Skin Care Recommendation System

T2-23-24-AI 705 / Recommendation Systems Project

Aakash Khot
dept. ECE
IIIT Bangalore
IMT2020512
Aakash.Khot@iiitb.ac.in

Agastya Thoppur
dept. ECE
IIIT Bangalore
IMT2020528
Agastya.Thoppur@iiitb.ac.in

Anwit Damale
dept. ECE
IIIT Bangalore
IMT2020532
Anwit.Damale@iiitb.ac.in

***Abstract*—In the realm of skincare, selecting suitable products amidst the abundance available poses a challenge for consumers. To alleviate this, we propose a personalized skincare recommendation system. By integrating user ratings, ingredient analysis, brand reputation, and reviews, our system delivers tailored product suggestions. Employing collaborative filtering, ingredient compatibility assessment, and sentiment analysis, our approach ensures recommendations align with individual preferences and skincare goals. Through experimentation, we validate the efficacy of our system in simplifying product selection and enhancing user satisfaction.**

***Index Terms*—skincare recommendation, personalized system, ingredient analysis, collaborative filtering, sentiment analysis.**

I. Data Collection

Our dataset, sourced from Kaggle, originates from a meticulous web scraping effort on Sephora's official website (sephora.com). Sephora, a global beauty retailer renowned for its diverse skincare selection and expert beauty guidance, provides the backdrop for over 10,000 reviews included in the dataset. Each review encompasses 26 features, including product ingredients, brand details, category classifications, user ratings, and textual reviews. This comprehensive dataset forms the basis for our exploration into personalized skincare recommendations, aiming to distill actionable insights to aid users in making informed skincare decisions.

II. EDA

Our exploratory analysis delved into key facets of consumer behavior and product perceptions within the skincare domain.

Think about the case when you are a person who establishes an online platform for providing skincare advice. Through the analysis of the distribution of ratings for different skincare products, you will be able to find out which products are the most liked by users and which ones are not. This data can act as a basis for your recommendation engine to suggest top-rated products which will in turn increase the user satisfaction rate.

Likewise, realizing the range of skin tones, skin types, hair colors, and eye colors among your users can be useful when they give recommendations that are more in line with the specific needs of a person. An example is, if you find out that a big group of your users has sensitive skin you can opt to recommend products that are mild and proper for sensitive skin types.

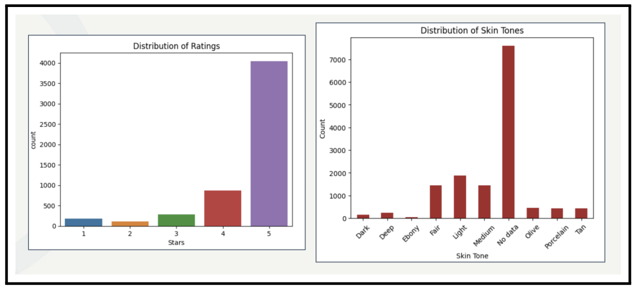


Fig. 1

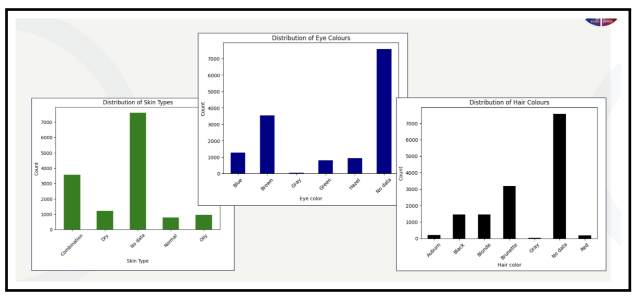


Fig. 2

- **Rating Distribution:** We examined the distribution of user ratings across skincare products, providing insights into overall satisfaction levels and identifying noteworthy trends.
- **Skin Tone and Skin Type Distribution:** Analysis of skin tone and type distributions among reviewers revealed valuable demographic insights, aiding in the creation of tailored recommendations.
- **Word Cloud Analysis:** Utilizing text mining, we generated word clouds to visualize prevalent terms in both positive and negative reviews. Notably, complaints about product scent emerged prominently among dissatisfied consumers. As you can see, negative based word cloud

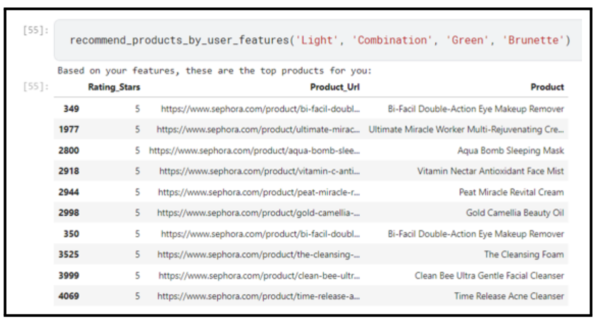
contains words like ”smell” while positive based have words ”love” etc.

Fig. 3: Negative and Positive Review Based

III. Some Simple Approaches

Customer Feature Based Recommendation System

- **Personalized Recommendations:** This approach tailors product suggestions based on user features such as skin tone, type, eye, and hair color.
- **Efficient Filtering:** Products are filtered from a dataset using user-provided features, ensuring relevance and alignment with individual preferences.
- **Rating-based Ranking:** Recommended products are ranked by rating stars, facilitating user satisfaction and simplifying the selection process.



```
[55]: recommend_products_by_user_features('Light', 'Combination', 'Green', 'Brunette')
Based on your features, these are the top products for you:
```

[55]:	Rating_Stars	Product_Url	Product
349	5	https://www.sephora.com/product/bi-facil-doubl...	Bi-Facil Double-Action Eye Makeup Remover
1977	5	https://www.sephora.com/product/ultimate-mirac...	Ultimate Miracle Worker Multi-Rejuvenating Cre...
2800	5	https://www.sephora.com/product/aqua-bomb-slee...	Aqua Bomb Sleeping Mask
2918	5	https://www.sephora.com/product/vitamin-c-anti...	Vitamin Nectar Antioxidant Face Mist
2944	5	https://www.sephora.com/product/peat-miracle-r...	Peat Miracle Revital Cream
2998	5	https://www.sephora.com/product/gold-camellia-...	Gold Camellia Beauty Oil
350	5	https://www.sephora.com/product/bi-facil-doubl...	Bi-Facil Double-Action Eye Makeup Remover
3525	5	https://www.sephora.com/product/the-cleansing-...	The Cleansing Foam
3999	5	https://www.sephora.com/product/clean-bee-ultr...	Clean Bee Ultra Gentle Facial Cleanser
4069	5	https://www.sephora.com/product/time-release-a...	Time Release Acne Cleanser

Fig. 4: A sample for User feature based recommendation system

Collaborative Filtering with LightFM

LightFM is a cutting-edge recommendation algorithm that merges the collaborative filtering and the matrix factorization techniques to offer very personalized recommendations.

On the contrary to the conventional techniques such as Singular Value Decomposition (SVD), LightFM provides several benefits, which is why it is the commonly used method for recommendation systems. The SVD-based approaches usually fail in the case of big data and do not have the capability of introducing the extra item features except for the ones that are about user-item interactions.

LightFM overcomes the previous problems by the combination of collaborative filtering with matrix factorization, thus it is able to learn the complex patterns of the user behavior and the characteristics of the item. Its multiplicity, scalability, and performance are the reasons that make it a useful tool for designing recommendation systems in different fields.

Personalized Recommendations: The lightFM's feature of the recommendation-of-personality-specific is the output of its successful application of the collaborative filtering. LightFM figures out the analogues and connotations that are common among the users who have the same liking. LightFM, by considering these studies, is able to create the recommendations for every user according to his/her particular taste and interests. Thus, LightFM is able to deal with the data given and pick the items that are most suitable for the most people's tastes, therefore, the recommendation experience is enhanced.

Matrix factorization: Matrix factorization is a method used in recommendation systems to extract underlying patterns or features from user-item interaction data.

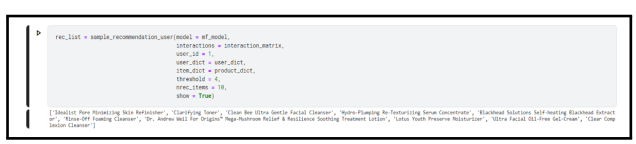
As we know, you can represent user-item interactions in a matrix, where rows represent users, columns represent items (such as skincare products), and the cells contain information about interactions (e.g., skin types, ratings). However, this matrix is usually sparse because not all users have interacted with all items. For example, a user may have only rated or purchased a few skincare products out of a large catalog.

Matrix factorization techniques like LightFM aim to decompose this sparse user-item interaction matrix into two lower-dimensional matrices: **one representing users** and the other **representing items**. Each user and item is represented by a vector of latent factors or features. These latent factors capture underlying characteristics that influence user-item interactions, such as preferences, tastes, or product attributes. By breaking down the interaction matrix into these latent factors, matrix factorization effectively reduces the dimensionality of the data while preserving important information about user-item relationships.

LightFM is a library that employs the matrix factorization in combination with other features such as hybrid models (collaborative and content-based filtering) and side information (user demographic or item attributes).

With LightFM, the recommendation system solves the problem of users and items by finding the optimal latent factors that decrease the difference between the reconstructed interaction matrix and the observed interactions.

The system of optimization is based on the updated latent factor vectors that are the result of the repeated applications of the methods like the stochastic gradient descent until the model is stabilized and converges to a solution that represents the user-item relationships perfectly.



```
rec_list = sample_recommendation_user(model = mf_model,
                                      interactions = interaction_matrix,
                                      user_id = 1,
                                      user_dict = user_dict,
                                      item_dict = product_dict,
                                      threshold = 4,
                                      nrec_items = 10,
                                      show = True)
['Idealist Pore Minimizing Skin Refinisher', 'Clarifying Toner', 'Clean Bee Ultra Gentle Facial Cleanser', 'Hydro-Plumping Re-Texturizing Serum Concentrate', 'Blackhead Solutions Self-heating Blackhead Extract
or', 'Rinse-Off Foaming Cleanser', 'Dr. Andrew Weil For Origins™ Mega-Mushroom Relief & Resilience Soothing Treatment Lotion', 'Lotus Youth Preserve Moisturizer', 'Ultra Facial Oil-Free Gel-Cream', 'Clear Comp
lexion Cleanser']
```

Fig. 5: A sample for Collaborative filtering with Light FM

Content-based Recommendation System

Content-based recommendation in the field of our skincare product recommendation system is about recommending products that are similar to the ones the user has already used and liked. Instead of depending only on the user-item interactions, content-based recommendation studies the characteristics of skincare products to find the similarities and make the personal recommendations. This way of thinking is very useful in skincare, for instance, the product effect was always affected by the specific ingredients, product types, and brands.

In our implementation, we utilize two key techniques for content-based recommendation: attribute-based similarity and TF-IDF vectorization.

Attribute-based Similarity: Our system analyses different attributes of skincare products like ingredients, product type, and brand to determine the attribute-based similarity. By comparing the attributes of skincare products, we identify similarities and recommend items that closely match the user's preferences and past interactions.

TF-IDF Vectorization: TF-IDF (Term Frequency-Inverse Document Frequency) vectorization is a technique that quantifies the significance of words in product descriptions.

We begin by gathering product descriptions from our dataset, which are loaded with the information about each skincare product, such as its benefits, features, and usage instructions.

TF-IDF is a method that assigns a numerical value to each word in the product descriptions, showing how important that word is to the overall description. Words that are often used in a product description but are not used as often in other descriptions are considered to have higher weights. These numerical values form a vector which represents each product description and thus, distinguish its special features from the other products.

Through the comparison of the TF-IDF vectors of different items, we can find the similarity of the products and suggest the ones that are similar to the ones the user has liked or shown interest in the past.

For instance, take a user who has in the past liked skincare products with hyaluronic acid and vitamin C. The system recognizes other products with similar ingredients and product types like serums and moisturizers, through attribute-based similarity. Besides, TF-IDF vectorization emphasizes the words like "hydrating" and "brightening" in product descriptions and thus improves the recommendations. Therefore, the user gets the skincare plans that suit his/her likings and requirements.

```
content_recommendations('The Rice Polish Foaming Enzyme Powder')
```

[102...

	Product	Ing_Tfidf	Rating
38	The Essence Plumping Skin Softener	saccharomyces, camellia, sinensis, leaf, clado...	4.4
90	Gold Camellia Beauty Oil	caprylic, capric, triglyceride, ethylhexyl, pa...	4.6
108	Purifying Cleansing Gel	hydrogenated, starch, hydrolysate, diglycerin,...	4.5
165	Clear Complexion Cleanser	hydrogenated, starch, hydrolysate, disodium, c...	4.4
43	Luminous Dewy Skin Mist	glycerin, squalane, olive, origin, cyclopentas...	4.0

Fig. 6: A sample for content based recommendation system

CLASSIFICATION BASED ON FACE

In real life, people often lack awareness about their skin conditions, such as dryness, redness, or acne, which can impact their skincare routines and overall well-being. To address this gap, our study focuses on leveraging facial image analysis to classify skin conditions and types, providing valuable insights to users about their skin health.

Our study utilizes two primary datasets: one for skin type classification and another for skin condition detection. The skin type dataset consists of facial images labeled with skin types (e.g., oily, dry, normal), while the skin condition dataset contains images labeled with various skin conditions (e.g., redness, bags, acne). These datasets serve as the foundation for training and evaluating our classification models.

Fig. 7: Skin type data

Fig. 8: Skin condition data

About the models

For skin type classification, we employ a ResNet-based convolutional neural network (CNN) architecture.

ResNet is a deep learning architecture renowned for its effectiveness in image classification tasks. By analyzing facial features extracted from the images, the ResNet model can accurately classify skin types based on learned patterns and characteristics.

For skin condition detection, we utilize a Vision Transformer (ViT) model. ViT is a state-of-the-art transformer-based architecture originally designed for image classification. It leverages self-attention mechanisms to capture intricate patterns and features in facial images, enabling precise detection of skin conditions like redness, bags, and acne.

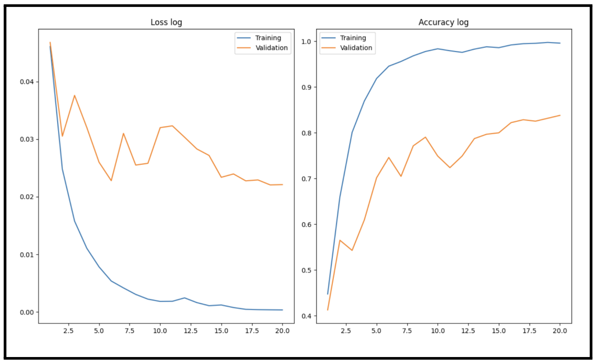


Fig. 9: Result – Skin type model

- Loss log: This graph illustrates how the loss (typically measured using cross-entropy loss) changes over training epochs. A decreasing loss indicates that the model is learning and improving its predictions.
- Accuracy log: This graph shows how the accuracy of the model on both training and validation datasets evolves over epochs. Increasing accuracy over epochs signifies that the model is becoming more proficient at classifying skin types.

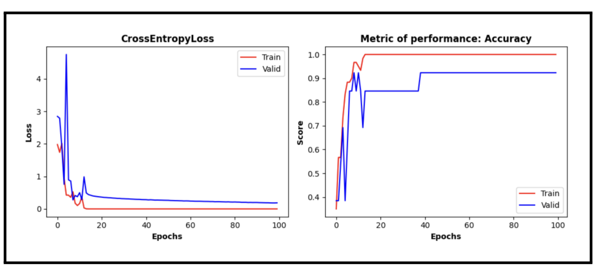


Fig. 10: Result – Skin Condition model

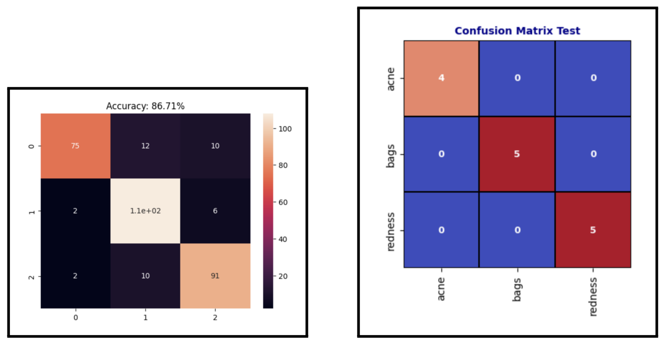
- Cross-entropy loss vs. epochs graph: Similar to the skin type model, this graph depicts the change in cross-entropy loss over training epochs. A decreasing loss indicates that the model is learning to make more accurate predictions.
- Accuracy vs. epochs graph: This graph displays how the accuracy of the model on training and validation datasets changes over epochs. Increasing accuracy suggests that the model is improving in its ability to detect skin conditions like redness, bags, and acne.

A. *Confusion Matrix*

The confusion matrix is a basic tool for evaluating the effectiveness of classification models. This provides an exhaustive summary of the model's prediction by comparing them with the true labels. The rows and columns are organized in a matrix, where each row corresponds to the real class and each column refers to predicted class labels.

Correct predictions are shown on the diagonal elements while misclassification is indicated when we go off diagonally.

Conversely, this particular confusion matrix enables us to scrutinize the behavior of our classifier across several classes and see how it confuses them. The information that could be learned from this study cannot be overestimated when it comes to understanding strong and weak points in classification as well as improvements necessary.



(a) For Skin type data (b) For Skin Condition data

Fig. 11: Confusion Matrices

- The diagonal elements of the confusion matrix represent the number of correctly classified instances for each class. We observe that the majority of instances fall along the diagonal, indicating that the model performs well overall.
- Off-diagonal elements highlight instances of misclassification, where the model predicts the wrong class label. We can identify specific patterns of misclassification and assess which classes are frequently confused with one another.

To sum it up, the observations we got from the confusion matrices of our skin type and skin condition classification models are a key to evaluate their performance. We notice that the models show the effectiveness in the classification of skins types and conditions, but certain patterns of misclassification are also observed. These misclassifications may originate from the class imbalance, complexities within the dataset or the limitations of the model architecture. The problems of overfitting and the reduction of accuracy may be tackled and the insights got from the confusion matrix can be used to make the models better and more reliable in the real-world application.

IV. Glove embedding + Deep CoNN Model

One of the main limitations of the aforementioned models is that they do not fully utilize all the latent information that can be extracted from user review texts. Using tools such as word2vec and TFIDF to capture the underlying meaning of reviews will always be sub-optimal since they do not capture context properly. Hence, we decided to use Glove embeddings, which not only transforms each word in a review to a 200-dimensional vector, but also captures semantic relationships by analyzing patterns between the nearest neighbors of a word. In addition to the review text, we appended information present in other columns such as SkinTone, SkinType, ProductName, etc. to ensure that the DeepCoNN model uses as much information as possible.

DeepCoNN, or Deep Cooperative Neural Networks, is a special type of deep learning model built for recommendation systems. It has a similar structure as a dual or siamese neural network, consisting of two parallel neural network structures known as DeepCoNN towers. Each of these DeepCoNN towers understand user features and product features respectively. In short, the DeepCoNN recommender model takes a user and product embedding as input and outputs the estimated rating of the user on that product.

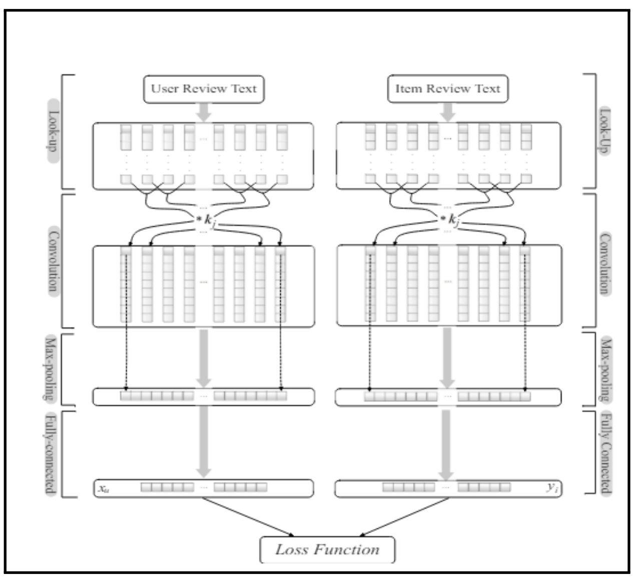


Fig. 12: DeepCoNN architecture

To effectively utilize the above, we implemented a function that finds predicted ratings of a user on all products in the dataset and outputs the products that correspond to the top 5 estimated ratings. We found that this provides the user with extremely robust and sometimes even non-obvious recommendations. For example, one user mentions having oily skin with big pores in one of his reviews. The model recommends toners and moisturizers to him (these are known to seal large pores in your skin), deducing that he would rate these products highly despite the user only having rated cleansers beforehand.

V. Conclusion

In conclusion, we propose a solution that looks for certain skin conditions from a user's facial image and provides skin-care product recommendations based on data extracted from the image. We also highlight a text embedding model (Glove) that can use this data along with their review texts and ratings to form user and product embeddings. These embeddings allow the DeepCoNN model to provide recommendations based on extremely subtle patterns and latent variables.

References

[1] https://medium.com/@fitfindings/sephora-skin-care-essentials-glow-radiance-d722c0c1ae81
[2] https://medium.com/@vatsaldadiavd/movie-recommender-using-nlp-word2vec-and-tf-idf-8860812e15c9
[3] https://medium.com/@dikosaktiprabowo/hybrid-recommendation-system-using-lightfm-e10dd6b42923
[4] https://medium.com/@tusharaggarwal/master-the-power-of-glove-a-step-by-step-guide-6249a61a6e1f
[5] https://towardsdatascience.com/co-operative-graph-neural-networks-34c59bf6805e
[6] Zheng, L., Noroozi, V., Yu, P. S. (2017). Joint Deep Modeling of Users and Items Using Reviews for Recommendation (Version 1). arXiv. https://doi.org/10.48550/ARXIV.1701.04783