Trường Đại học Khoa học tự nhiên – Khoa Công nghệ thông tin.

Đồ án thực hành cuối kỳ

Nhập môn khoa học dữ liệu – Introduction to Data Science.

Nhóm 9
Tháng 11, 2024.

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN

ĐỒ ÁN THỰC HÀNH CUỐI KÌ

Bộ môn: Nhập môn khoa học dữ liệu.
Tên đề tài:

"Phân tích mối quan hệ giữa các loại gia vị, nguyên liệu và khả năng kết hợp của chúng trong các món ăn".

STT nhóm: 9.

Thành viên:

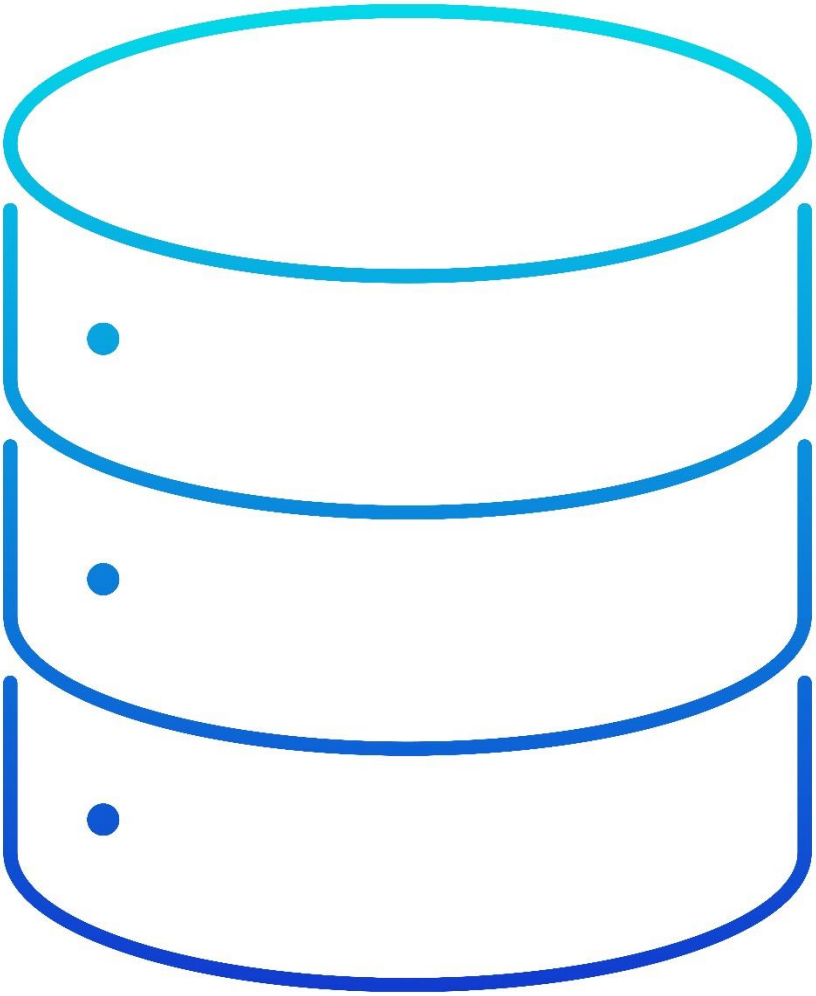
1. 22120384 – Nguyễn Đình Trí.
2. 22120398 – Vũ Hoàng Nhật Trường.
3. 22120412 – Nguyễn Anh Tường.
4. 22120424 – Phạm Ngọc Bảo Uyên.
5. 22120449 – Lê Nguyễn Huyền Vy.

Thông tin chung:

1. **Bộ môn:** Nhập môn khoa học dữ liệu.
2. **Giảng viên lý thuyết:** Thầy Lê Ngọc Thành.
3. **Giảng viên thực hành:** Thầy Lê Nhựt Nam.
4. **Mã lớp:** 22_21.
5. **STT nhóm:** 9.
6. **Danh sách thành viên:**
    a. 22120384 – Nguyễn Đình Trí.
    b. 22120398 – Vũ Hoàng Nhật Trường.
    c. 22120412 – Nguyễn Anh Tường.
    d. 22120424 – Phạm Ngọc Bảo Uyên.
    e. 22120449 – Lê Nguyễn Huyền Vy.
7. **Link github repository:** "Click here to go to our github repository."

MỤC LỤC

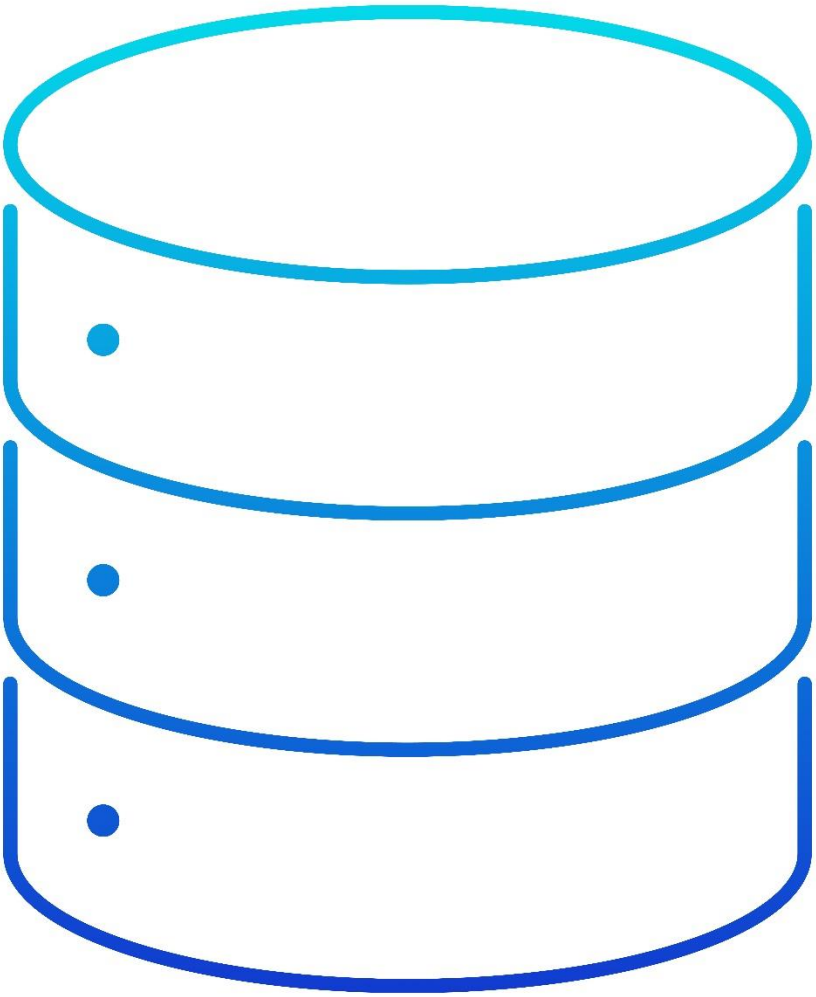
Section 0: *Bảng phân công công việc.*



I. Bảng phân công công việc:

Công việc	Người thực hiện	Tiến độ	Đóng góp	Note
Data Collection	Nguyễn Anh Tường	100%	90%	Các bạn chạy cào giúp vì bị trang web chặn.
	Nguyễn Đình Trí		2.5%	
	Vũ Hoàng Nhật Trường		2.5%	
	Phạm Ngọc Bảo Uyên		2.5%	
	Lê Nguyễn Huyền Vy		2.5%	
Data Preprocessing	Lê Nguyễn Huyền Vy	100%	95%	
	Nguyễn Anh Tường		5%	Thảo luận cách làm
Data Exploration	Nguyễn Đình Trí	100%	50%	
	Phạm Ngọc Bảo Uyên		50%	
Data Modeling & Evaluation	Nguyễn Anh Tường	66.66%	50%	Context Machine + Model 01
	Vũ Hoàng Nhật Trường		50%	Model 02 + 03

Section 1: *Data Collection.*



I. Các trường dữ liệu cần cào:

1. Tên bài viết:

Ví dụ:

- Title: Đổi Vị Với Phở Gà Trộn Lạ Miệng, Thơm Ngon Ăn Mãi Mà Không Thấy Ngán
- Title: Mì Ý Thịt Viên Sốt Cà Chua Nữ Hoàng Của Nền Ẩm Thực Nước Ý.
- Title: Khoai Tây Tẩm Gia Vị Nướng Kiểu Âu.

2. Tên của các nguyên liệu của món ăn đó:

Ví dụ:

Ingredients: ½ con gà ta 1 miếng gừng nhỏ (nạo vỏ, đập dập) 250ml nước 1 tsb muối 2 tbs xì dầu 2 tbs đường 1 tsp giấm tỏi 1 củ hành tây 500g bánh phở tươi 1 nắm giá Rau mùi, hành lá Hành phi Lạc rang

Ingredients: 6-8 củ khoai tây 1 tbs dầu olive ½ tsp muối ½ tsp tiêu xay ½ bột ớt paprika ½ tsp bột tỏi Hương thảo tươi

II. Định dạng của dữ liệu sau khi cào về:

Các món ăn hay bài viết khác nhau sẽ được phân tách với nhau bởi một dòng:

‘----------‘

Ảnh minh họa:

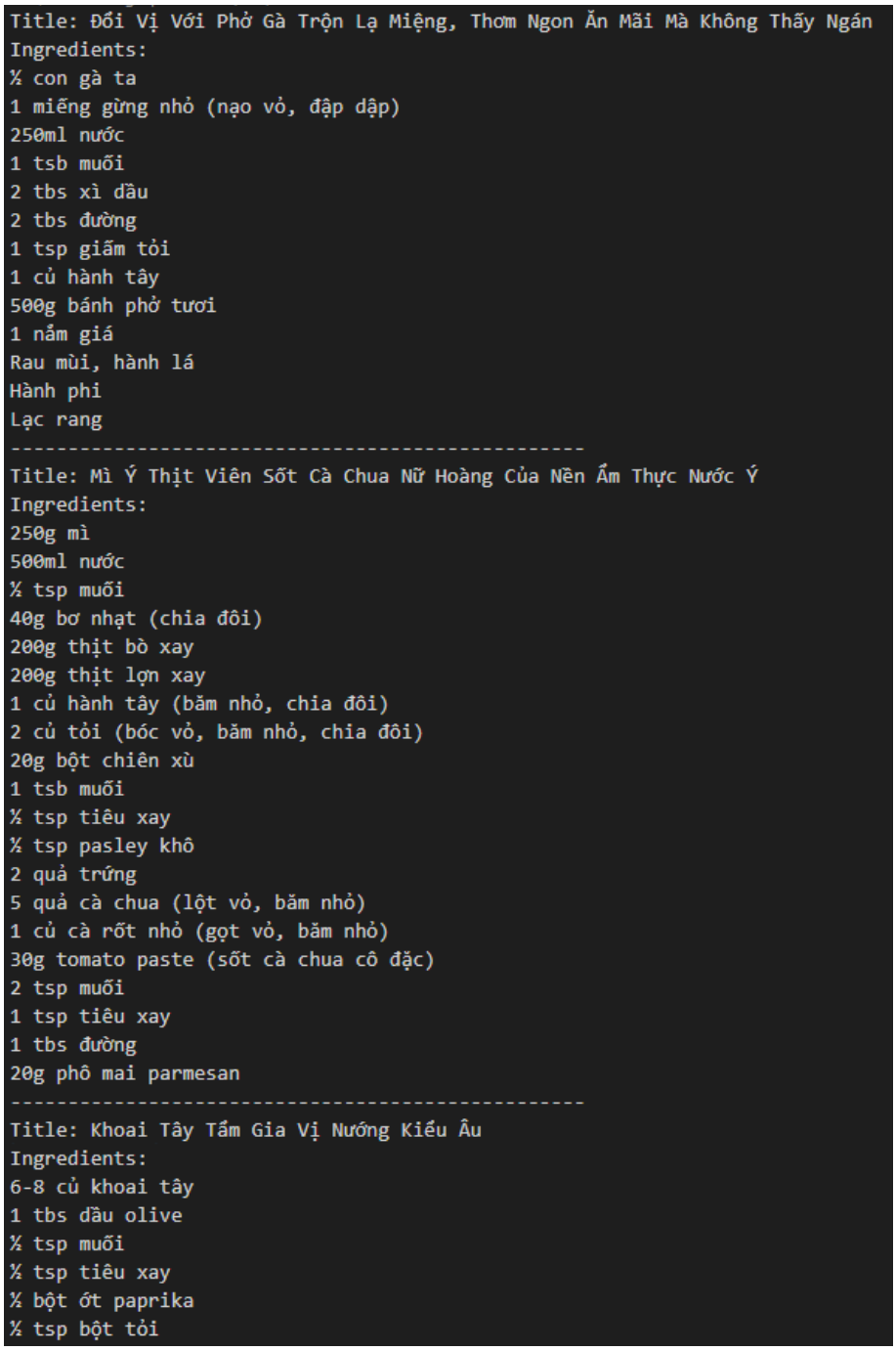


Figure 1. Ảnh minh họa cho tập dữ liệu sau khi cào.

III. Quá trình thực hiện:

1. Kokotaru.com:

a. Nhận xét: Sử dụng tương tác cuộn chuột để load trang và load bài viết.

b. Chi tiết các bước thực hiện cào dữ liệu từ Kokotaru:

Bước 1: Sử dụng selenium để giả lập trình duyệt chrome.

Bước 2: Sử dụng trình duyệt chrome với thao tác cuộn chuột qua tất cả các trang con của kokotaru.

Bước 3: Lấy hết tất cả các liên kết bài viết từ trang web.

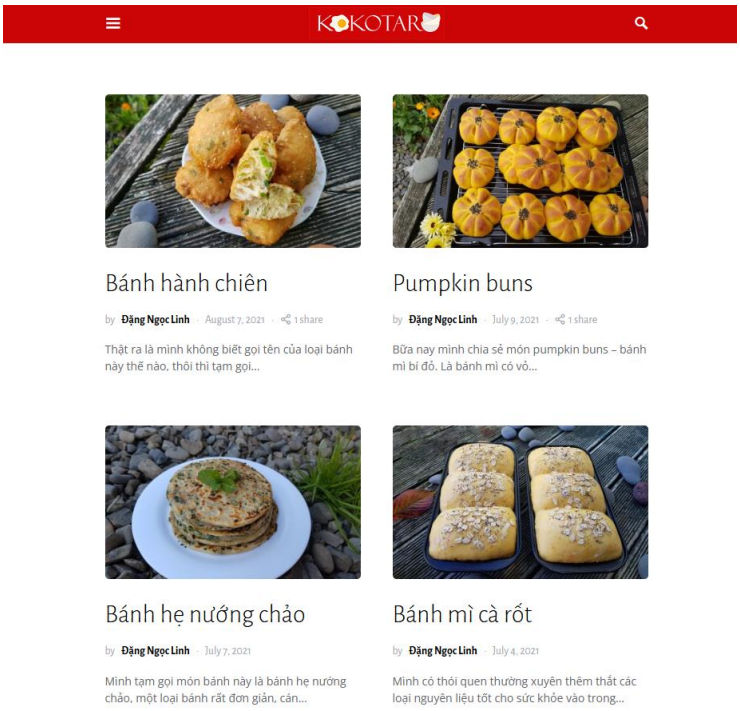


Figure 2. Ảnh trang web kokotaru.

Bước 4: Truy cập vào từng bài viết và lấy ra thành phần của các món ăn.

c. Một số vấn đề xảy ra:

- Người viết bài và code web viết ở quá nhiều dạng trình bày khác nhau.
- Các bài viết không có nguyên liệu.
- Các bài viết nguyên liệu được chèn sẵn vào trong tiêu đề.
- Các bài viết nguyên liệu được trộn vào trong dòng văn giới thiệu.
- Các bài viết sử dụng nguyên liệu A,B,C nhưng không dùng đến.

2. Kitchenart.com

a. Nhận xét: Các trang được phân chia theo dạng bình thường, gồm 38 trang. Trong mỗi trang sẽ có đường dẫn đến trang tiếp theo.

b. Chi tiết các bước cào dữ liệu:

Bước 1: Thực hiện lấy hết đường dẫn dẫn đến các trang chứa bài viết.

Bước 2: Thực hiện vào từng trang tổng hợp lấy ra hết các đường dẫn dẫn đến các bài viết có trong trang đó.

Bước 3: Thực hiện vào từng đường dẫn của bài viết để lấy dữ liệu.

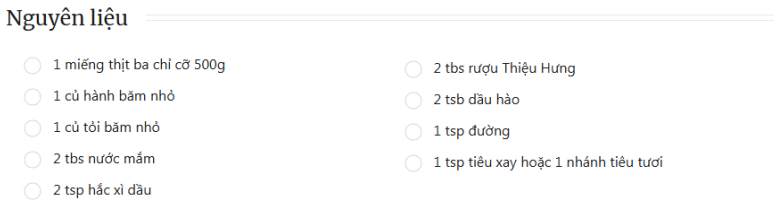


Figure 3. Ảnh mục nguyên liệu của kitchenart.com

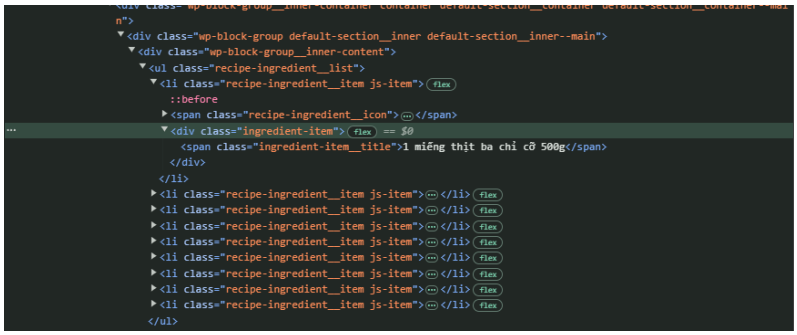


Figure 4. Mã HTML của phần nguyên liệu.

c. Một số vấn đề xảy ra:

- Trang web cấm truy cập với tốc độ rất nhanh và nhiều lần.
- Trung bình truy cập lần lượt 5-10 bài với tốc độ nhanh thì trang web đã phát hiện và tiến hành chặn truy cập. Đã thử truy cập 1 – 2 – 3 bài nhưng vẫn bị chặn.
- Thời gian chờ mở chặn rất lâu. Vẫn chưa có con số chính xác nhưng dao động trong khoảng từ 5 – 8 phút.

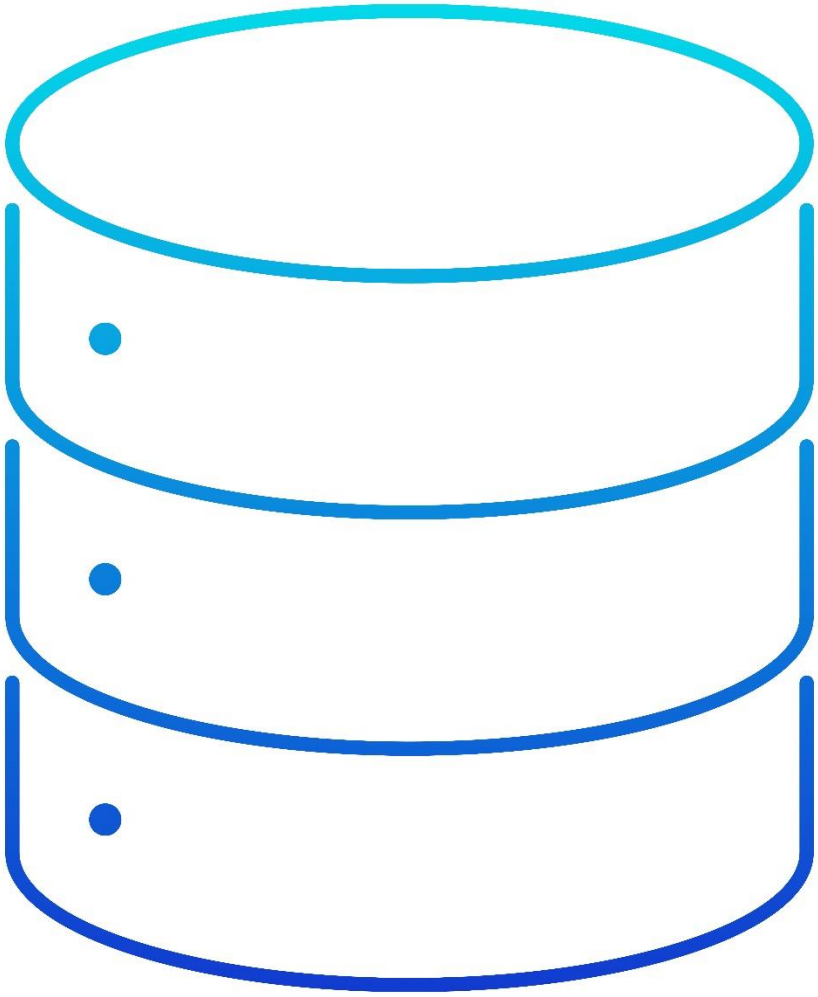
IV. Kết quả:

Kokotaru: 399 / 468 bài viết.

Kitchenart: 682 / 760 bài viết.

Tổng cộng: 1081 bài viết.

Section 2: *Data Cleaning and Normalizing*



I. Data Cleaning

1. Dịch tập tin

Để thuận tiện cho việc tiền xử lý và khám phá dữ liệu mà không phải quan tâm quá nhiều đến vấn đề xử lý unicode, các tập tin lưu dữ liệu được cào về từ hai website sẽ được dịch sang Tiếng Anh bằng Google Translate.

Trong quá trình dịch thuật, dữ liệu sẽ được kiểm tra lại để bảo đảm tính đồng nhất về bản dịch tên của các nguyên liệu.

2. Làm sạch dữ liệu:

Từng dòng dữ liệu sẽ được xử lý để đảm bảo chỉ giữ lại đúng tên nguyên liệu, lược bỏ các thông tin khác như lượng từ, tính từ, mô tả, cách sơ chế nguyên liệu đó.

a. Ví dụ:

Dữ liệu trước khi làm sạch

Ingredients: 6-8 củ khoai tây 1 tbs dầu olive ½ tsp muối ½ tsp tiêu xay ½ bột ớt paprika ½ tsp bột tỏi Hương thảo tươi

Dữ liệu sau khi làm sạch:

Ingredients: khoai tây dầu olive muối tiêu xay bột ớt paprika bột tỏi hương thảo tươi

b. Cách thực hiện

- Bởi vì từng các dòng nguyên liệu sẽ có các format khác nhau nên chúng ta không thể nào lọc lấy tên nguyên liệu bằng cách chỉ sử dụng những lệnh đơn giản như “bỏ đi kí số chỉ số lượng”, “lấy chuỗi con từ vị trí i đến vị trí j”…
- Cho nên, việc làm sạch nguyên liệu cần được hỗ trợ bởi những từ khóa, những ký tự đặc biệt.

Ví dụ:

- Những từ ngữ là lượng từ (half, clove, handful,…) hay đơn vị đo (g, kg, cup, bowl,…)
- Những từ ngữ chỉ tính chất hoặc cách sơ chế nguyên liệu đó (roasted, chopped, peeled,…)
- Những từ ngữ, kí hiệu là giới từ đánh dấu phần chú thích (*for* decorations, (),…)

• Các bước làm sạch một dòng dữ liệu:

- Chuẩn hóa Unicode, đưa chuỗi về kí tự thường.
- Loại bỏ nội dung trong ngoặc đơn.
- Tách thành các nguyên liệu khác nhau khi gặp các kí tự hoặc từ ngữ đóng vai trò phân cách.
- Loại bỏ kí tự không phải ASCII.
- Chuẩn hóa các phân số bị lỗi.
- Loại bỏ số lượng và đơn vị đo lường.
- Thay thế các kí tự không cần thiết còn sót lại thành khoảng trắng.
- Loại bỏ các thành phần là chú thích về cách sơ chế, tính chất của nguyên liệu.

-Chuẩn hóa về dạng số ít của nguyên liệu.

3. Định dạng của dữ liệu sau khi làm sạch:

Sau khi làm sạch, mỗi dòng nguyên liệu sẽ trả về một hoặc nhiều nguyên liệu đã được xử lý. Từ đó chuyển đến bước tiếp theo để tiến hành chuẩn hóa dữ liệu.

II. Data Normalizing

1. “Gộp” dữ liệu

Sau khi làm sạch, mỗi món ăn sẽ có tên món ăn và một danh sách chứa các nguyên liệu của món ăn đó.

Nhiệm vụ của chúng ta là tạo ra một danh sách chứa tất cả các nguyên liệu phân biệt có trong tất cả những món ăn được cào về từ website ban đầu. Ta sẽ tiến hành tạo ra một dataframe với từng dòng biểu thị cho từng món ăn, từng cột biểu thị cho từng nguyên liệu.

2. Chuẩn hóa dữ liệu

Sau khi dữ liệu được đọc từ tập tin, ta đưa thông tin của từng món ăn bao gồm tên món ăn và danh sách nguyên liệu vào một danh sách (list). Mỗi phần tử trong đó sẽ có kiểu dictionary với key ‘title’ lưu tên món ăn và key ‘ingredients’ lưu danh sách các nguyên liệu.

Các bước thực hiện:

- Đọc file và xử lý từng món ăn (dựa vào dòng phân cách 50 kí tự ‘-‘)
- Lưu tên món ăn và danh sách nguyên liệu đã được làm sạch vào combined_data

```
combined_data.append({
```

```
                'title': title,
                'ingredients': clean_ingredients
            })
```

- Tạo dataframe từ combined_data
- Lấy danh sách các nguyên liệu phân biệt trong dataframe và sắp xếp lại theo thứ tự bảng chữ cái.
- Tạo một dataframe nhị phân gồm cột "Name of dish" và các cột còn lại là tên các nguyên liệu có trong danh sách nguyên liệu phân biệt.
- Điền giá trị vào dataframe: Nếu món ăn nằm ở dòng i có sử dụng nguyên liệu ở cột j thì vị trí tương ứng đó có giá trị bằng 1, ngược lại bằng 0.

III. Kết quả

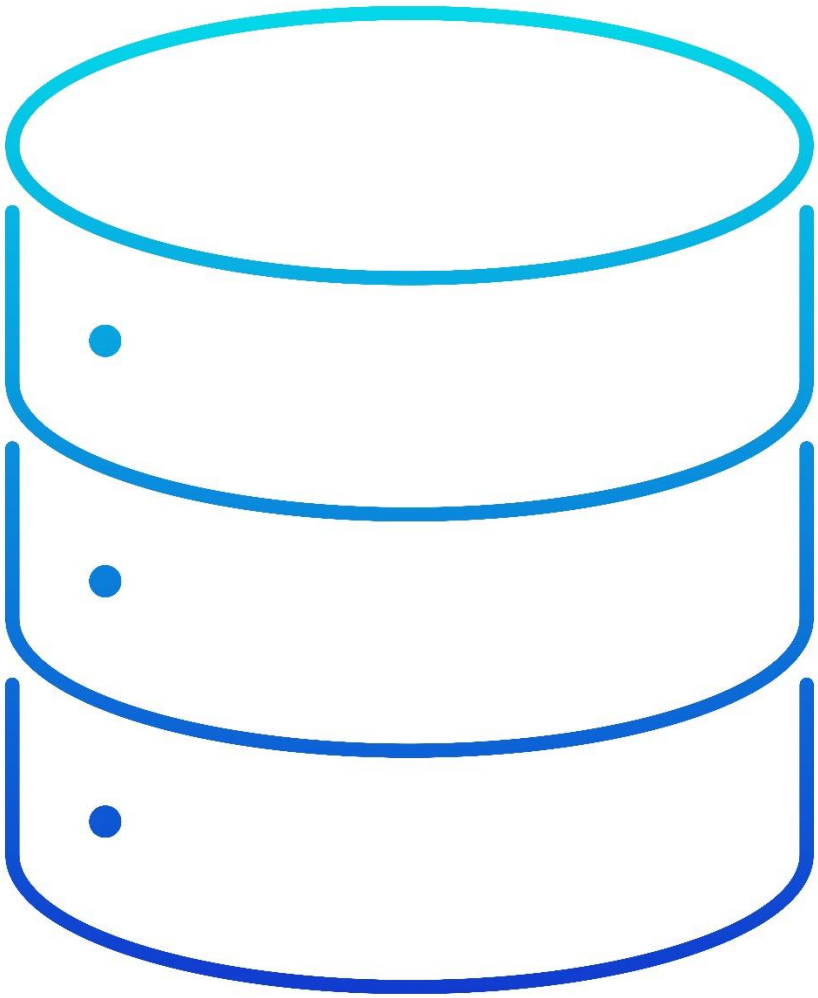
Sau khi thực hiện làm sạch và chuẩn hóa dữ liệu, ta kiểm tra xem số lượng dòng của dataframe ta tạo ra có khớp với số lượng món ăn đã cào được ở phần Data Collection hay không.

```
binary_df, all_ingredients = normalize_recipes_to_dataframe (file_paths,units,key_words,black_list,lemmatizer)

print(f"Number of dishes: {binary_df.shape[0]}")
print(f"Number of ingredients: {binary_df.shape[1]}")
```

Kết quả thu được: Number of dishes: 682 (khớp với số lượng món ăn trong file dữ liệu được cào về từ web Kitchenart).

Đối với website Kokotaru, bởi vì cấu trúc đặc biệt của web đã dẫn đến việc trong một số trường hợp, kết quả trả về từ việc cào dữ liệu không phải là nguyên liệu của món ăn. Trong một số trường hợp khác, tên nguyên liệu không được trình bày thành các dòng phân biệt mà được biểu thị trong một đoạn văn miêu tả, cho nên phải mất thời gian lâu hơn cũng như phương pháp phức tạp hơn để giải quyết vấn đề này.

Phần 3: *Khám phá dữ liệu.*



I. Tiền xử lý dữ liệu:

Dựa trên data đã thu thập được, ta bắt đầu quy trình tiền xử lý và phân tích để hiểu rõ hơn về tập dữ liệu. Tiến hành đặt một vài câu hỏi sơ bộ:

- Dữ liệu gồm mấy dòng, mấy cột?
- Ý nghĩa của mỗi dòng? Có tồn tại các dòng trùng nhau hay không?
- Ý nghĩa của từng cột? Kiểu dữ liệu hiện tại của từng cột? Có cột nào có kiểu dữ liệu không phù hợp hay không?
- Đối với các cột có kiểu số, mỗi giá trị trong cột đóng góp ra sao? Có tồn tại giá trị thiếu không? Min? Max? Có tồn tại các giá trị bất thường trong cột hay không?
- Đối với các cột phân loại, có tồn tại giá trị thiếu không? Có bao nhiêu giá trị phân biệt?
- Kiểm tra lại toàn bộ tập dữ liệu đã phù hợp hay chưa?

II. Phân tích dữ liệu:

Phần khám phá dữ liệu được chia thành 2 phần để chúng ta dễ theo dõi, đó là:

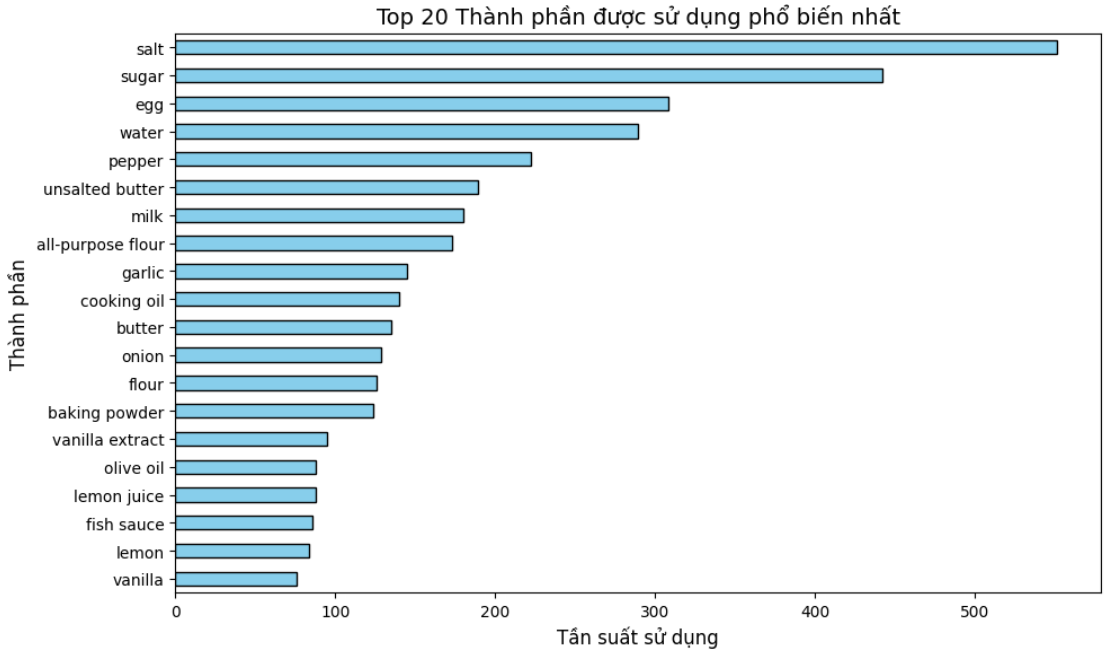
- Phần 1: Phân tích sự phổ biến của các nguyên liệu thành phần và khả năng kết hợp của chúng.
- Phần 2: Phân tích độ phức tạp của các món ăn dựa trên số lượng nguyên liệu cũng như độ phổ biến của chúng.

Sau đây chúng ta đi vào phần thứ nhất:

1. Phân tích sự phổ biến của các nguyên liệu thành phần và khả năng kết hợp của chúng.

1.1 Sự phổ biến của thành phần

Dựa vào tập dữ liệu, ta quan sát biểu đồ top 20 thành phần được sử dụng nhiều nhất sau đây:



Nhận xét :

- Thành phần phổ biến nhất:
  - Muối (salt) đứng đầu với tần suất sử dụng cao nhất, vượt trội so với các thành phần khác. Đây là một gia vị cơ bản và không thể thiếu trong hầu hết các món ăn.
- Các thành phần cơ bản khác:
  - Đường (sugar), trứng (egg), nước (water), và tiêu (pepper) cũng có tần suất sử dụng rất cao, cho thấy đây là những nguyên liệu thiết yếu trong cả nấu ăn và làm bánh.
- Chất béo và bột:
  - Các thành phần như bơ nhạt (unsalted butter), sữa (milk), dầu ăn (cooking oil), và các loại bột (all-purpose flour, baking powder, flour) xuất hiện nhiều, thể hiện vai trò quan trọng trong chế biến các món ăn và bánh.
- Gia vị và phụ liệu:
  - Tỏi (garlic), nước cốt chanh (lemon juice), nước mắm (fish sauce), và chiết xuất vani (vanilla extract) là những gia vị, phụ liệu thường dùng để tăng hương vị cho món ăn.
- Đa dạng ứng dụng:
  - Sự kết hợp giữa các thành phần cơ bản (muối, đường, trứng, bột) và các gia vị (tỏi, tiêu, nước mắm) cho thấy tập dữ liệu có thể liên quan đến nhiều loại món ăn, từ món ăn mặn đến món tráng miệng.
- Quan sát bổ sung:
  - Có sự phân hóa rõ rệt giữa các thành phần hàng đầu (muối, đường) và các thành phần xếp cuối (vanilla, lemon), cho thấy tần suất sử dụng không đồng đều.

Một số thông tin hữu ích :

- Thành phần thiết yếu trong nấu ăn
  - Muối (salt) và đường (sugar) là hai thành phần được sử dụng phổ biến nhất, cho thấy chúng là nguyên liệu cơ bản trong hầu hết các công thức nấu ăn, bất kể món ăn mặn hay ngọt.
  - Các thành phần như trứng (egg), nước (water), và tiêu (pepper) cũng rất phổ biến, nhấn mạnh vai trò của chúng trong nhiều loại món ăn từ món chính đến món tráng miệng.
- Đặc điểm của công thức trong dữ liệu
  - Sự xuất hiện đồng thời của bột mì (all-purpose flour), baking powder, và chiết xuất vani (vanilla extract) cho thấy tập dữ liệu này có thể chứa nhiều công thức làm bánh hoặc món ăn cần nướng.
  - Các thành phần như tỏi (garlic), nước mắm (fish sauce), và chanh (lemon) thường liên quan đến các món ăn châu Á hoặc món ăn mặn, cho thấy sự đa dạng về văn hóa ẩm thực.
- Thành phần cơ bản và bổ trợ
  - Bơ nhạt (unsalted butter) và dầu ăn (cooking oil) đều xuất hiện trong danh sách, cho thấy dầu mỡ đóng vai trò quan trọng trong chế biến, từ chiên xào đến nướng.
  - Các thành phần bổ trợ như nước cốt chanh (lemon juice) và vanilla extract ít xuất hiện hơn, chứng tỏ chúng thường là nguyên liệu đặc trưng dùng để gia giảm hương vị.

- Sự phổ biến theo tần suất

- Các thành phần đầu bảng như muối và đường có tần suất sử dụng gấp đôi, thậm chí gấp ba lần so với các thành phần xếp cuối như nước mắm hay vanilla, điều này gợi ý rằng:

- Muối và đường được sử dụng gần như trong mọi công thức.

- Một số thành phần như nước mắm hoặc vanilla chỉ xuất hiện trong những món đặc trưng.

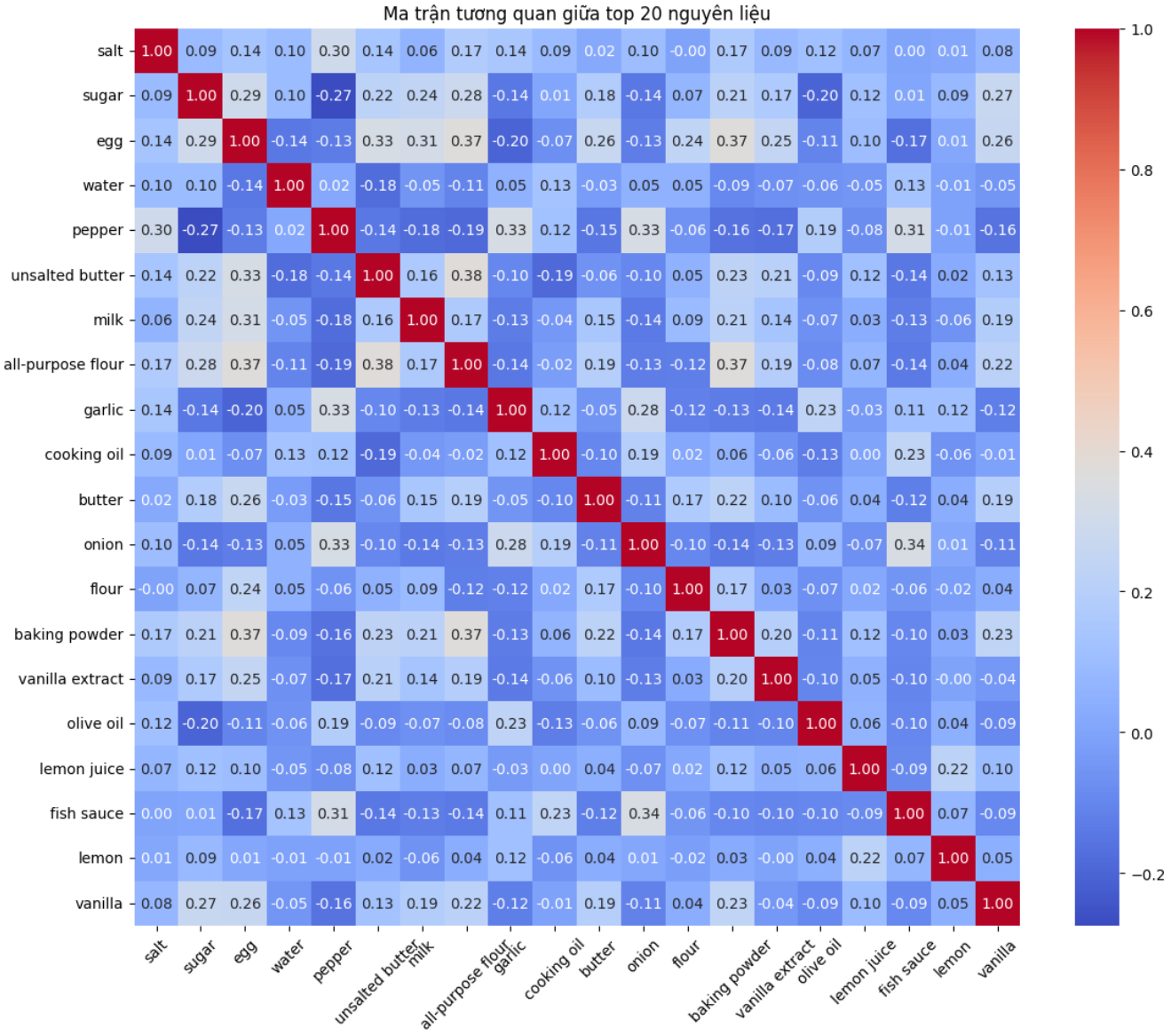
- Ứng dụng thực tế

- Đối với đầu bếp hoặc nhà kinh doanh thực phẩm: Họ có thể ưu tiên trữ lượng lớn các thành phần phổ biến như muối, đường, và trứng vì chúng có tần suất sử dụng cao.

- Đối với người lập kế hoạch sản xuất: Sự phổ biến của các nguyên liệu này có thể giúp xác định trọng tâm cho các chiến dịch quảng cáo hoặc phát triển sản phẩm.

1.2 Khả năng kết hợp của các thành phần phổ biến

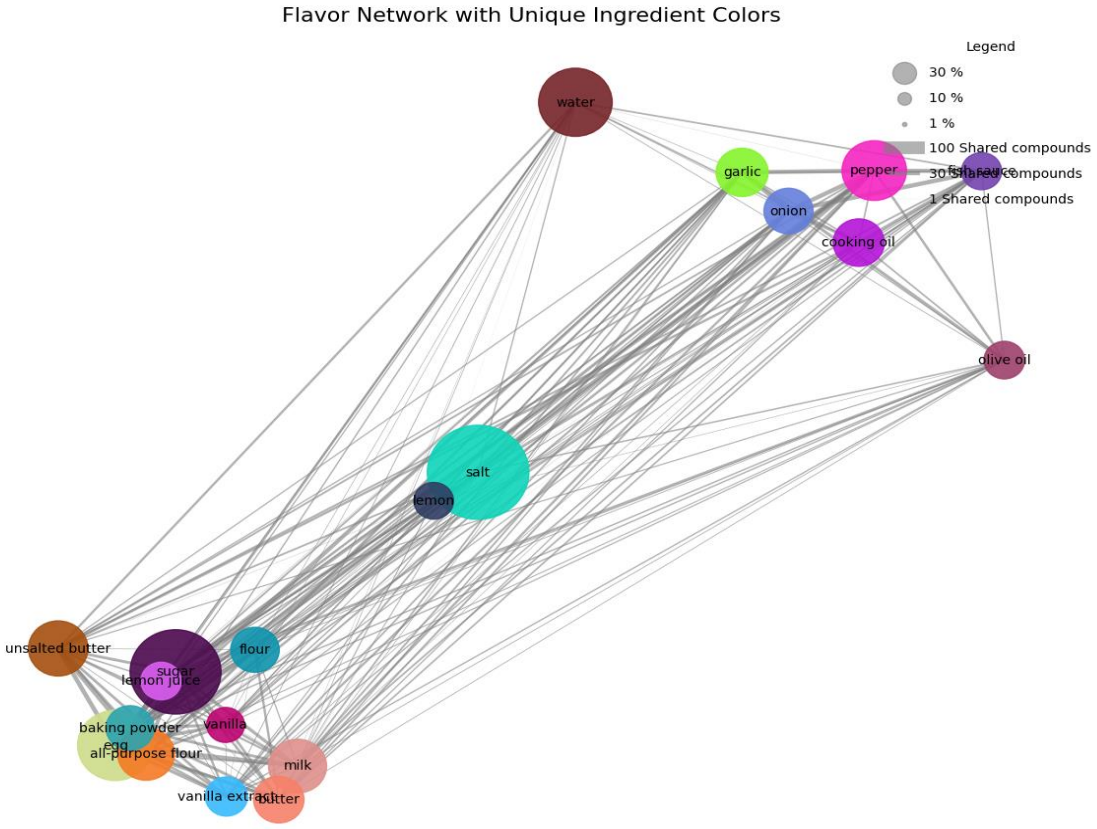
Chúng ta cùng xem qua ma trận tương quan giữa 20 nguyên liệu phổ biến nhất đã lọc ra ở trên:



Nhận xét :

- Tương quan cao giữa các nguyên liệu (màu đỏ đậm)
  - Bột mì đa dụng (all-purpose flour) và baking powder có mức tương quan cao (~0.38): Đây là cặp nguyên liệu thường xuất hiện cùng nhau trong các công thức làm bánh.
  - Bơ nhạt (unsalted butter) và bột mì đa dụng (all-purpose flour): Mức độ tương quan cao (~0.37) cho thấy chúng cũng thường được sử dụng cùng trong các món bánh hoặc nướng.
- Tương quan thấp hoặc gần như không liên quan (màu xanh nhạt)
  - Nước mắm (fish sauce) và các nguyên liệu làm bánh (ví dụ: vanilla extract, baking powder): Tương quan thấp hoặc âm, điều này phản ánh sự khác biệt rõ ràng giữa món ăn mặn và món tráng miệng.
  - Tiêu (pepper) và đường (sugar): Mức tương quan thấp (~-0.27), cho thấy đường và tiêu ít khi được sử dụng cùng nhau.
- Sự liên quan nhóm nguyên liệu
  - Nhóm làm bánh: Các nguyên liệu như bột mì (flour), bột nở (baking powder), vani (vanilla extract), và bơ (butter) có mức độ liên quan cao, phản ánh nhóm nguyên liệu chính cho các món bánh.
  - Nhóm nấu ăn mặn: Nước mắm (fish sauce), tỏi (garlic), tiêu (pepper), và hành (onion) có xu hướng liên quan chặt chẽ hơn, phù hợp với các món ăn mặn trong nấu ăn.
- Nguyên liệu có vai trò trung gian
  - Muối (salt): Có mức tương quan nhẹ dương (~0.1–0.3) với hầu hết các nguyên liệu, cho thấy đây là thành phần cơ bản có thể xuất hiện trong cả món ăn mặn và món ngọt.
  - Dầu ăn (cooking oil): Tương quan tương đối đồng đều với nhiều nguyên liệu khác (~0.1–0.2), thể hiện tính linh hoạt trong cả làm bánh và nấu ăn.
- Ứng dụng thực tế
  - Đối với đầu bếp: Biểu đồ giúp nhận ra các cặp nguyên liệu thường xuất hiện cùng nhau để lên kế hoạch mua sắm và chuẩn bị nguyên liệu hiệu quả hơn.
  - Đối với nhà phát triển sản phẩm thực phẩm: Có thể sử dụng các cặp nguyên liệu có tương quan cao để thiết kế sản phẩm hoặc công thức mới, ví dụ: các sản phẩm bánh nướng nên ưu tiên kết hợp bơ, bột mì và bột nở.
  - Trong phân tích dữ liệu: Các mối quan hệ này có thể giúp tối ưu hóa cách gợi ý công thức trong các ứng dụng nấu ăn thông minh.

Mạng lưới hương vị: Mối quan hệ giữa top 20 nguyên liệu phổ biến



Nhận xét

- Kích thước nút

- Kích thước của nút (node) phản ánh mức độ phổ biến hoặc tầm quan trọng của nguyên liệu trong mạng lưới.

- Water (nước) và salt (muối) là các nút lớn nhất, cho thấy chúng là nguyên liệu phổ biến, được sử dụng rộng rãi và liên kết với nhiều nguyên liệu khác.

- Olive oil (dầu ô liu) cũng là một nguyên liệu quan trọng trong mạng lưới, mặc dù ít liên kết hơn so với nước và muối.

- Màu sắc của nút

- Mỗi nút có màu sắc riêng biệt để dễ nhận biết và đại diện cho các nguyên liệu khác nhau.

- Nhóm làm bánh (flour, sugar, butter, vanilla) thường tập trung gần nhau.

- Nhóm nguyên liệu mặn (pepper, garlic, onion, cooking oil) cũng nằm gần nhau, phản ánh sự tương đồng về cách sử dụng.

- Đường liên kết

- Độ dày của đường liên kết thể hiện mức độ chia sẻ hợp chất hương vị giữa hai nguyên liệu.

- Ví dụ: Salt (muối) có liên kết mạnh với lemon (chanh) và water (nước), phản ánh vai trò quan trọng của muối trong việc cân bằng hương vị.

- Flour (bột mì) và baking powder có kết nối chặt chẽ, điều này phù hợp với các công thức làm bánh.

- Phân cụm nguyên liệu

- Các nguyên liệu có xu hướng tạo thành cụm (cluster) gần nhau dựa trên ứng dụng thực tế:

- Cụm nguyên liệu mặn: Garlic, onion, pepper, cooking oil thường được sử dụng cùng nhau trong các món ăn mặn.

- Cụm nguyên liệu làm bánh: Flour, sugar, vanilla, butter xuất hiện trong các món bánh.

- Ứng dụng

- Phát triển công thức nấu ăn: Các nguyên liệu có liên kết chặt chẽ có thể được kết hợp để tạo ra các món ăn mới hoặc cải thiện hương vị của món ăn hiện tại.

- Phân tích dữ liệu ẩm thực: Dựa vào mạng lưới, các nhà nghiên cứu hoặc đầu bếp có thể tìm hiểu sâu hơn về sự tương thích hương vị giữa các nguyên liệu.

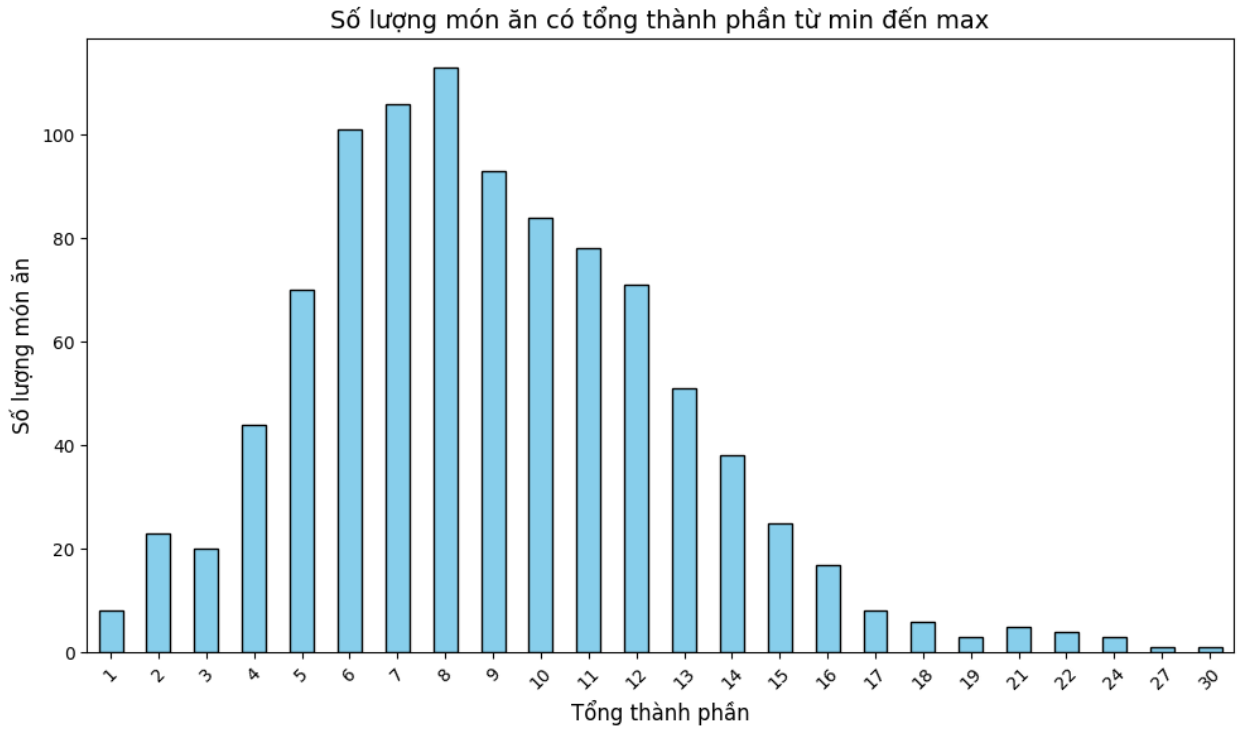
1.3 Phân tích nguyên liệu đặc trưng cho từng món

Ở phần này, nguyên liệu được gọi là đặc trưng nếu nó chỉ xuất hiện 1 lần trong duy nhất 1 món ăn đó

	Name of dish	Distinctive Ingredients
0	Answers to questions about moon cake ingredients	d ingredients
1	Blackberry crumble cheesecake bars	weetbix
2	Chocolate, raspberry and ginger chocolate mous...	gelatin leaf
3	Coconut Cookies	please give test
4	Cookies with jam filling – Rosenmunnar	fruit jam according

2. Phân tích độ phức tạp của các món ăn dựa trên số lượng nguyên liệu cũng như độ phổ biến của chúng

Đầu tiên chúng ta cùng quan sát xem, để tạo ra mỗi món ăn thì số nguyên liệu cần thiết thường là bao nhiêu.

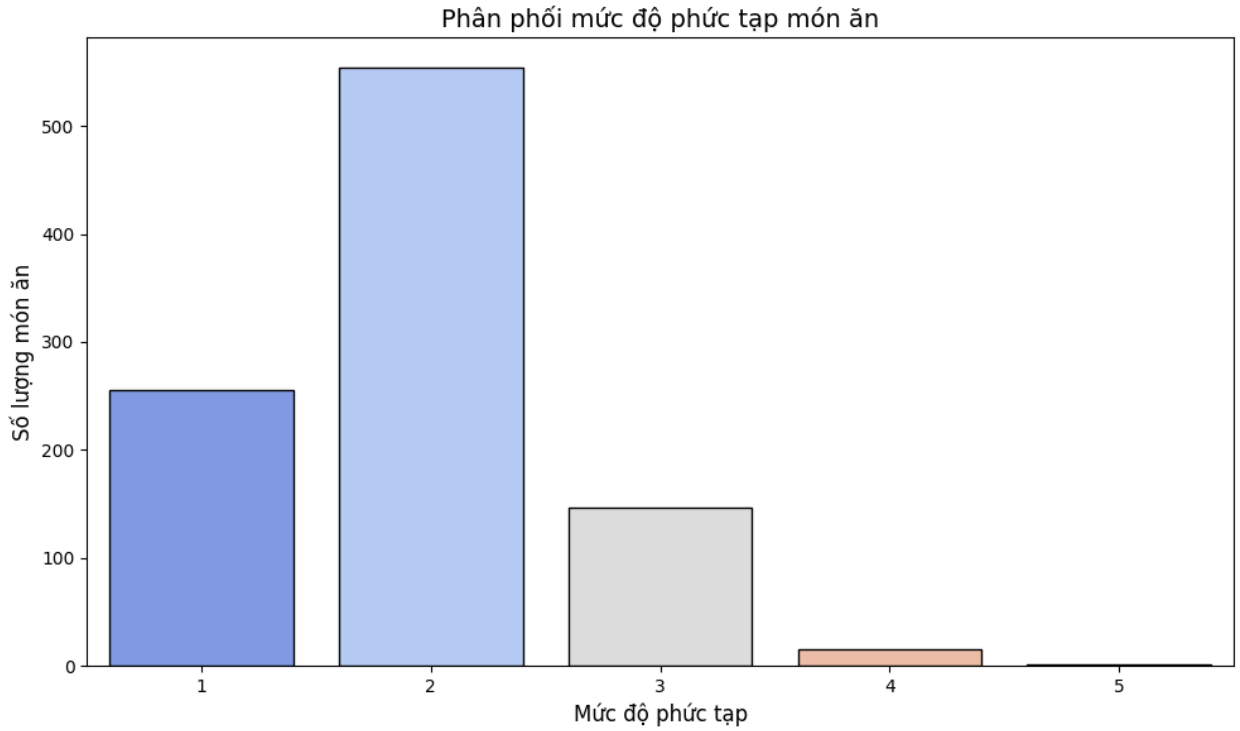


Nhận xét

- Phân phối tổng thành phần:
  - Số lượng món ăn tăng dần khi tổng thành phần tăng từ 1 đến 8, đạt đỉnh tại khoảng 8 thành phần.
  - Sau đó, số lượng món ăn giảm dần khi tổng thành phần vượt qua 8, với rất ít món có tổng thành phần lớn hơn 20.
- Dạng phân phối:
  - Dữ liệu có xu hướng phân phối lệch phải, với nhiều món ăn tập trung ở số lượng thành phần nhỏ (1–15) và rất ít món có số lượng thành phần lớn (trên 20).
- Phổ biến nhất:
  - Tổng thành phần phổ biến nhất rơi vào khoảng 6–8, với hơn 100 món ăn trong các nhóm này.
- Độ hiếm:
  - Các món ăn có trên 15 thành phần là rất hiếm, chỉ chiếm tỷ lệ nhỏ trong tổng số món.

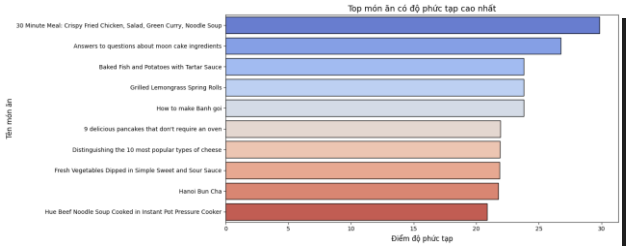
Tiếp theo, ta đánh giá độ phức tạp của một món ăn

Đánh giá theo tiêu chí: số nguyên liệu của món ăn và mức độ khan hiếm của nguyen liệu đó. Sau đó, chia các món ăn thành năm độ phức tạp dựa vào điểm phức tạp đã tính được từ 2 tiêu chí trên.



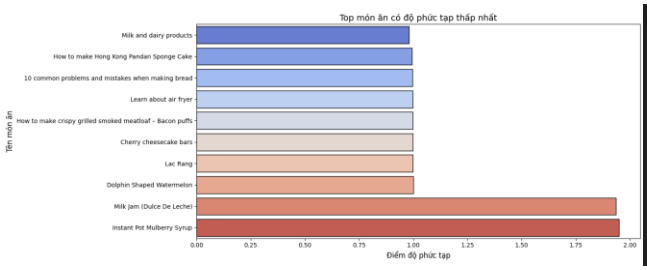
Dựa vào biểu đồ, ta nhận thấy đa số các món ăn có độ phức tạp dễ và trung bình, các món ăn cần nhiều nguyên liệu phức tạp rất ít.

Ta quan sát xem số nguyên liệu cần có ở các món ăn phức tạp, cũng như đơn giản nhất là bao nhiêu.




```
Số lượng nguyên liệu trong mỗi món ăn trên:

1. Món 30 Minute Meal: Crispy Fried Chicken, Salad, Green Curry, Noodle Soup: 60
2. Món Answers to questions about moon cake ingredients: 54
3. Món Baked Fish and Potatoes with Tartar Sauce: 48
4. Món Grilled Lemongrass Spring Rolls: 48
5. Món How to make Banh goi: 48
6. Món 9 delicious pancakes that don't require an oven: 44
7. Món Distinguishing the 10 most popular types of cheese: 44
8. Món Fresh Vegetables Dipped in Simple Sweet and Sour Sauce: 44
9. Món Hanoi Bun Cha: 44
10. Món Hue Beef Noodle Soup Cooked in Instant Pot Pressure Cooker: 42
```

Ta có thể thấy các món ăn có độ phức tạp cao thường là các món ăn có chứa nhiều nguyên liệu (và có thể chứa các nguyên liệu ít phổ biến), thậm chí, món ăn phức tạp nhất có đến 60 nguyên liệu cần thiết, các món còn lại cũng có số nguyên liệu dao động từ 40-50.



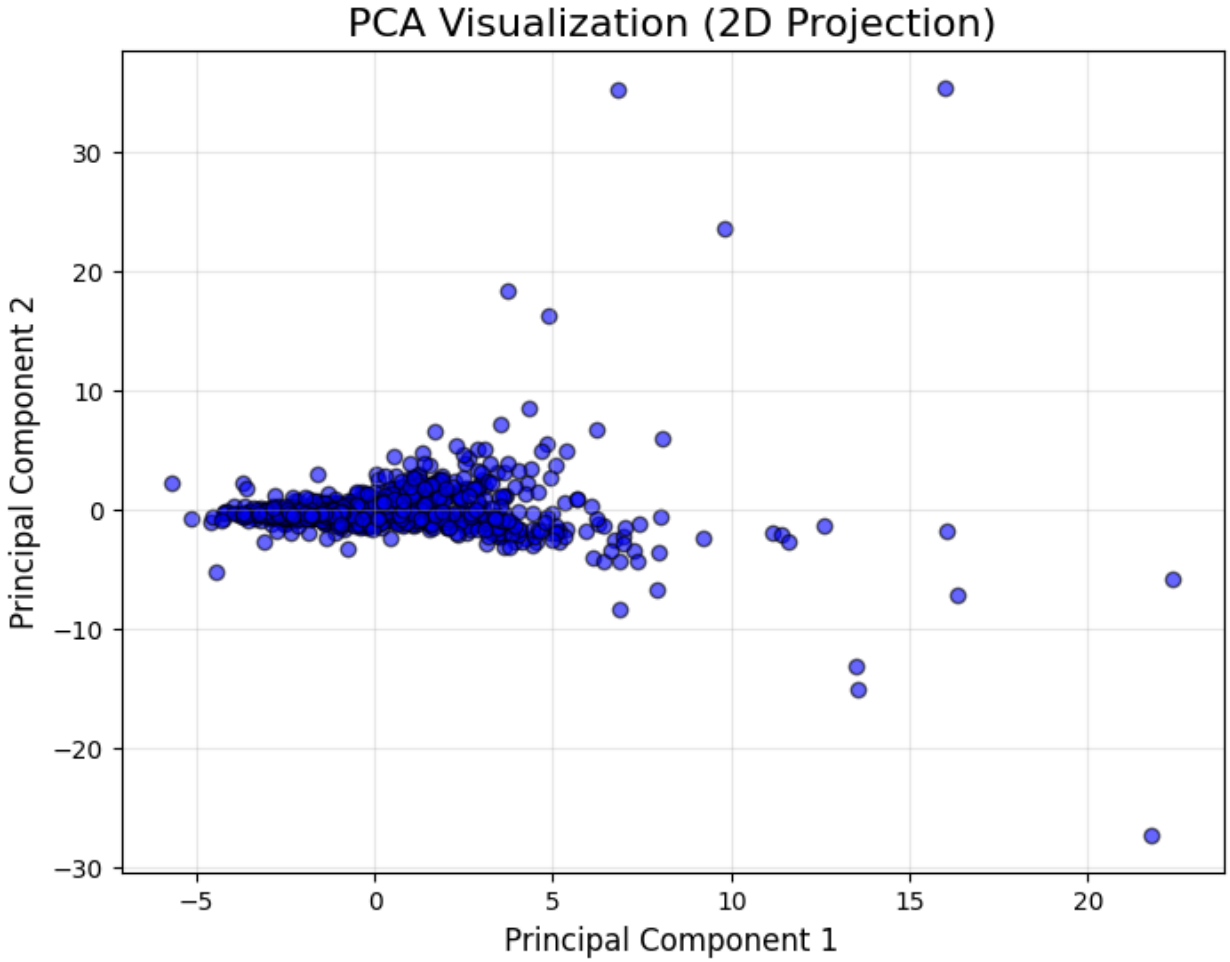

```
Số lượng nguyên liệu trong mỗi món ăn trên:

1. Món Milk and dairy products: 2
2. Món How to make Hong Kong Pandan Sponge Cake: 2
3. Món 10 common problems and mistakes when making bread: 2
4. Món Learn about air fryer: 2
5. Món How to make crispy grilled smoked meatloaf – Bacon puffs: 2
6. Món Cherry cheesecake bars: 2
7. Món Lac Rang: 2
8. Món Dolphin Shaped Watermelon: 2
9. Món Milk Jam (Dulce De Leche): 4
10. Món Instant Pot Mulberry Syrup: 4
```

Ngược lại với các món ăn phức tạp, các món đơn giản chỉ cần 1-2 nguyên liệu là có thể hoàn thành.

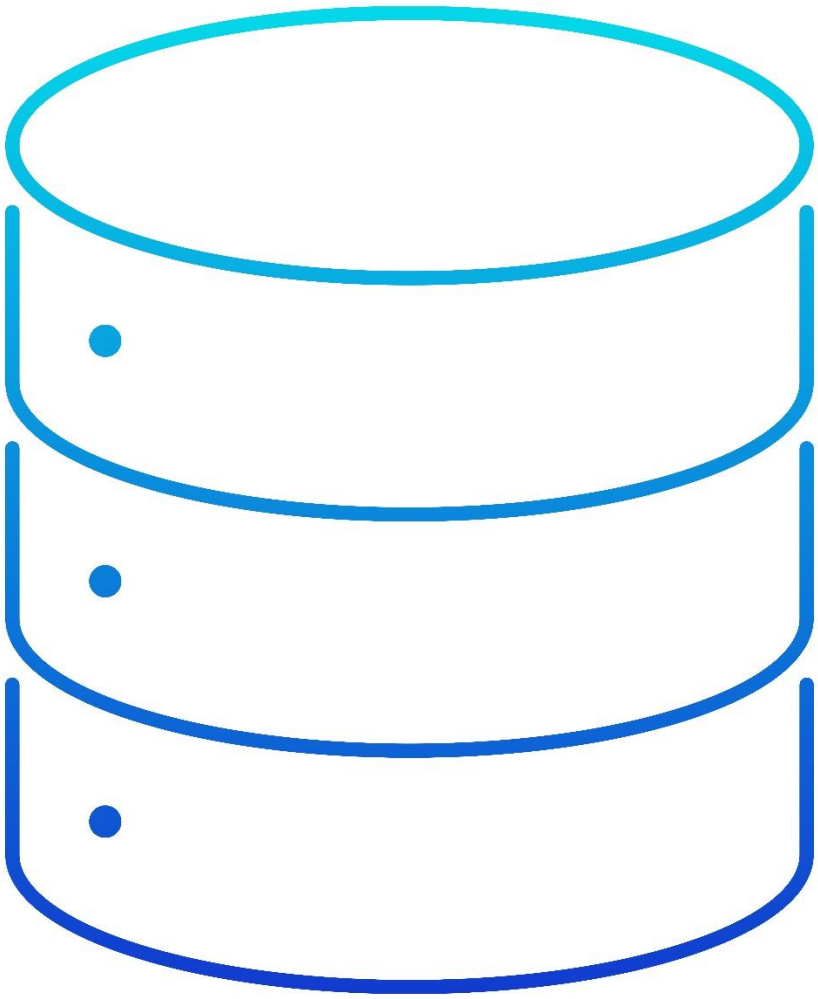
Xác định thành phần chính của dữ liệu

Áp dụng PCA để tìm các thành phần chính của tập dữ liệu

Hình ảnh trực quan (2D):



Phần 4: *Xây dựng và đánh giá mô hình.*



I. Xây dựng tập ý nghĩa cho các nguyên liệu:

Công cụ sử dụng: LMStudio – **Model**: QwQ 2.5 14B Instruct.

Vì ở đây ta sử dụng một model để generate ra các context của nguyên liệu nên có khả năng sẽ sai, cho nên ta sẽ tối đa hóa độ lớn của mô hình để tăng cường độ chính xác cho các ngữ cảnh này (ở đây em sử dụng 14B, 32B là hơi quá sức với 3060 12Gb VRAM và 32Gb RAM) (phần này thì sau khi ta tiến hành đánh giá mô hình và kết luận chung).

Quy trình:

Bước 1: Gởi request lên server LMStudio.

Bước 2: Nhận câu trả lời từ LMStudio.

Bước 3: Phân tách các ý nghĩa ngữ cảnh trong câu trả lời từ LMStudio.

Bước 4: Lưu vào tập context.

Nội dung của request

```
payload = {
    "model": "qwen2.5-14b-instruct",
    "messages": [
        {
            "role": "user",
            "content": (
                f"Tell me the type, context, flavor, smell of {ingredient}. "
                "with knowledge that must be true to reality and must not be fabricated"
                "The result will be in the following format: "
                "\"type : your answer, context : your answer, flavor : your answer, smell : your answer\"; "
                "where type can only belong to one or more of the following factors ['Spice', 'Fruit', 'Vegetable', 'Dairy', 'Meat', 'Grain', 'Condiment', "
                "'Seafood', 'Herb', 'Nut', 'Sweetener', 'Oil', 'Beverage', 'Fermenting Agent', "
                "'Legume', 'Mushroom', 'Pasta', 'Bread', 'Sauce'], "
                "flavor can only belong to one or more of the following factors ['Sweet', 'Salty', 'Spicy', 'Bitter', 'Sour', 'Umami'], "
                "context can only belong to one or more of the following factors ['Binding', 'Thickening', 'Flavoring', 'Sweetening', 'Preserving', 'Topping', "
                "'Fermentation', 'Main Ingredient', 'Garnishing', 'Base', 'Tenderizing', 'Emulsifying', 'Coating', "
                "'Moisturizing', 'Coloring', 'Binding Agent'], "
                "smell can only belong to one or more of the following factors ['Sweet', 'Sour', 'Spicy', 'Bitter', 'Umami', 'Fruity', 'Nutty', 'Smoky', "
                "'Herbal', 'Earthy', 'Fishy', 'Yeasty', 'Citrusy', 'Milky', 'Pungent', 'Floral', 'Fresh', 'Savory', 'Neutral']."
                "No need to add any other information such as notes or cautions."
            )
        }
    ]
}
```

Các yếu tố ngữ cảnh nguyên liệu cần lấy:

- **Type – Loại:** Trái cây, rau củ, thịt, hạt,…
- **Context – Công dụng:** Topping, làm ngọt, gia vị, trang trí,…
- **Favor – Vị:** Ngọt, đắng, chua, cay,…
- **Smell – Mùi:** Mùi thảo mộc, cay, khói, tanh,…

II. Xây dựng hệ thống gợi ý nguyên liệu thay thế cho các nguyên liệu bị thiếu , sao cho phù hợp với các nguyên liệu hiện có:

Sử dụng phương pháp: Collaborative similarity matrix.

Ở đây ta sẽ lần lượt tính toán độ tương tính (cosine_similarity) của các nguyên liệu với nhau cho cả hai tập là :

- Tần suất xuất hiện của các nguyên liệu trong món ăn (mục tiêu là lấy ra các nguyên liệu nào hay đi cùng nhau – theo ý nghĩa là nguyên liệu nào hay đi cùng nhau thì có khả năng kết hợp cùng nhau, thay thế cho nhau)
- Ngữ cảnh của nguyên liệu (mục tiêu là thêm vào ý nghĩa của các món ăn như món cay sẽ có thể thay thế cho món cay, món cá thì có thể thay thế cho món cá,…)

Tiếp theo ta sử dụng kỹ thuật áp dụng trọng số cho cả 2 ma trận này.

Ở đây em sẽ sử dụng độ quan trọng của 2 tập là ngang nhau, tức là 50:50. Để biết được trọng số nào là thích hợp thì ta cần phải sử dụng kỹ thuật vòng lặp và xem xét độ chính xác của mô hình sau bước đánh giá. Ví dụ ở tỉ lệ a:b thì mô hình cho ra đúng hơn ở tỉ lệ a':b'.

Cuối cùng là ta sẽ lấy ra top các nguyên liệu có độ thay thế cao nhất ra.

Kết quả:

```
current_ingredients = ['active yeast', 'agave nectar', 'all-purpose flour']
missing_ingredients = ['brown sugar', 'bacon']

recommendations = recommend_replacements(current_ingredients, missing_ingredients)
recommendations
```

```
{'brown sugar': ['sugar', 'maple syrup', 'icing sugar'],
 'bacon': ['pacetta bacon', 'ham', 'pork loin']}
```

Ý nghĩa:

- Ta có thể thay thế đường nâu bằng đường, bằng maple syrup, hoặc đường bột (đường xay – icing sugar)
- Ta có thể thay thế thịt xông khói (bacon) bằng thịt xong khói pacetta, thịt nguội hoặc thịt lợn loin.

III. Đánh giá mô hình số 01 – thay thế nguyên liệu:

Vì hạn chế về mặt dữ liệu và khả năng thu thập thông tin nên ở đây em sẽ minh họa – lưu ý chỉ là minh họa:

```
replacements_by_category = {
    'all-purpose flour': ['corn flour', 'almond flour', 'coconut flour'],
    'ginger': ['ginger powder', 'cinnamon'],
    'agave nectar': ['honey', 'maple syrup'],
    'brown sugar': ['sugar', 'molasses'],
    'bacon': ['pacetta bacon', 'turkey bacon'],
    'chili': ['ginger', 'paprika']
}
```

Ở đây em sẽ tạo ra một tập các nguyên liệu có thể thay thế được trong thực tế (ý kiến cá nhân) – đối với các mô hình thực tế thì phần này được thu thập từ ý kiến người dùng. (Các nguyên liệu trong list là các nguyên liệu có thể thay thế cao nhất – nếu mô hình dự đoán có 1 trong các thành phần trong list thì coi như mô hình dự đoán chuẩn xác).

```
current_ingredients = ['active yeast', 'almond', 'all-purpose flour']
test_pairs = [
    ('all-purpose flour', 'corn flour'),
    ('ginger', 'ginger powder'),
    ('agave nectar', 'honey'),
    ('brown sugar', 'sugar'),
    ('bacon', 'pacetta bacon'),
    ('agave nectar', 'honey'),
    ('chili', 'ginger')
]
```

Tiếp theo em sẽ định nghĩa các nguyên liệu hiện có **current_ingredients.**

Và một tập các test_pairs tượng trưng cho các nguyên liệu bị mất – pair là để ta đánh giá việc thiếu các nguyên liệu gần nhau thì có kết quả ra đúng hay không.

Ví dụ như lần 1 thiếu ginger, lần 2 thiếu giner power.

Kết quả:

```
Improved Evaluation Results:
Precision@N: 0.4666666666666667
Recall@N: 1.0
F1@N: 0.6363636363636364
```

Precision@N: 0.4667, nghĩa là gần 47% các nguyên liệu được gợi ý là chính xác (phù hợp với kỳ vọng). Điều này phản ánh rằng danh sách gợi ý đã được tinh chỉnh để phù hợp hơn với tập kiểm tra.

Recall@N ở mức tối đa (1.0), nghĩa là tất cả các nguyên liệu kỳ vọng đều nằm trong danh sách gợi ý trả về.

Điều này cho thấy hệ thống vẫn bao phủ toàn bộ các nguyên liệu thay thế mong muốn. Tức là dù không top 1 nhưng vẫn nằm trong top 3.

F1@N: 0.6364, phản ánh sự cân bằng tốt hơn giữa Precision và Recall.

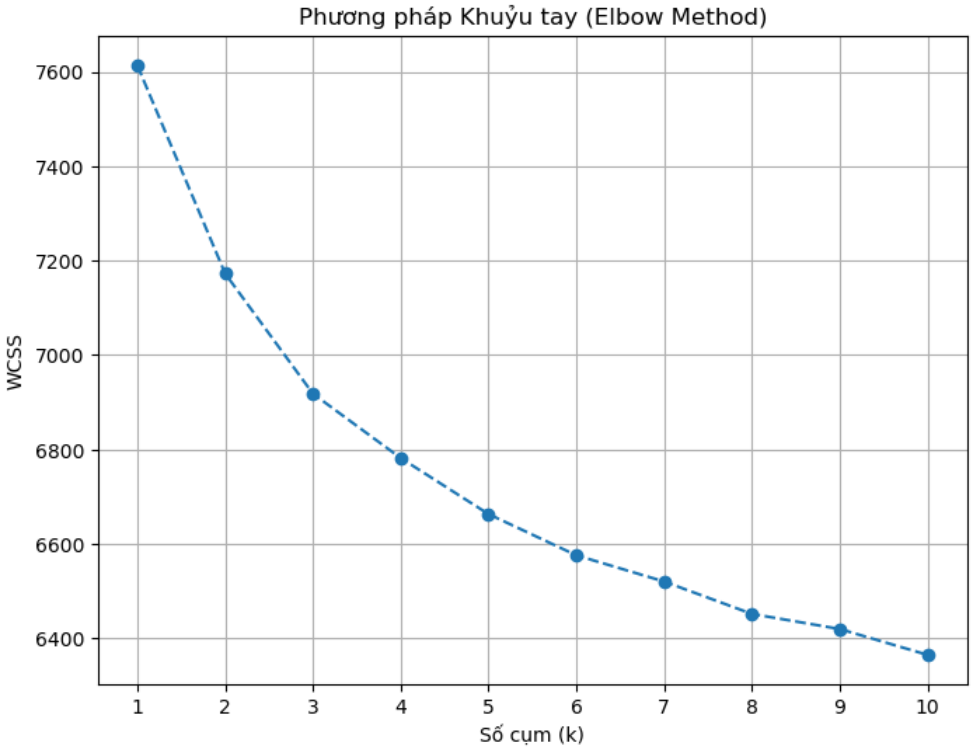
Kết luận: Như vậy thì mô hình dự đoán cũng khá chuẩn chỉ với gần 50% các dự đoán chính xác nguyên liệu có thể thay thế ở vị trí đầu tiên và 100% các nguyên liệu có thể thay thế thực tế đều được đưa ra và nằm trong gợi ý.

Tuy nhiên: Đây chỉ mới là minh họa phần đánh giá, để đánh giá chính xác hoạt động của mô hình thì cần thêm nhiều yếu tố khác. Vì mô hình hiện giờ còn phụ thuộc vào rất nhiều yếu tố như: độ lớn của dữ liệu, ngữ cảnh của nguyên liệu (vì do LLM tạo ra – vẫn chưa thực sự đánh giá độ chính xác), hoặc cần phải được chính người dùng đánh giá và có được số liệu thực tế từ hoạt động gợi ý của mô hình,… Chính vì vậy mà phần tỉ lệ (trọng số) giữa 2 tập vẫn còn lại một dấu hỏi chấm. Em sẽ tạm sử dụng một tỉ lệ là 50:50 nhằm cân bằng độ quan trọng giữa hai mô hình ở hai tập dữ liệu riêng biệt này.

I. Xây dựng hệ thống gợi ý món ăn thay thế (Model 2)

Sử dụng phương pháp: K-Means Clustering

- Đầu tiên, muốn sử dụng phương pháp K-Means Clustering thì phải biết trước số cụm. Do đó ta sẽ sử dụng phương pháp **Khuỷu tay** (Elbow Method) thì xác định số cụm cần thiết.



Nhận xét chung

- Từ k = 1 đến k = 3: Đường cong giảm khá mạnh, cho thấy việc tăng số cụm từ 1 lên 3 giúp giảm đáng kể WCSS.

- Từ k = 3 trở đi: Đường cong tiếp tục giảm nhưng với tốc độ chậm hơn nhiều.

Kết luận

- Dựa vào biểu đồ, có thể cho rằng điểm **khuỷu tay** nằm ở vị trí k = 3.

- Sau khi biết trước số cụm cần thiết là 3, ta sẽ khởi tạo mô hình K-Means.

```
kmeans = KMeans(n_clusters=3, random_state=0)
kmeans.fit(ingredients)

predict_labels = kmeans.predict(ingredients)
```

- Tiếp đến, ta sẽ phân các món ăn cùng cụm vào các nhóm khác nhau.

```
# Lấy các món ăn cùng label vào 1 nhóm
def group_dishes(predict_labels):
    first_group = dishes[predict_labels == 0]
    second_group =  dishes[predict_labels == 1]
    third_group =  dishes[predict_labels == 2]

    return (first_group, second_group, third_group)
```

- Nhóm 1 (5 samples):

	Name of dish
979	Almond Cookies
58	Chocolate Vanilla Butter Cake
199	How to make crispy sesame cookies
322	Rosemary Garlic Challah Bread
1021	Mango Bread

- Nhóm 2 (5 samples):

	Name of dish
599	Dulce De Leche Sauce
655	Super Fast Club Sandwich For Breakfast
123	Garlic and Scallion Butter Bread
1024	Super Fast Strawberry Jam With Fruit Pectin
1003	Milk Jam (Dulce De Leche)

- Nhóm 3 (5 samples):

	Name of dish
795	Fried Avocado with Cosori Air Fryer
891	Pan-Fried Pork Cutlets with Lemon Mustard Sauce
848	Instant Pot Chicken Soup
799	French Fries Using Air Fryer
522	Sauteed Ham – Traditional Northern Dish, Rich in Tet Flavor

Nhận xét

- Ở nhóm đầu tiên, đa phần là các **món bánh**.

- Ở nhóm thứ 2 là các **món ăn vặt**, **chè**, **thức uống tráng miệng.**

- Ở nhóm thứ 3, các món có xu hướng được làm từ **thịt**, **cá**, **tôm**.

Kiểm thử

- Nhập vào danh sách các món ăn.

```
input = ['Muesli','Grilled Orange Chicken','Chocolate Ricotta Muffin']
substitute_recipes = substituteRecipes(input)
printRecipes(substitute_recipes)
```

- Danh sách các món ăn thay thế cho từng món sẽ hiện ra.

```
Substitutive dishes for: Muesli
1. Bo Bo Cha Cha Tea
2. Korean Style Bulgogi Grilled Beef
3. Ginseng Bo Luong Water
4. Cookie and cream cheese truffle chocolate balls
5. 11 ways to use leftover egg yolks

Substitutive dishes for: Grilled Orange Chicken
1. Braised Duck in Coconut Juice
2. Fried Fish Balls
3. Sweet and Sour Pickled Eggs with Natural Pink Color
4. Instant Pot Bolognese Spaghetti
5. Instant Pot Chicken Alfredo Pasta

Substitutive dishes for: Chocolate Ricotta Muffin
1. How to make Pineapple Cookies
2. How to make simple green tea shortbread
3. Lady fingers
4. Delicious carrot cake recipe – Carrot cake nuts with cream cheese icing
5. Monte Carlo Cookies with Cream & Fruit Jam
```

II. Đánh giá, phân tích

- Ta thử nhập vào danh sách nguyên liệu, liệu mô hình có phân cụm đúng không.

```
input_1 = ["shrimp","salt","pork","fish sauce"] # Tôm thịt kho nước mắm :)))
input_2 = ["sugar","all-purpose flour","salt","apple","egg","butter"] # Bánh táo :)))
input_3 = ["milk","aloe vera","sugar"] # Sữa nha đam
input_4 = ["chicken","pepper","carrot","salt","potato"] # Gà hầm rau củ
input_5 = ["sugar","egg","banana","salt","vanilla extract","milk","baking powder"] # Bánh chuối hương vani :)))
input_6 = ["tuna","beef bone","shrimp"] # Cá bò tôm :)))
```

```
[2] - Món mặn
[0] - Món bánh
[1] - Món sữa, chè, thức uống tráng miệng
[2] - Món mặn
[0] - Món bánh
[1] - Món sữa, chè, thức uống tráng miệng
```

Nhận xét

- Ta thấy các món mặn: **Tôm thịt kho nước mắm** và **gà hầm rau củ** chung nhóm. Các món bánh: **Bánh táo** và **bánh chuối hương vani** chung nhóm. **Sữa nha đam** thuộc về nhóm còn lại.

- Tuy nhiên, món **cá bò tôm** lại cùng nhóm với **sữa nha đam** (lẽ ra nó phải thuộc nhóm **mặn**), cho thấy rằng nếu số lượng nguyên liệu ít thì dễ bị phân cụm nhầm sang nhóm **sữa, chè, thức uống tráng miệng.**

- Chung quy lại, nếu số lượng nguyên liệu đủ, mô hình K-Means phân cụm gần như đúng với dự đoán ban đầu.

Đánh giá thuật toán

- Trong thuật toán K-Means Clustering, ta sẽ phân tích các chỉ số như Inertia, Silhouette score và Davies Bouldin score để xem kết quả phân cụm có tốt hay không.

```
Evaluation Score
- Inertia: 6917.189103598309
- Silhouette score: 0.045490248743261 91
- Davies Bouldin score: 4.03154304688805
```

Nhận xét

- Inertia: 6917.19 (k = 3)
- Tổng khoảng cách bình phương từ các điểm dữ liệu đến tâm cụm đã giảm (trên biểu đồ Elbow Method), cho thấy sự cải thiện về độ chặt chẽ của các cụm khi chọn k = 3.

- Silhouette Score: 0.045
- Giá trị này khá thấp, cho thấy các cụm vẫn còn chồng lấn, các điểm dữ liệu không được phân tách rõ ràng.

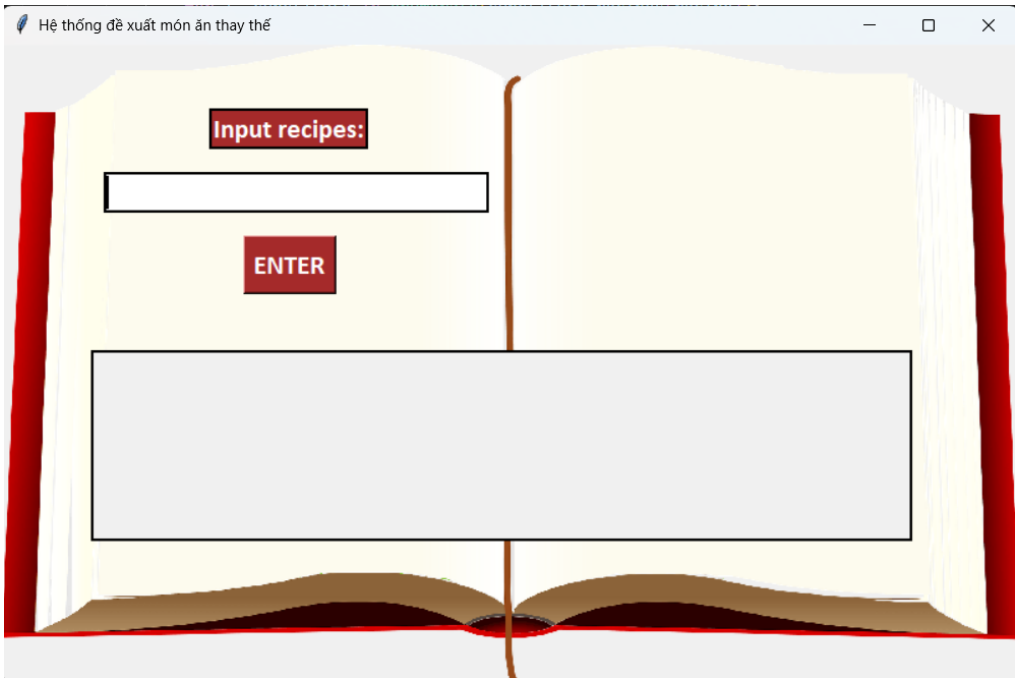
- Davies-Bouldin Score: 4.03
- Chỉ số này tương đối cao, phản ánh các cụm chưa thực sự phân biệt tốt và còn có sự tương đồng giữa các cụm.

Kết luận

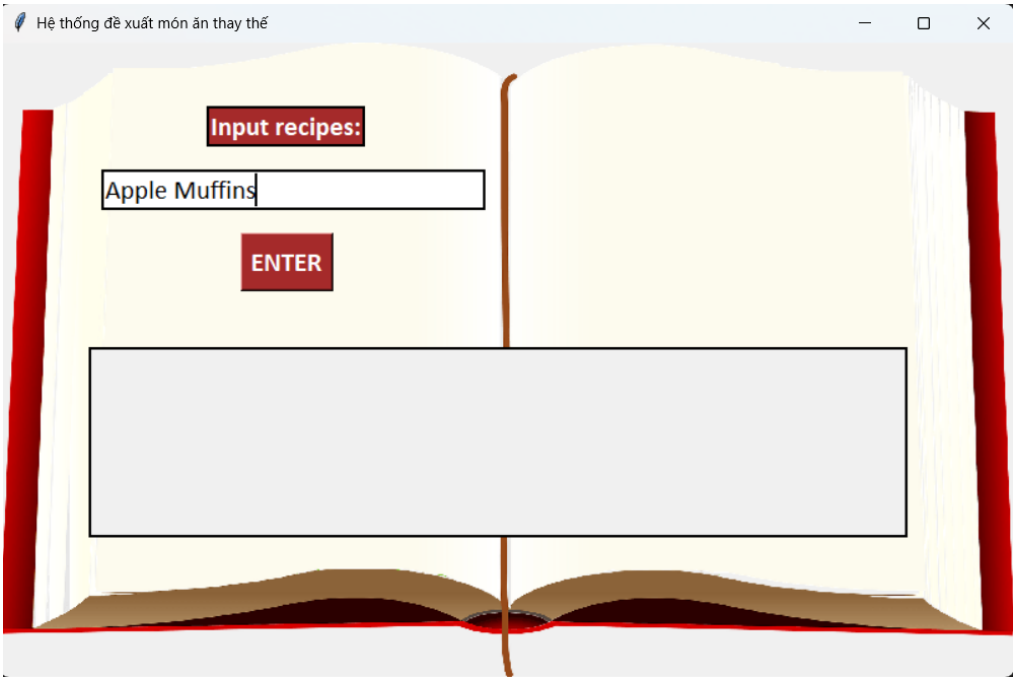
- Số cụm k = 3 là hợp lý theo Elbow Method.
- Tuy nhiên, các chỉ số đánh giá như Silhouette và Davies-Bouldin cho thấy kết quả phân cụm chưa đạt chất lượng cao, các cụm còn khá mờ nhạt, cho thấy cần cải thiện thêm dữ liệu hoặc thử nghiệm các phương pháp phân cụm khác để đạt kết quả tốt hơn.

III. Sử dụng GUI

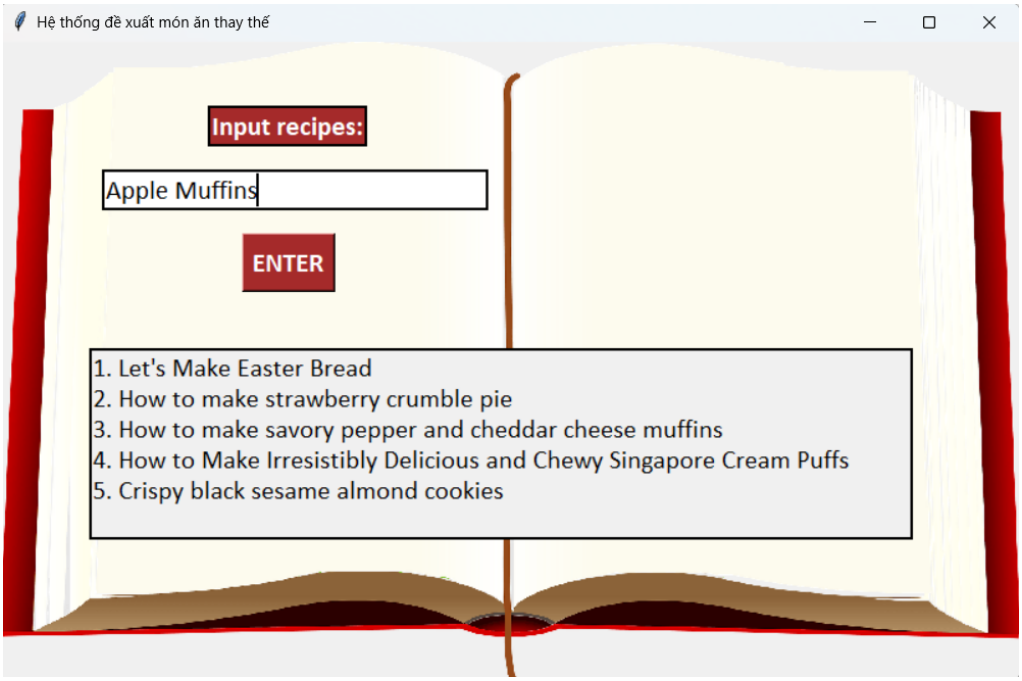
- Sau khi màn hình GUI hiện ra, người dùng nhập tên món ăn mình muốn vào ô entry (lưu ý chỉ nhập 1 món cho mỗi lần search).



- Sau khi nhập tên món xong, nhấn vào ô Enter. Danh sách các món ăn thay thế sẽ hiện ra.



- Kết quả.


Hệ thống đề xuất món ăn thay thế
Input recipes:
Apple Muffins
ENTER
1. Let's Make Easter Bread
2. How to make strawberry crumble pie
3. How to make savory pepper and cheddar cheese muffins
4. How to Make Irresistibly Delicious and Chewy Singapore Cream Puffs
5. Crispy black sesame almond cookies

I. Xây dựng hệ thống gợi ý món ăn từ nguyên liệu cho trước (Model 3)

Sử dụng phương pháp: K-Nearest Neighbors

- Ở đây, ta sẽ tính khoảng cách của dãy nguyên liệu đầu vào và so sánh nó với dãy nguyên liệu của từng món ăn, sau đó xuất ra 5 món ăn có khoảng cách ngắn nhất.

Tiền xử lý input:

```
def convertInputToVector(input):
    ingredients_list = ingredients_df.columns[1:]
    input_df = pd.DataFrame([ingredients_list.isin(input)],columns=ingredients_list)

    return input_df
```

```
input = ['shrimp']
input_df = convertInputToVector(input)
```

- Ở đây ta chỉ có thể cho người dùng nhập vào những nguyên liệu mà họ muốn, nên khi có được danh sách các nguyên liệu đầu vào, ta phải chuyển đổi chúng thành DataFrame (**1** dòng, **tất cả nguyên liệu trong file.csv** cột**),** với giá trị là **True/False** tùy thuộc vào **sự hiện diện của nguyên liệu đó có trong input** hay không.

Lưu ý:

	Name of dish	soda	turmeric mixture	herb	yogurt	butter	beef bone	chili lemongrass fish sauce	beef fillet	chipotle smoked pepper powder	...	kiwi	lime juice	tea leaves	white sesame	eel	green onion	ice cubes
0	10 common problems and mistakes when making bread	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
1	11 ways to use leftover egg yolks	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
2	12 types of nuts for baking	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0

- Do đặc thù các món ăn được lưu dưới dạng vector, nguyên liệu nào có thì bật lên 1, ngược lại thì là 0. Nếu sử dụng khoảng cách **Euclidean** đơn thuần thì sẽ xảy ra 1 vấn đề: Các món ăn có số lượng nguyên liệu đầu vào không bằng nhau (Ví dụ: Tồn tại món ăn chỉ cần 2 nguyên liệu, trong khi có món phải mất đến 30 nguyên liệu) => Sẽ gây ra tình trạng biased do: Nếu ta lấy 2 vector trừ nhau, sau đó tổng bình phương lên, những món ít nguyên liệu thường có khoảng cách ngắn hơn (kể cả không có 1 nguyên liệu nào khớp), dẫn tới đề xuất không chính xác.

Giải pháp: Sử dụng khoảng cách **Jaccard**

Công thức tính khoảng cách Jaccard giữa hai tập hợp A và B được định nghĩa như sau:

$$\text{Jaccard Distance}(A, B) = 1 - \frac{|A \cap B|}{|A \cup B|}$$

Trong đó:

- $|A \cap B|$: Số phần tử chung giữa hai tập A và B.
- $|A \cup B|$: Số phần tử thuộc tập hợp A hoặc B (hợp của A và B).

Giải thích:

- Khoảng cách Jaccard nằm trong khoảng từ 0 đến 1.
- Nếu hai tập hoàn toàn giống nhau, khoảng cách Jaccard là 0.
- Nếu hai tập hoàn toàn không giao nhau, khoảng cách Jaccard là 1.

```
k = 5
knn = KNeighborsClassifier(n_neighbors=k, metric='jaccard')
knn.fit(ingredients, dishes)
```

Thuật toán này khắc phục hoàn toàn những vấn đề trước đó!

- Sau đó, ta sẽ lấy ra những món ăn có khoảng cách ngắn nhất (kiểm tra điều kiện là phải >= 0 và < 1).

```
def dishesRecommenderKNN(ingredients_input: pd.DataFrame):
    distances, indices = knn.kneighbors(ingredients_input)
    filtered_dishes = []

    for i, dist in enumerate(distances[0]):
        if dist < 1:  # Chỉ chọn các món có khoảng cách nhỏ hơn 1
            filtered_dishes.append(dishes.iloc[indices[0][i]])

    return filtered_dishes
```

Kiểm thử

- Nhập vào danh sách các nguyên liệu.

```
input = ['shrimp']
input_df = convertInputToVector(input)

predict = dishesRecommenderKNN(input_df)
for i, recipe in enumerate(predict):
    print(f"{i + 1}. {recipe}")
```

- Danh sách các món ăn đề xuất sẽ hiện ra.

```
1. Shrimp Floss Using Cooking Robot Team Cuisine
2. How to Make Crispy Fried Shrimp and Pork Dumplings Using an Air Fryer
3. Grilled Shrimp Skewers
4. Coconut Fried Shrimp With Cosori Air Fryer
5. Radish Soup Rolled with Shrimp and Pork
```

II. Đánh giá, phân tích

- Vì mô hình KNN làm việc với dữ liệu đầu vào có các **nhãn khác nhau đôi một** nên **không thể áp dụng** các chỉ số **Precision**, **Accuracy** và **Recall** như các thuật toán **Học có giám sát thông thường**.

- Thay vào đó, ta sẽ sử dụng **Mean Pairwise Jaccard Similarity** để đánh giá mức độ tương đồng giữa các mẫu trong bộ dữ liệu đầu vào.

```
Mean Pairwise Jaccard Similarity: 0.07122587428134408
```

Ý nghĩa

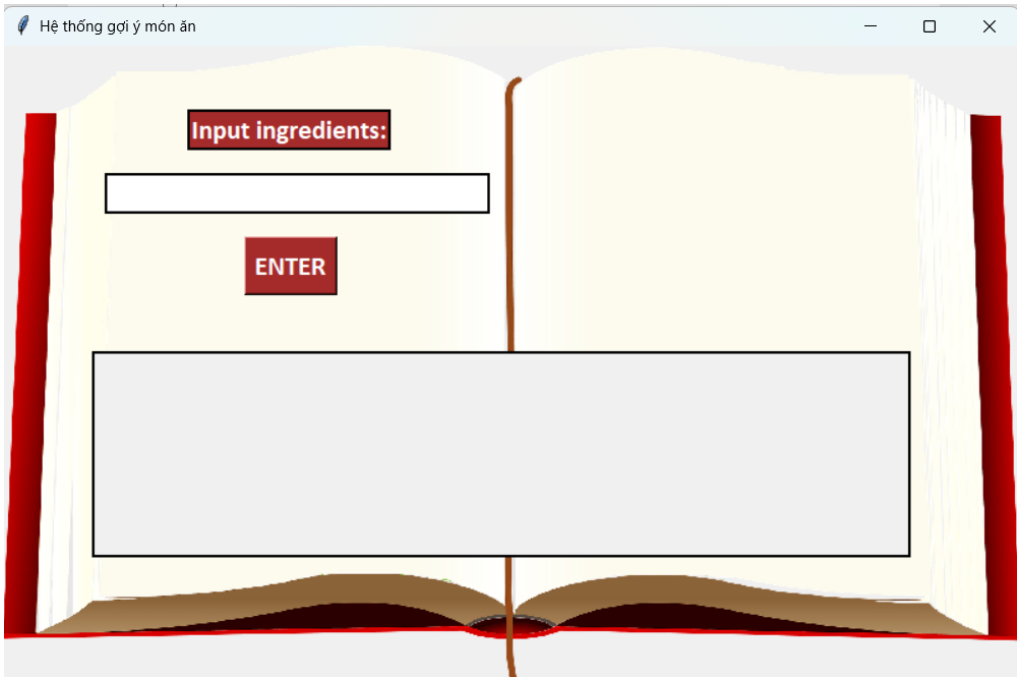
- Jaccard Similarity đo lường tỷ lệ các phần tử chung giữa hai tập dữ liệu so với tổng số phần tử trong hai tập đó.
- Giá trị Jaccard nằm trong khoảng [0, 1]:
    + Gần 1: Tính tương đồng cao giữa các mẫu.
    + Gần 0: Tính tương đồng thấp giữa các mẫu.
- Trong trường hợp này, giá trị 0.071 cho thấy **mức tương đồng rất thấp** giữa các mẫu trong dataset.

Nhận xét chi tiết

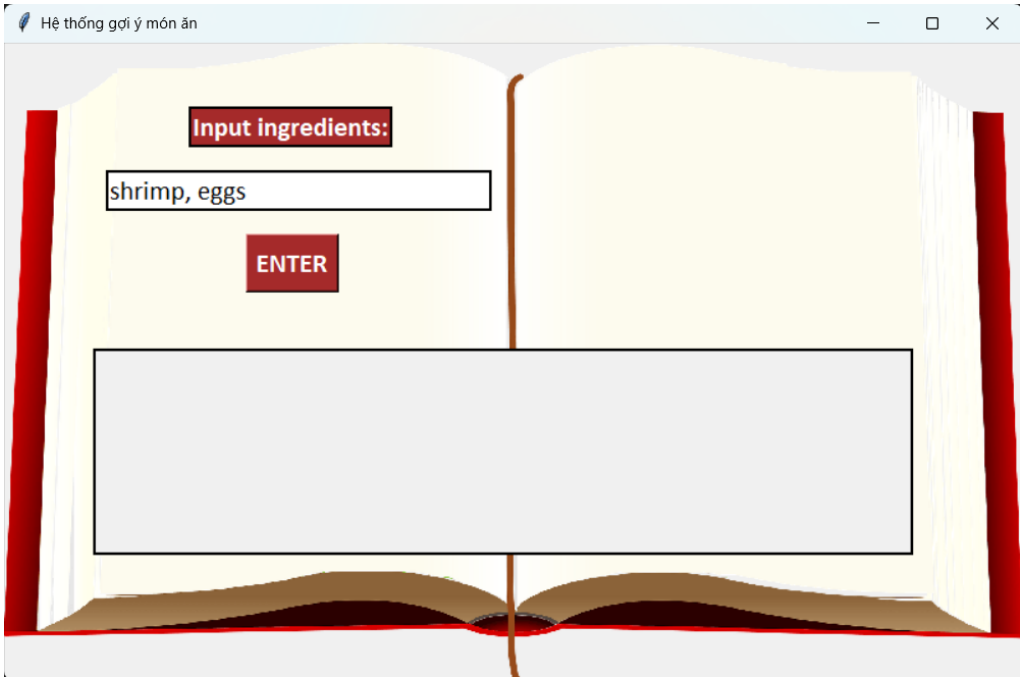
- Giá trị 0.071 chỉ ra rằng các mẫu dữ liệu trong dataset **ít có sự trùng lặp** về các thành phần nguyên liệu.
- Trong một số trường hợp, điều này có thể phản ánh tính **đa dạng** của dataset, giúp mô hình học được sự phong phú của các kết hợp nguyên liệu khác nhau.
- Tuy nhiên, mức tương đồng thấp này khiến mô hình KNN **khó tìm được** các mẫu gần nhất có tính tương đồng đáng kể. Điều này ảnh hưởng đến khả năng dự đoán các món ăn tương tự vì mô hình không thể nhận ra các mối liên kết hợp lý giữa các thành phần nguyên liệu.

III. Sử dụng GUI

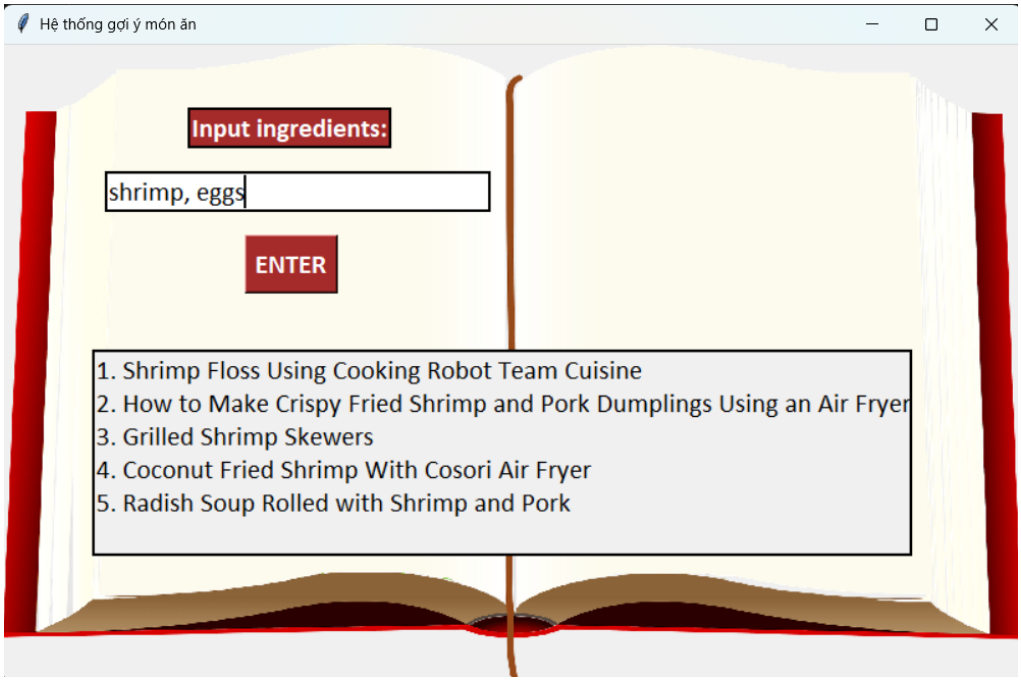
- Sau khi màn hình GUI hiện ra, người dùng nhập nguyên liệu mình muốn vào ô entry (các nguyên liệu phân tách bởi dấu phẩy).



- Sau khi nhập nguyên liệu xong, nhấn vào ô Enter. Danh sách các món ăn đề xuất sẽ hiện ra.



- Kết quả.


Hệ thống gợi ý món ăn
Input ingredients:
shrimp, eggs
ENTER
1. Shrimp Floss Using Cooking Robot Team Cuisine
2. How to Make Crispy Fried Shrimp and Pork Dumplings Using an Air Fryer
3. Grilled Shrimp Skewers
4. Coconut Fried Shrimp With Cosori Air Fryer
5. Radish Soup Rolled with Shrimp and Pork