Sephora Product User Predictions

Josh Puray
jpuray@ucsd.edu

Nikki Rejai
nrejai@ucsd.edu

1 INTRODUCTION

The skincare industry, marked by its continuous innovation and a vast array of products, plays a pivotal role in the beauty and personal care landscape. As consumers increasingly turn to online platforms for beauty purchases, the wealth of available product information and user reviews becomes an invaluable resource. Our study focused on skincare products from the Sephora online store. We employed several collaborative filtering models to unravel patterns within user-product interactions, aiming to predict user ratings and offer personalized recommendations.

2 PRELIMINARIES

The analysis involved two distinct data sets. The initial data set encompassed a comprehensive compilation of over 8,000 beauty products obtained from the Sephora online store. It contained detailed information, including product and brand names, prices, ingredients, ratings, and other relevant attributes. However, for the analysis, only skincare products were considered due to the reviews being scraped exclusively from the skincare category section of the Sephora website, resulting in a subset of 2,420 unique skincare products.

The second data set consisted of reviews specifically extracted from the Skincare category, totaling 1,094,411 reviews. Collected in March 2023, the data set spans from August 2008 to March 2023 and involves 503,216 unique users and 2,351 unique items appearing in the reviews. Notably, not all skincare products from the comprehensive product data set are present in the reviews data set. The review data set provides crucial details, including user identification, product identification, review text, rating, and additional relevant parameters. For our analysis, however, the features primarily utilized were the product IDs, user IDs, and ratings of each product for both data sets.

It is important to note that when creating some figures (Fig. 1 & Fig. 2) outliers were removed when creating the graphs to enhance the clarity of the distributions. When determining which products were outliers we used the Interquartile Range (IQR), which divides the data set into four equal parts known as quartiles. Using this we created a lower bound and upper bound defined by the following equations.

$$IQR = Q3 \ - Q1$$

$$lower\ bound = Q1 - 1.5 * IQR$$

$$upper\ bound = Q3 + 1.5 * IQR$$

Q1 represents the 25th percentile and Q3 represents the 75th percentile. Anything out of this range was excluded from the graph. When creating the models, however, these outliers were not removed, as our analysis suggested that popular items are not more likely to receive higher ratings (More on this in Section 3.2)

3 EXPLORATORY ANALYSIS

3.1 Uni-variate Analysis

Upon analyzing the ratings distribution (Table 1), it is evident that higher ratings are much more prevalent in the data than lower ratings. Indicating a general tendency of users to rate products higher. Within the data set an average rating is 4.299 and a median rating of 5.

Table 1: Rating Distribution

rating	count	percentage
5	698,951	0.638655
4	199,389	0.182188
3	81,816	0.074758
2	53,032	0.048457
1	61,223	0.055942

Next, we wanted to see the number of reviews each product has. Doing so would allow us to confirm that each product has a "reasonable" number of reviews for our model. If some items dominated the number of reviews it may not be best for our model. When analyzing the distribution of the number of reviews for each product (Fig. 1) there is a notable right skewness of the data. Revealing that some products are much more popular than others. The average number of reviews for each product received is 452.24, while the median is 153.

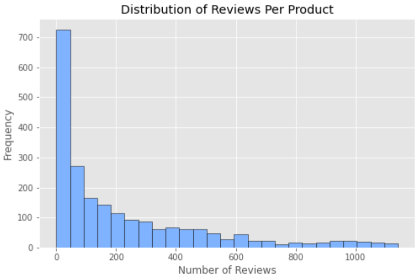

Figure 1: Review Counts

3.2 Bi-variate Analysis

Upon uncovering that some items in the data set were much more popular than others, we wanted to see if there was a correlation between the popularity of an item and it's average rating. When plotting the average rating of an item and its respective number of reviews we observed that there is no strong correlation between the two and that there many items in the data set with high ratings despite having a low number of reviews (Fig. 2).

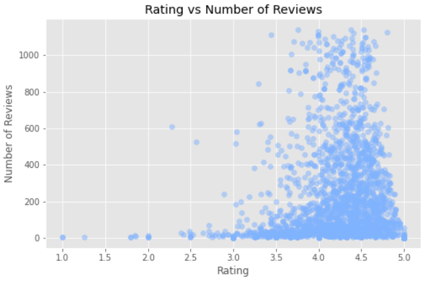

Figure 2: Rating vs. Review Count

An interesting observation we made about data is that users have differing tendencies when rating products. Examining the average ratings of each user revealed apparent clusters (Fig. 3). With a vast majority of users tend to rate items a perfect score of 5 every time. Contemplating a model capable of identifying these different clusters and users could potentially lead to more accurate ratings.

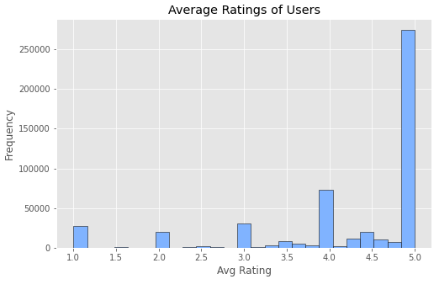

Figure 3: Rating vs. Review Count

4 TASK: RATING PREDICTIONS

Our predictive task is to estimate the ratings that users would assign to specific beauty products within our dataset. For this task we used the authorID, productID, and ratings columns of the review dataset. We began by partitioning the review dataset into a train-test split, allocating 75% to the training set and 25% for the test set.

Table 2: Reviews

authorID	productID	rating
1741593524	P504322	5
6941883808	P420652	2
6015087794	P7880	4

To evaluate the accuracy of our predictive models we employed root mean squared error (RMSE). RMSE is the square root of the mean of the squared differences between predicted and observed values. It gives the average magnitude of the errors in the predictions, ensuring a robust evaluation framework.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}}$$

Our objective was to identify the model that minimizes RMSE, thereby establishing the most effective predictive model.

4.1 Baseline

We began our exploration for predictive models by implementing a baseline model which uses the products dataset to find the overall average rating for a given product. The baseline model predicts that all users will give the average rating for any given product. This baseline serves as a benchmark against which the performance of our more complex models can be assessed.

$$r(u,i) = avgRating[i]$$

The baseline model had an RMSE of 1.115.

4.2 Collaborative Filtering

We then developed and implemented a collaborative filtering model. In collaborative filtering, the prediction of user ratings for products is enhanced by leveraging the similarities between users and items. This approach aims to exploit patterns in user behaviors and preferences, allowing for more accurate predictions on whether a user is likely to rate a particular product favorably. By examining relationships between users and items, collaborative filtering seeks to uncover latent connections in the data that contribute to more refined and personalized predictions. To implement this model we used the following equation.

$$r(u,i) = \alpha + \beta_{\text{user}} + \beta_{\text{item}}$$

This parameterized equation allows the collaborative filtering model to adapt to individual user and item characteristics, enhancing its ability to make nuanced predictions. The α, β_{user}, and β_{item} terms enable the model to capture and adjust for global trends, user-specific preferences, and item-specific characteristics, respectively. Our collaborative filtering model had an RMSE of 1.126, outperforming the baseline.

4.2.1 *Jaccard Similarity.*

$$\text{jaccard\_sim}(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

We can also build a more complex collaborative filtering model which uses Jaccard Similarity to measure the similarity between users and items.

$$r(u,i) = \frac{\sum_{j \in I_u \setminus \{i\}} R_{u,j} \cdot \text{jaccard\_simi}(i,j)}{\sum_{j \in I_u \setminus \{i\}} \text{jaccard\_sim}(i,j)}$$

This model uses the weighted sum of ratings from similar items ($R_{u,j}$) based on Jaccard similarity ($\text{Sim}(i,j)$), to provide nuanced predictions by considering shared user preferences among items in the set I_u. The incorporation of Jaccard similarity enables the model to discern and capitalize on latent patterns within user-item interactions, resulting in a more accurate collaborative filtering approach. This model had an RMSE score of 1.052.

4.2.2 *Cosine Similarity.*

$$\text{cosine\_sim}(\mathbf{A}, \mathbf{B}) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|}$$

We can use cosine similarity as the similarity metric of the collaborative filtering model as well. Cosine similarity captures the directional similarity between item vectors, making it particularly suitable for scenarios where the magnitude of vectors may vary.

$$r(u,i) = \frac{\sum_{j \in I_u \setminus \{i\}} R_{u,j} \cdot \text{cosine\_sim}(\mathbf{I_i}, \mathbf{I_j})}{\sum_{j \in I_u \setminus \{i\}} \text{cosine\_sim}(\mathbf{I_i}, \mathbf{I_j})}$$

Cosine similarity takes in two vectors I_i and I_j which are the item vectors corresponding to i and j respectively. This gave us a slightly better RMSE score of 1.050.

4.3 Matrix Factorization

We then tried using matrix factorization to make more accurate rating predictions. Matrix factorization is a powerful approach within collaborative filtering, which decomposes the user-item interaction matrix into lower-rank matrices that capture latent features of users and items. In this study, we utilized matrix factorization to enhance the accuracy of our rating predictions. We first began by using the Singular Value Decomposition (SVD) algorithm as it is a prominent matrix factorization technique. SVD decomposes the original user-item interaction matrix into three matrices: a user matrix, a diagonal matrix containing singular values, and an item matrix. The resulting lower-rank matrices represent users and items in a latent space, allowing the model to capture hidden patterns in the data. Our choice of SVD for collaborative filtering stems from its effectiveness in extracting meaningful latent factors, facilitating accurate predictions of user-item interactions. We imported the model from the Surprise library, a Python scikit for building and analyzing recommender systems that deal with explicit rating data. The model's prediction of ratings is then determined by the following equation:

After fine-tuning hyperparameters, our SVD model achieved an overall RMSE of 0.904 on the test set. This result shows a substantial improvement from the RMSE values obtained by previous models, indicating a significantly enhanced proficiency in predicting user ratings with this particular model.

To robustly evaluate the performance of our collaborative filtering model, we employed k-fold cross-validation. This technique partitions the data set into k equally sized folds, iteratively training the model on k-1 folds and evaluating it on the remaining fold. The process is repeated k times, ensuring that each fold serves as both training and validation data. The RMSE values for each iteration of where k=5 were 0.8871, 0.8899, 0.8847, 0.8910, and 0.8893 (Table 3). These consistent and low RMSE values across different validation splits reinforce the reliability and stability of our collaborative filtering model. The iterative nature of k-fold cross-validation ensures that our model generalizes well to unseen data, validating its effectiveness in making accurate predictions.

Table 3: SVD K-Fold Cross-Validation RMSE

Fold	RMSE
1	0.8871
2	0.8899
3	0.8847
4	0.8910
5	0.8893

5 LITERATURE REVIEW

During our research for sufficient models for our task, we were inspired by several papers that developed and implemented recommender systems. We found that a vast majority of papers cited using Collaborative Filtering methods utilizing the similarities between user behaviors to make predictions about a user's rating. We examined several of these models below and utilized many of them when developing models of our own.

Avi Rana and K. Deeba [1] developed an Online Book Recommendation System utilizing Jaccard Similarity (JS). Using a dataset encompassing books, users, and their ratings, they achieved an RMSE of 1.504. However, a notable drawback highlighted in their paper is that JS "considers the number of users who have rated the books but does not account for the absolute rating," potentially leading to recommendations from books a user rated low. Which may lead to recommendations from a book a user gave low ratings to. Additionally, the paper addresses the trust issue inherent in user feedback, emphasizing the need to acknowledge the possibility that user recommendations may lack genuineness.

In Mcauley's discussion on recommendation systems [4], various similarity measurements are explored. Of particular note is the mention of Cosine Similarity as a metric that addresses the challenges posed by Jaccard Similarity. Cosine Similarity represents interactions as vectors rather than sets. This new metric enables the calculation of similarity for numerical interaction data and leverages the inherent polarity of ratings, where higher ratings convey more positivity and lower ratings signify more negativity. Given our task of predicting user ratings, the adoption of this metric did not seem to improve the model by a significant amount. Contributing to a marginal improvement in RMSE by a mere 0.002, as detailed in Section 4.2.2.

Robert M. Bell, Yehuda Koren and Chris Volinsky [5], the winners of the "Netflix Grand Prize" combine multiple different approaches by blending multiple predictors to improve the RMSE of their model, an ensemble of methods. The models used in this solution feature Asymmetric factor models, regression models, Restricted Boltzmann Machines with Gaussian visible units, Matrix factorization, just to name a few. In their winning solution, they blended 107 predictors to obtain an RMSE of 0.8712. Motivated by their success, we opted to integrate Singular Value Decomposition into one of our models. Through this approach, we saw a significant improvement in RMSE performance.

Koren [3] created a subsequent paper detailing his contribution to the winning solution, wherein he discussed some of his earlier models. Within this paper, he explores a straightforward collaborative filtering model aimed at capturing tendencies for certain users to assign higher ratings than others, as depicted in the initial equation in section 4.2. Nevertheless, his final model equation proved to be considerably more intricate than ours, incorporating temporal dynamics and the frequency of user ratings. Despite this, when employing the model, we observed that we could achieve comparable results to Koren's. Notably, our findings indicated that a model utilizing matrix factorization outperformed the simple collaborative filtering model, consistently reducing the Root Mean Squared Error (RMSE) to a greater extent even beating out our other more advanced collaborative filtering models such as Jaccard and Cosine similarity.

Keshava, M., Reddy, P., Srinivasulu, S., & Naik, B. [2] developed a recommendation system using the same Netflix dataset. Instead of employing ensemble methods, they opted to optimize multiple models independently. Their research involved various models, including XGBoost, Surprise Baselineonly, Surprise KNNBaseline, and Matrix Factorization SVD. Through meticulous optimization of each model, they identified SVD++ as the most effective. SVD++ is an extension of the SVD model that incorporates implicit feedback, such as implicit ratings and introduces bias terms for each user and item to enhance the capture of user preferences. Their finalized model achieved an RMSE of 1.0675, significantly lower than the winning solution's ensemble method. However, it's important to acknowledge that resource constraints prevented them from utilizing the entire training dataset, making a direct comparison with the winning solution challenging.

6 RESULTS

In our evaluation of diverse collaborative filtering models for predicting ratings of Sephora skincare products, the Singular Value Decomposition (SVD) algorithm emerged as the most accurate, achieving an RMSE of 0.904. This finding is further supported by consistently low RMSE values observed across various validation splits during k-fold cross-validation. These results underscore the robustness and reliability of the SVD-based model in accurately predicting user ratings for Sephora skincare products

One practical application of our user-product rating prediction task is to generate personalized product recommendations. Leveraging our model, we can suggest products to users based on their rating history. As a matrix factorization model provides rating predictions for any user-item pair, we can sort these predictions and recommend the top-k items that the user has not yet purchased.

Consider an example where a user has a history of purchasing primarily moisturizers. Our recommendation system reflects this preference by suggesting other moisturizers and hydrating products in the following way.

By accurately determining the example user's interest in moisturizers based on their purchase history, our recommendation system showcases the model's capability to capture nuanced preferences. The precision in suggesting similar products aligns with the user's distinct taste, demonstrating the model's proficiency in tailoring recommendations to individual preferences. This enhances the overall shopping experience and demonstrates robustness and reliability of our SVD-based model in translating predictions into actionable and user-centric product recommendations.

Table 4: Sample User's Rating History

Actual Rating	Product	Brand
4.3202	Revitalizing Supreme+ Youth Power Creme Moisturizer	Estée Lauder
4.4533	Urban Environment Oil-Free Sunscreen Broad-Spectrum SPF 42	Shiseido
4.6311	Urban Environment Fresh-Moisture Sunscreen Broad-Spectrum SPF 42	Shiseido
4.7556	Ultra Facial Advanced Repair Barrier Cream	Kiehl's Since 1851
4.3202	Mini Revitalizing Supreme+ Youth Power Creme Moisturizer	Estée Lauder

Table 5: Sample User's Recommendations

Pred Rating	Product	Brand
5	C-Rush Vitamin C Gel Moisturizer	OLEHENRIKSEN
5	T.L.C. Sukari Babyfacial AHA + BHA Mask	Drunk Elephant
5	Hello FAB Coconut Skin Smoothie Priming Moisturizer	First Aid Beauty
5	Rose Face Mask	fresh
4.966	Evercalm Overnight Recovery Balm	REN Clean Skincare

7 CONCLUSION

Our evaluation of collaborative filtering models for Sephora skincare product ratings identifies the Singular Value Decomposition (SVD) algorithm as the most accurate. Beyond predictive accuracy, we also explored generating personalized product recommendations by leveraging the the SVD model to suggest products based on user rating history, offering a tailored shopping experience.This capability holds significant potential in enhancing user satisfaction and engagement in the competitive beauty retail space, and gave us an insight into how large scale, retail recommender systems work.

REFERENCES

[1] Avi Rana and K. Deeba 2019 J. Phys.: Conf. Ser. 1362 012130

[2] Koren, Y. (2009). The BellKor Solution to the Netflix Grand Prize.

[3] Keshava, M., Reddy, P., Srinivasulu, S., & Naik, B. (2020). Machine Learning Model for Movie Recommendation System. International Journal of Engineering Research and Technology, 9(V9), 10.17577/IJERTV9IS040741.

[4] McAuley, J. (2022). Personalized Machine Learning. Cambridge: Cambridge University Press. doi:10.1017/9781009003971

[5] R. Bell; Y. Koren; C. Volinsky (2007). "The BellKor solution to the Netflix Prize