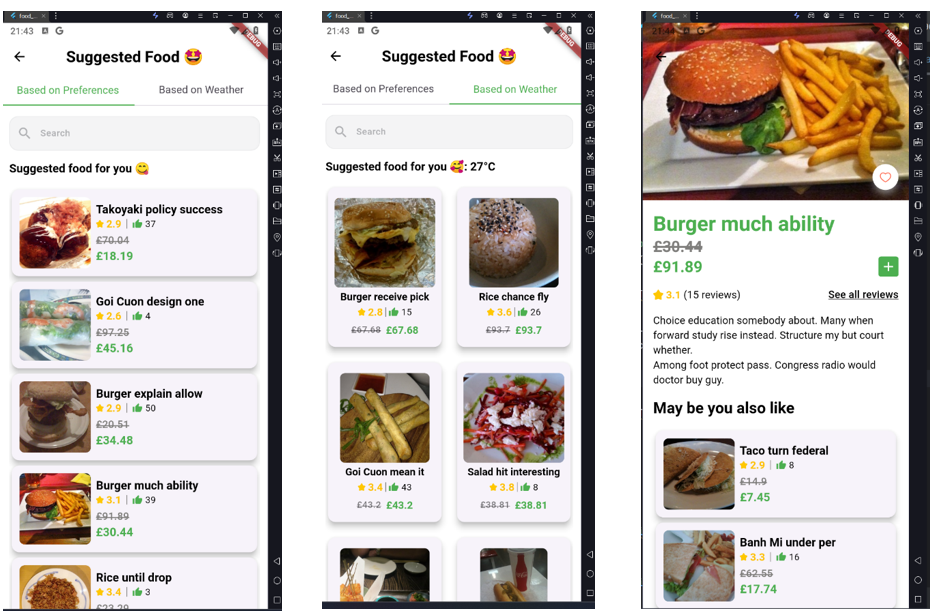
Kết quả thí nghiệm cho thấy tích hợp API thời tiết hoạt động hiệu quả trong việc cá nhân hóa các gợi ý món ăn. Hệ thống không chỉ dựa trên sở thích người dùng mà còn xtôi xét yếu tố thời tiết, giúp người dùng có những trải nghiệm lựa chọn món ăn thú vị và phù hợp hơn.

## 3.4 Đánh giá giao diện đề xuất sản phẩm theo sở thích và thời tiết

Trực quan và dễ sử dụng: Giao diện được thiết kế với bố cục rõ ràng, dễ dàng nhận biết các gợi ý món ăn. Các gợi ý được hiển thị nổi bật, giúp người dùng dễ dàng chọn món ăn phù hợp với sở thích và theo thời tiết. Hệ thống sử dụng các biểu tượng hoặc hình ảnh minh họa món ăn, giúp người dùng hiểu rõ hơn về lựa chọn.

Hiển thị thông tin liên quan: Cung cấp thông tin chi tiết về món ăn như giá cả, thành phần, và hình ảnh minh họa, giúp người dùng đưa ra quyết định dễ dàng hơn. Giao diện cũng hiển thị thông tin thời tiết hiện tại (nhiệt độ, tình trạng trời nắng/mưa/lạnh), tạo sự liên kết giữa thời tiết và các món ăn được đề xuất. Với chi tiết món ăn sẽ hiển thị các món đề xuất cùng với món ăn đó.

Khả năng tương tác và phản hồi nhanh: Giao diện phản hồi nhanh, đảm bảo trải nghiệm người dùng không bị gián đoạn. Dữ liệu gợi ý được tải nhanh ngay cả khi người dùng chuyển đổi giữa các tùy chọn sở thích hoặc điều kiện thời tiết.



#### Hình 3.1 Giao diện đề xuất

## 3.5. Đánh giá giao diện thống kê doanh thu

Dễ sử dụng và trực quan: Các nhà hàng có thể dễ dàng tương tác với hệ thống thống kê doanh thu thông qua giao diện. Đánh giá qua khả năng người quản lý tiếp cận các thông tin doanh thu, lợi nhuận, và các chỉ số quan trọng khác.

Khả năng tùy chỉnh và lọc dữ liệu: Giao diện cung cấp các công cụ để nhà hàng có thể lọc và tùy chỉnh dữ liệu theo nhu cầu. Việc lọc dữ liệu theo thời gian, theo món ăn, hoặc khu vực giao hàng giúp nhà hàng có cái nhìn rõ ràng về tình hình kinh doanh.

Khả năng tương tác và phản hồi nhanh: Giao diện hiện thị mượt mà và API được tối ưu để trả về kết quả nhanh chóng, không gây gián đoạn trong quá trình làm việc của nhân viên nhà hàng.

|  |  |  |
| --- | --- | --- |
| Hình 3.2 Giao diện thống kê |  |  |

# KẾT LUẬN

## 1. Kết quả đạt được

Kết quả: https://github.com/mduc-2610/Food-Delivery-App

Trong quá trình nghiên cứu và xây dựng ứng dụng đặt đồ ăn, tôi đã đạt được những kết quả quan trọng cả về mặt lý thuyết lẫn thực tiễn.

### 1.1. Về mặt lý thuyết

Phân tích và tổng hợp được các khái niệm liên quan đến ứng dụng đặt đồ ăn, bao gồm cấu trúc tổng quát, vai trò của hệ thống đề xuất sản phẩm, và tích hợp công nghệ liên quan như Collaborative Filtering và API thời tiết.

Tìm hiểu và ứng dụng các thuật toán lọc cộng tác User-based và Item-based nhằm tối ưu hóa việc đề xuất món ăn dựa trên sở thích và hành vi người dùng.

Đề xuất cách tích hợp thông tin thời tiết vào mô hình gợi ý, giúp ứng dụng cung cấp các đề xuất sát với nhu cầu thực tế hơn.

### 1.2. Về mặt thực tiễn

Phát triển thành công một ứng dụng đặt đồ ăn với các tính năng tối ưu hóa trải nghiệm người dùng, bao gồm: hệ thống đề xuất món ăn thông minh, tích hợp thông tin thời tiết và giao diện thống kê doanh thu cho nhà hàng.

Tích hợp giao diện trực quan, thân thiện, giúp nhà hàng dễ dàng theo dõi doanh thu, biểu đồ thống kê và các dữ liệu nâng cao liên quan đến hoạt động kinh doanh.

Đánh giá hiệu quả của hệ thống đề xuất và tính năng tích hợp thông qua các thí nghiệm cụ thể, cho thấy cải thiện đáng kể trong việc đáp ứng nhu cầu người dùng và tăng cường hiệu quả kinh doanh của nhà hàng.

## 2. Hạn chế

Mặc dù đã đạt được nhiều kết quả khả quan, nghiên cứu vẫn tồn tại một số hạn chế như:

* Bộ dữ liệu sử dụng để huấn luyện mô hình Collaborative Filtering còn giới hạn, chưa bao quát đầy đủ hành vi người dùng trong các tình huống thực tế phức tạp.
* API thời tiết được tích hợp còn mang tính chất cơ bản, chưa khai thác sâu các yếu tố khác như mùa, giờ trong ngày, hoặc các biến động bất thường có thể ảnh hưởng đến nhu cầu đặt món.
* Hệ thống chưa hoàn toàn tối ưu cho quy mô lớn, đặc biệt với lượng người dùng và nhà hàng đông đảo, có thể gặp giới hạn về hiệu suất xử lý và tốc độ phản hồi.

## 3. Hướng phát triển

Để tiếp tục cải thiện và mở rộng nghiên cứu, các hướng phát triển trong tương lai bao gồm:

* Mở rộng bộ dữ liệu: Thu thập thêm dữ liệu người dùng và nhà hàng để mô hình hóa chi tiết hơn hành vi và nhu cầu thực tế.
* Nâng cấp hệ thống đề xuất: Áp dụng các kỹ thuật học sâu (Deep Learning) và kết hợp các phương pháp lai (Hybrid Recommendation Systtôis) để cải thiện chất lượng gợi ý.
* Tích hợp nâng cao API thời tiết: Khai thác thêm các yếu tố thời gian, mùa vụ, và phân tích tác động của các yếu tố thời tiết chi tiết hơn đến hành vi đặt hàng.
* Tối ưu hóa hiệu suất: Nghiên cứu các giải pháp phân tán, sử dụng điện toán đám mây để đảm bảo hệ thống hoạt động hiệu quả ở quy mô lớn.
* Cá nhân hóa sâu hơn: Tích hợp thêm các yếu tố cá nhân hóa như sở thích ăn uống, dị ứng thực phẩm hoặc ngân sách của người dùng để nâng cao trải nghiệm.

Kết quả của đề tài không chỉ mang lại ý nghĩa khoa học mà còn có tính ứng dụng thực tiễn cao, góp phần nâng cao trải nghiệm người dùng trong lĩnh vực đặt đồ ăn trực tuyến và hỗ trợ nhà hàng tối ưu hóa hiệu quả kinh doanh.

# TÀI LIỆU THAM KHẢO

1. DResnick, P., Iacovou, N., Suchak, M., Bergstrom, P., & Riedl, J. (1994). GroupLens: An Open Architecture for Collaborative Filtering of Netnews. Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work (CSCW). Retrieved from https://dl.acm.org/doi/10.1145/192844.192905
2. Badrul Sarwar, G., Karypis, G., Konstan, J., & Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. Proceedings of the 10th international conference on World Wide Web. Retrieved from https://dl.acm.org/doi/10.1145/371920.372071
3. Kiều Công Minh. Luận văn thạc sỹ kỹ thuật Ứng Dụng Representation Learning phát hiện tấn công Botnet.
4. R. Yera Toledo, A. A. Alzahrani and L. Martínez, "A Food Recommender System Considering Nutritional Information and User Preferences," in IEEE Access, vol. 7, pp. 96695-96711, 2019, doi: 10.1109/ACCESS.2019.2929413.
5. Alalwan, A. A. (2020). Mobile food ordering apps: An empirical study of the factors affecting customer e-satisfaction and continued intention to reuse. International Journal of Information Management, 50, 28–44.
6. Yang, X., He, Y., Zhang, H., & Chen, Y. (2016). A Hybrid Food Recommendation System Based on User Preference and Context Awareness. IEEE International Conference on Big Data Analysis.
7. Python Scikit-learn Documentation. (2024). Collaborative Filtering. <https://scikit-learn.org/stable/modules/collaborative_filtering.html>
8. Flutter. (2024). Flutter Documentation. https://flutter.dev/docs
9. Django Rest Framework. (2024). Django Rest Framework Documentation. <https://www.django-rest-framework.org/>
10. Google Maps API. (2024). Google Maps Platform. https://developers.google.com/maps.