Alex Kim
Justin Lee
Shayal Singh
DSC 180B - A05

Makeup Recommender Report

I. Abstract

Although product recommenders are conventional in the world of machine learning based recommender systems, cosmetics are an overlooked field. By providing a complete set of cosmetic recommendations, we can reduce the time and effort required for users to find the best products for a user's personalized needs. Our goal is to create a recommender that will provide a one-stop shop experience where a user will get recommended an array of products to create an entire makeup look based on similar products that the user enjoys, products that similar users have purchased, as well as products that are personalized to the user including skin type, skin tone, ingredient preferences, and budget. The website recommends a complete makeup set personalized to the user. The user inputs their skin type, skin tone, budget, and any ingredient preferences so that we can suggest the best products for their personalized needs. The user also inputs a product of their choice from one of the four categories to aid with further personalization. Using this preference and knowledge about the user, we will suggest a complete set of products to complete a look. Our recommender provides four categories of products: face, cheeks, eyes, and lips. Our project utilizes collaborative filtering recommendations to ensure user satisfaction and success when creating their desired look.

II. Description of Data

The dataset is scraped from Sephora.com using Selenium to acquire products and review information from the four categories of face, cheek, eyes, and lips. Initial products were scraped on 1/4/21 and all reviews were scraped by 1/24/21. We collected up to 300 reviews per product, sorted by most helpful. Our dataset includes explicit feedback on each product in the form of numbers of stars received. Steps of data pre-processing include encoding the ingredients.

The features of the product dataframe are as follows:

Feature	**Type**	**Description**
Label	str	The product's category: face, cheek, eye, or lip
URL	str	URL of the product
productID	int	Unique product ID for product Note: Products with multiple colors or sizes have unique product IDs for each variance. Products are

		treated as having one color/size in our model.
brand	str	Brand of the product
name	str	Name of the product
price	str	Price of the product
description	str	‘Details’ tab of product that outlines: - What the product is - Coverage (light, medium, or full) - Finish - Formulation - Skin Type (normal, dry, combination, or oily) - Ingredient Callouts - Anything else you need to know
ingredients_rubber	boolean	Whether or not the ingredients contain common rubber allergens
ingredients_preservatives	boolean	Whether or not the ingredients contain common preservative allergens
ingredients_fragrances	boolean	Whether or not the ingredients contain common fragrance allergens
ingredients_metals	boolean	Whether or not the ingredients contain common metal allergens
subCategory	str	Subcategories of the main four categories include: - Face: foundation, concealer, powder - Cheek: contour powder, bronzer, highlight, blush - Eye: eyebrow, mascara, eyeshadow, eyeliner, false eyelashes - Lip: lipstick, lip liner, lip gloss
total_reviews	int	Total reviews for the product

Table 1: Description of product dataframe

The features of the reviews dataframe are as follows:

Feature	**Type**	**Description**
userID	str	ID of user who reviewed
productID	str	Unique product ID for product

rating	integer	Number of stars given in review (1-5)
skin_tone	str	Skin tone of user who reviewed: - Porcelain, fair, light, medium, olive, tan, deep, dark
skin_type	str	Skin type of user who reviewed: - Normal, dry, combination, oily

Table 2: Description of review dataframe

Post processing, various statistics can be seen below:

Category	Number of Users	Number of Products	Number of Reviews
Face	91518	522	114590
Cheek	27642	217	31819
Eyes	120065	836	153777
Lips	59043	537	115011

Table 3: Various statistics relating to the four categories of products

III. Method

A. Baseline Comparison Method

To ensure that our model is working properly and effectively, we must compare it to a baseline method. We chose Top Popular as this baseline, so that users have the choice to see recommendations from this model as well. The Top Popular model returns the top most popular products for each category of makeup. Popularity is determined by the number of reviews a product has received.

B. Recommender model

1. Collaborative filtering

Since our data contains users and product reviews, we chose to include collaborative filtering in our model. Collaborative filtering uses similarities of users and items to recommend products. Our model recommends items based on the assumption that people enjoy products that are enjoyed by other users with similar taste. Based on a user's previous reviews, user-based collaborative filtering calculates the similarity between the target user and all of the other users. Two users are deemed similar if they rate items similarly. The model weights the ratings from the similar users and predicts the rating of the new items for the target user.

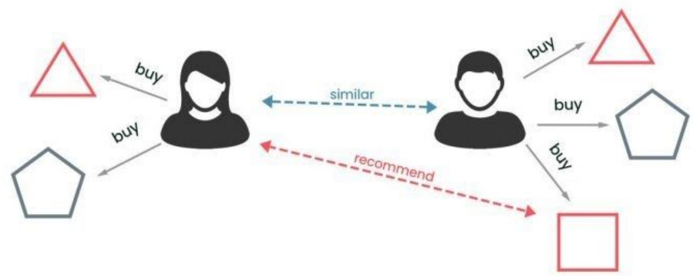

Figure 1: Diagram of collaborative filtering approach

It should be noted that collaborative filtering has a cold start problem. If an item has not been reviewed by enough users, the model cannot use it in its recommendations. Our model uses up to 300 reviews per product, though ideally we would gather all of the reviews for each product.

2. Additional Filtering

Once we have received the initial recommendations from the collaborative filtering model, we must further personalize the recommendations according to the user's preferences. The final step in our recommendations is to perform basic filtering on the following:

- Budget: ensure that the total cost of the products recommended are less than or equal to the budget selected
- Ingredient preferences: ensure that the products recommended do not contain ingredients from the ingredient group(s) (rubber, preservatives, fragrances, metals) selected

C. Website

Our website was built using the Python package, Streamlit. Stylistic touches were added with CSS. It includes a section for the user to enter their skin information along with their preferences such as ingredients and budget. To deploy our functional website publicly, we utilized Heroku to generate the following domain for our website: https://makeup-recommender.herokuapp.com/.

IV. Metrics

To measure the effectiveness of our recommendations, we calculated the AUC, or area under the curve. The AUC is an estimate of the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance. It measures the model's skill in ranking recommendations, which is a key idea for our model. The higher the AUC the better the model is at distinguishing between the positive and negative classes. The scores can range from 0 to 1, with 1 being a perfect classifier. A score of 0.5 suggests no discrimination between positive and negative instances.

V. Results

A. AUC Score

The current AUC for our model lies at 0.68, with a train-test split of 0.9, 0.1 respectively. This is a decent score representing an acceptable job of predicting items in each category that the user would enjoy. We noticed that when we add more data, our AUC increments, so we believe that one aspect of improving our AUC is a matter of obtaining more data.

B. User Studies

The results of our user studies allowed us to tailor our product and website to be more useful and user friendly. After showing the website prototype to our peers, family, and friends we gained helpful feedback about features that would enhance the user experience of our site. These improvements included a helpful guide on how to determine one's skin type, images to aid in choosing a skin tone, and a depiction of how to find product IDs on Sephora's website. Taking these suggestions into consideration, we made modifications to the website's functionality, features, and stylistic properties.

VI. Conclusion

The beauty industry is in need of recommenders that not only work for a diverse range of individuals, but also ones that can recommend an array of products so that the user can effortlessly create a complete look. Our recommender provides a one-stop shop for users to get recommended an entire set of makeup products needed to create a complete makeup aesthetic. These products not only align with the user's physical characteristics but also with the type of makeup look that the user desires. The model uses collaborative filtering to make recommendations on similar users who have purchased the same products. These products are then filtered even further to cater to user characteristics and preferences such as allergy and budget.

A. Limitations

A limitation to our approach is the amount of data we have in our dataset. Ideally, we would gather all the reviews for every product in our dataset; however, this was not a feasible task due to lack of time. Our web scraper is slow, so we are not detected as bots and are blocked from using the website. Therefore, we had to limit the number of reviews to up to 300 reviews per product. We were also limited by structural changes in the website while we were scraping data. We had to make several changes to our web scraper further limiting our ability to obtain more data.

The cold start problem with collaborative filtering models addresses another limitation to our approach. Since products with more reviews tend to get weighted more and products with minimal reviews get weighted less, there is a slight popularity bias towards products that have many reviews. Newer products will get recommended less because they have less reviews. However, this does not mean that newer products are bad and should not be recommended, but because of the nature of our recommender, there is a bias towards more established products and thus have more reviews. It is likely that products with no reviews at all will not be recommended in our model.

B. Future Work

Future work includes improving upon and fine tuning our collaborative filtering model to yield a better accuracy. By including more products and reviews, we can improve the results of our model. Additionally, we would like to polish our website further by including images for the recommended products. This would help with the visual appeal of our recommendations. Lastly, we would like to scrape products in real time to include the most up to date products in our model. This would ensure that new products to Sephora are included in our recommendations, and that discontinued products are excluded.

VII. References

Original Sephora scraper can be found here: https://github.com/jjone36/Cosmetic

Image Sources:
[Figure 1] https://useinsider.com/top-recommendation-engines/

VIII. Appendix

Website GitHub Repo: https://github.com/alexxkim54/AMR_Website
Project GitHub Repo: https://github.com/alexxkim54/Aesthetic_Makeup_Recommender

Project proposal:
https://docs.google.com/document/d/1bAXSUrQHcss8uU_eeqIJ4ewX3N-I8RLAjGJi77Zqw6c/edit