CS6200 Information Retrieval Final Project Report

Beauty-Search: A Search Engine for Beauty Products

Yuting Sun | 001380847 | sun.yut@northeastern.edu

Github: https://github.com/3unyt/Beauty-Search

Abstract

Ingredients determine the properties of beauty products, thus can be useful for product recommendations. However, ingredients are rarely taken into account in current recommendation systems due to the difficulty in ingredient analysis. This project develops a search engine for beauty products that has two search modes: the basic search mode and ingredient search mode. The basic search mode returns relevant products based on user input queries. The ingredient search mode recommends products with similar ingredients. With the two search modes combined, this project expands the dimensions of product search results and provides a new perspective for beauty product recommendation systems.

Introduction

It is always a challenging task to try out a new cosmetic product. Different people have different skin types and product preference. A product that is good for one person may not be suitable for another. Improper products would make customers waste a lot of money and even suffer from severe skin problems. Since the COVID-19 pandemic, it's not convenient for people to go to the stores and try the products on themselves. Therefore, it would be helpful to build a searching tool for beauty products.

Ingredients are key to the property and effectiveness of a beauty product. For example, salicylic acid is widely used in treating acne, titanium dioxide (TiO_2) is used for UV absorbance in sunscreen products. Ingredient analysis can be very helpful for people to search for proper products. Some people like the certain property of a product, so they would probably like a product with similar ingredients. Some people are allergic to a certain ingredient, so it would be necessary to avoid products with the same ingredient.

However, ingredients are rarely taken into consideration in current recommendation systems for beauty products. The major reason could be the fact that most cosmetic products contain more than 30 ingredients, which is difficult for people to read and analyze. It would be a good idea to combine information retrieval techniques with ingredient analysis. This would help customers find out new products that are most suitable for them, as well as getting more insights on cosmetic ingredients.

Proposed Solution

This project develops a search engine for beauty products that contains two searching modes:

Basic search

The search engine conducts basic search based on queries input by the user. It returns top-10 relevant products that best match the query in fields of brand, name, and product descriptions.

Ingredient search

For each product returned in basic search, the search engine then extracts the ingredients of the product, and searches for other products that have similar ingredients. It returns top-5 similar products for each basic search result.

With the combination of the two search modes, this project would provide more perspectives of product search results and help customers find new suitable beauty products.

Dataset

The dataset used in this project was downloaded from Kaggle[1]. The Sephora dataset contains 9000+ products with product information, detailed description and ratings. 7538 (82%) products have ingredient information.

[1] https://www.kaggle.com/raghadalharbi/all-products-available-on-sephora-website

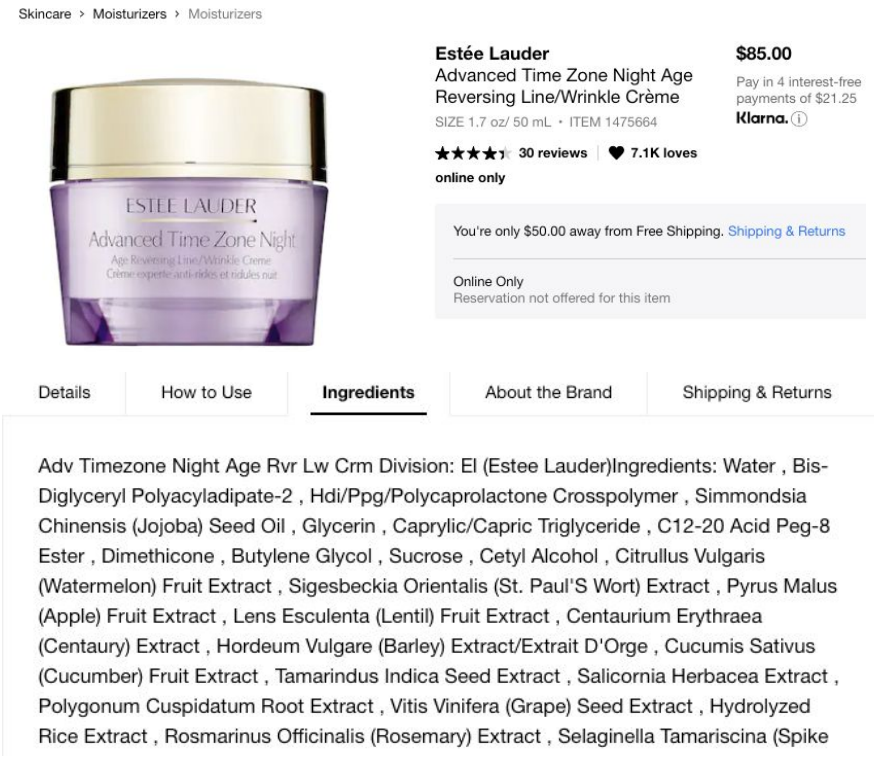

Figure 1. A typical product page with ingredients from Sephora website.

Implementation

This project was implemented with Python 3. The dataset was loaded and processed with Pandas library. The web framework and user interface for search input and result output were implemented with Flask[2]. The indexing and searching API was adapted from Elasticsearch[3].

Indexing

The data was stored in a .csv file and was loaded by pandas. The columns were selected to contain only `id`, `brand`, `category`, `name`, `price`, `URL`, `details`, `how_to_use`, and `ingredients`. To guarantee that every search result has ingredient

[2] https://flask.palletsprojects.com/en/1.1.x/

[3] https://www.elastic.co/

information, the products with unknown ingredients, which makes 18% of the entries, were removed from the dataframe.

The product list for indexing contains 7538 entries, each column was indexed with the field name same as the column name. The indexing process takes approximately one minute.

Basic Search

Basic search takes the query as user input in the search box, and returns a list of related products. The query is taken to Elasticsearch Search API and searched among the selected fields with boosted weights:

```
[ "brand^2", "name^2", "category^2", "details", "how_to_use"]
```

According to the sample queries obtained from class presentation, most users tend to input query with the brand (eg. Estee Lauder), and with a general description of the type of product (eg. lipstick, sunscreen). Since the query terms have high probability to appear in the fields of `brand`, `name` and `category`, these fields have two times weight in scoring.

The basic search matches the fields above as a `multi-match` query with the type of `cross_fields`. The `cross_fields` query type first analyzes the query string into individual terms, then looks for each term in any of the fields, as though they were one big field.

The first advantage of the `cross_fields` type comes from user input habits: users often input a mixture of brand, name or description. For example, in the query “clinique moisturizer”, the term “clinique” is the brand, and “moisturizer” can appear in the product name, category, and/or description. The second advantage is that in `cross_fields` type, all terms must be present in at least one field for a document to match. This prevents the return of irrelevant products that matches one query term frequently but does not contain other query terms.

Take the query “moisturizing lipsticks” as an example (Table 1). Using the default query type `best_fields`, only the first result returns a lipstick product. This is due to the fact that the term “moisturizing” appears less common in product names than “lipstick”, therefore is of greater importance in searching algorithms. The products with “moisturizing” in their name would receive a higher score than that with “lipstick”,

therefore 4 out of the top-5 results contain only moisturizing but without “lipstick”. On the other hand, with the query type of `cross_fields`, all terms must be present in at least one field for a document to match. As a result, all top-5 results are lipstick products.

Table 1. Top-5 search results for “moisturizing lipsticks” with different query types

Query Type	Top-5 Search Results

`best_fields` (default)	`Ciaté London | Liquid Velvet-Moisturizing Matte Liquid Lipstick`
	`Caudalie | Moisturizing Toner`
	`Caudalie | Moisturizing Mask`
	`CLINIQUE | Moisturizing Lotion`
	`Sachajuan | Moisturizing Conditioner`

`cross_fields`	`Ciaté London | Liquid Velvet-Moisturizing Matte Liquid Lipstick`
	`TOM FORD | Lip Blush`
	`bareMinerals | GEN NUDE™ Radiant Lipstick`
	`Laura Mercier | Rouge Essentiel Silky Crème Lipstick`
	`Charlotte Tilbury | K.I.S.S.I.N.G Lipstick`

Therefore, in the basic search mode of this project, the query type `cross_fields` that combines the selected fields into one big field would be a better choice than the most commonly used query type of `best_fields`.

Ingredient Search

The ingredient search takes the query of the ingredients of a product returned from basic search, and returns a list of products that have similar ingredients. The ingredients

of a certain product can be obtained with the method `es.get(id=product_id)`. The key ingredients were extracted from the original ingredient text and fed into the search. The search is conducted with the `match` query in the field of `ingredients`, and returns a list of products that have top similarities with the query ingredients.

For similarity product results, Elasticsearch Highlighters[4] were utilized to get snippets from the `ingredients` field with highlighted ingredients that match the query ingredients.

Table 2. Similar products with "CLINIQUE|Turnaround Overnight Revitalizing Moisturizer"

Exclude Same Brand	Similar Products
NO	CLINIQUE | Repairwear Laser Focus Night Line Smoothing Cream for Very Dry to Dry Combination Skin CLINIQUE | Repairwear Laser Focus Night Line Smoothing Cream for Combination Oily to Oily Skin CLINIQUE | Clinique For Men™ Anti-Age Moisturizer CLINIQUE | Superdefense SPF 25 Fatigue + 1st Signs of Age Multi-Correcting Cream
YES	CLINIQUE | Repairwear Laser Focus Night Line Smoothing Cream for Very Dry to Dry Combination Skin CLINIQUE | Repairwear Laser Focus Night Line Smoothing Cream for Combination Oily to Oily Skin Estée Lauder | Advanced Time Zone SPF 15- Normal/Combination Skin TOM FORD | Oil-Free Daily Moisturizer

[4] https://www.elastic.co/guide/en/elasticsearch/reference/current/highlighting.html

To provide more diversity for the product recommendations, the ingredients searcher is also designed with a brand checking method. First, it conducts the default ingredient search and returns a list of similar products. Next, it checks the number of products that share the same brand with the query brand. If more than 3 products are from the same brand, it conducts a second ingredient search that excludes the same brand. The second search is implemented with a boolean query that excludes results with the query_brand. As a result, at least two of the returned products come from different brands. This method would prevent customers stucking in the feedback loop of the same brand.

For example, in searching similar products with the product “CLINIQUE | Turnaround Overnight Revitalizing Moisturizer”, if no brand checking methods were applied, all returned results would be from the same brand. However, with the brand checking method, the last two products are from different brands. The detailed results are listed in Table 2.

User Interface

The user interface is created with Html and CSS with python flask framework. The user interface template is adapted and simplified from an open source search engine template[5]. The main page (Figure 2.) contains a search box that prompts users to input queries and a search button to start searching.

The result page (Figure 3.) contains a header with a search again button, a title containing the query text, and a list of result output. Result output for basic search contains the product brand, name, URL, and a short description followed by a list of selected ingredients. Below each basic search result, a list of similar products is presented with product brand and name, embedded with the product URL. A snippet of the ingredients is listed below each similar product, with the matched ingredients highlighted.

[5] https://github.com/phpSoftware/search-engine-template

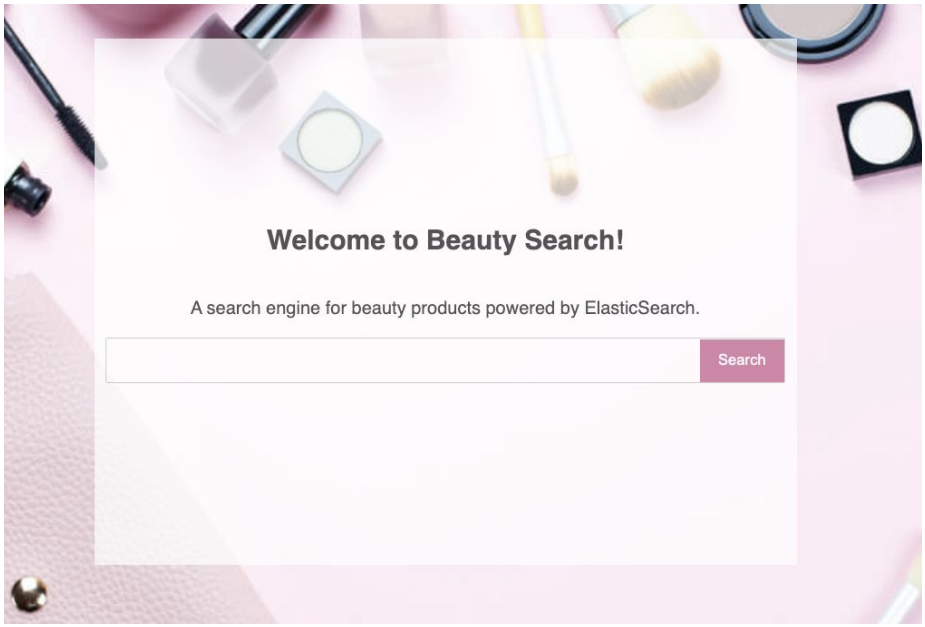

Figure 2. The main page of Beauty-Search

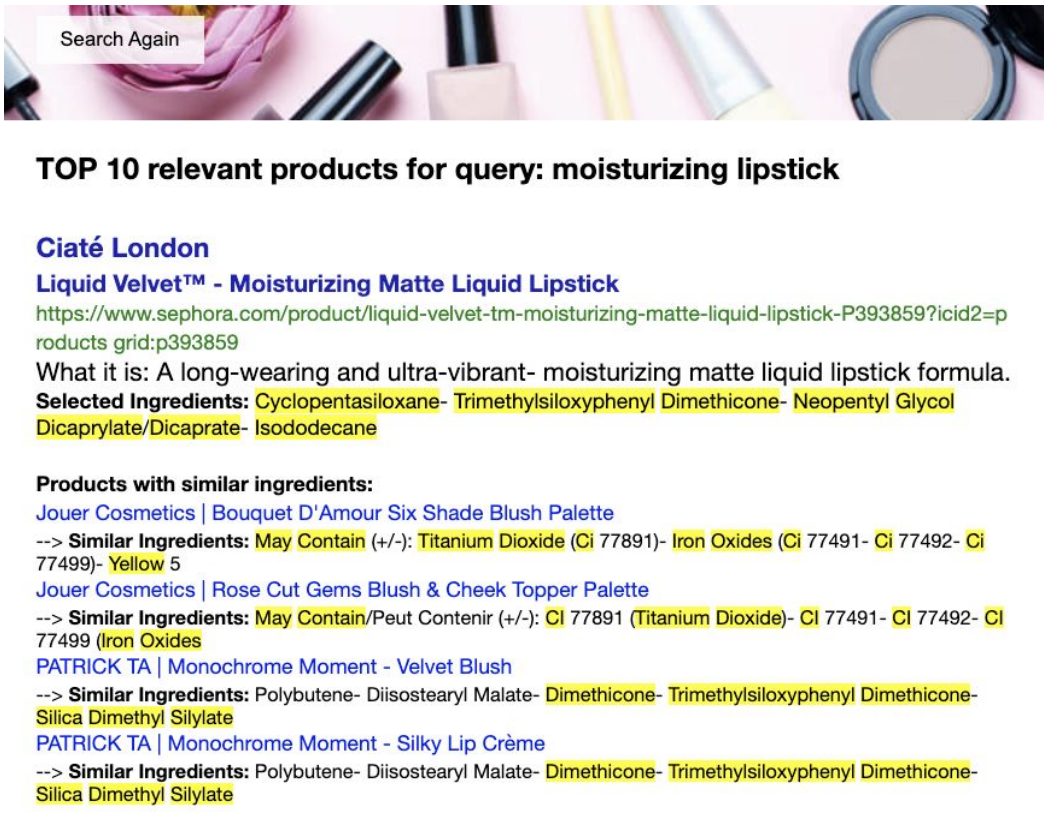

Figure 3. The layout of a search result page.

Summary

This project develops a search engine for beauty products that contains two search modes: i) basic search for query-related products and ii) ingredient search for similar products. The basic search mode is optimized with query type cross_fields that combines the selected fields into one big field. The ingredient search mode is enhanced with the ingredient highlighter and a brand checking method. The ingredient highlighter highlights the similar ingredients matched with the query ingredients. The brand checking method provides more diversity to the search results by returning similar products from different brands.

With the combination of two search modes, this project expands the dimensions of product search results and provides a new perspective for beauty product recommendation systems.