Enhanced Product Discovery: A Personalized Recommendation System for the Beauty Industry

Aasika ES[1a], Jeeva D[1b], Sangavi G[1c], Dr. David Maxim Gururaj[2]

[1]PG Student, Department of Mathematics, School of Advanced Sciences,
Vellore Institute of Technology, Chennai, Tamil Nadu, India.
Email: ashika.2023@vitstudent.ac.in, jeeva.2023@vitstudent.ac.in, sangvai.g2023@vitstudent.ac.in

[2]Assistant Professor (Sr), Division of Mathematics, School of Advanced Sciences,
Vellore Institute of Technology, Chennai, Tamil Nadu, India.
Email: davidmaxim.gururaj@vit.ac.in

***Abstract*—Personalization is essential to the beauty and cosmetics market, an advanced recommendation system is essential for improving user satisfaction and product discovery. In order to offer customized, trend-aware suggestions, this project presents a methodology that combines content-based filtering with collaborative filtering. Utilizing cutting-edge methods like TF-IDF, BERT, and ChemBERT, the content-based strategy analyzes product attributes like brand, categories, and chemical composition to match user preferences. Singular Value Decomposition (SVD), Non-Negative Matrix Factorization (NMF), and K-Nearest Neighbors (KNN) are some of the algorithms used in collaborative filtering to assess user-item interaction patterns and extract information from comparable users' preferences. This hybrid framework addresses the drawbacks of the conventional approach while bridging the gap between changing beauty trends and personal preferences.Because it guarantees scalability and versatility, the system is a perfect fit for a wide range of market demands. Its user-centered design encourages recurring business and revolutionizes the purchasing experience by making accurate and insightful suggestions.**

***Index Terms*—Beauty Industry, Personalized Recommendations, ChemBERT, Hybrid Systems, User-Centric Design**

I. Introduction

The retail industry is undergoing a radical change in the current digital era, especially in industries like beauty, as customers depend more and more on online buying channels. Customers may become overwhelmed by the sheer number of products available, making it difficult to choose those that genuinely fit their needs and tastes. As a result, there is an increasing need for more individualized and user-friendly shopping experiences that make it simple for clients to browse through extensive product catalogs. Retailers are now able to provide individualized solutions that satisfy consumer preferences by leveraging data analysis and cutting-edge technology, making the shopping experience more efficient and interesting.In order to satisfy this need, personalized recommendation systems have become an effective tool, providing users with recommendations that are specific to their behavior, preferences, and previous interactions.

An automatic recommender system was created by Al-Badarenah Alsakran (2016) to help students choose their courses. The technology improves decision-making efficiency in academic contexts by generating personalized recommendations based on individual preferences and past interactions. This approach illustrates how machine learning can enhance student satisfaction and streamline academic guidance. Additionally, it lays the groundwork for expanding these systems to more extensive educational fields [1].Lee Hosanagar (2019) used a randomized field experiment to examine how recommender systems affect sales diversity. According to their research, these methods can decrease product diversity by boosting popular items across categories, even while they increase consumer pleasure. In order to prevent market monopolization, they underlined the necessity of striking a balance between fair product exposure and customer happiness. This work affects algorithm design in a way that tackle the issues of diversity in e-commerce [2].A new paradigm known as Recommendation as Language Processing (RLP) was presented by Geng et al. in 2022. Prediction accuracy and user experience are greatly enhanced by this method, which unifies the recommendation process by utilizing language models, individualized prompts, and pretraining. Their model is a flexible tool for a variety of recommendation jobs since it shows how well natural language processing understands user situations. It also emphasizes how pretraining enhances system customization and adaptability [3]. Multi-criteria recommender systems were examined by Al-Ghuribi Noah (2019), who emphasized how well they can combine various review-based criteria to produce precise recommendations. Their study highlights the significance of content created by users and explores advanced techniques. Additionally, they examined the difficulties in managing noisy and insufficient reviews, promoting the use of strong algorithms.algorithms to ensure dependability. This study offers insightful information on how to use thorough data usage to improve suggestion quality [4]. An artificial neural network-based hybrid recommender system was presented by Paradarami, Bastian, and Wightman (2017). By combining collaborative filtering with machine learning models, the system delivers accurate and scalable recommendations suitable for various applications.

Their approach is a dependable solution in situations with little data since it tackles the issues of sparsity and cold-start. The study also emphasizes how neural networks can be tailored to learn intricate user-item interactions [5]. A social recommender system specifically developed for group commerce centered on specific locations. Their method uses network data to make contextually appropriate recommendations that improve group buying experiences. The methodology guarantees increased customer happiness and engagement by utilizing social and geospatial data. This method is especially significant in situations that call for collective decision-making and coordination [6]. In an e-commerce context, Aditya, Budi, and Munajat (2016) compared memory-based and model-driven collaborative filtering methods. Their findings highlight the strengths and limitations of both approaches, offering insights on their application in different business contexts. To overcome the drawbacks of separate strategies, particularly when managing changing user preferences, they suggested hybrid solutions. Businesses might use their work as a guide for choosing suitable recommendation approaches [7]. Lee, Chen used systematic method for innovating and examining smart product-service systems. They employed a case study of a smart beauty service to illustrate their approach, emphasizing the significance of customer-focused innovation in developing effective solutions. Their approach combines product customization and data analytics to enhance service results. This study draws attention to the revolutionary possibilities of combining user-driven service design with digital technologies [8]. The incorporation of multimedia content into recommender systems was examined by Deldjoo et al. (2020). They underlined how using multimedia information such as pictures, videos, and audio—improves user engagement and personalization in recommendation systems. Their analysis highlights difficulties in handling high-dimensional multimedia data and promotes the use of sophisticated computational methods. This study offers a thorough analysis of how multimedia can improve the caliber of recommendations [9].A customized recommendation system was created by Chiu et al. (2021) as part of an intelligent product-service system. Their unsupervised learning-based algorithm analyzes user behavior and service usage trends to provide personalized recommendations. The study emphasizes how crucial unsupervised methods are for revealing hidden user preferences and demands. Furthermore, their results indicate the increasing importance of smart systems in providing scalable and effective recommendations [10]. A graph neural network-based recommender system that takes advantage of the structural connections between users and objects was created by Chen et al. (2020). By combining collaborative filtering and graph learning, their method improves recommendation accuracy by identifying intricate dependencies. The model successfully resolves sparsity concerns and demonstrates how graph neural networks may produce recommendations that are easy to understand. According to their findings, suggestion quality has significantly improved across a range of application domains [11]. In order to improve tailored recommendations, Sun et al. (2021) presented a knowledge-aware recommendation model that integrates external knowledge graphs. Their results demonstrate enhanced performance in cold-start and scarce circumstances, especially in the fields of education and healthcare. Their methodology guarantees more personalization by associating user preferences with pertinent knowledge elements. The The scalability of knowledge graph integration in large-scale recommender systems is also covered in the paper [12]. To solve data privacy issues, Liu et al. (2022) suggested a federated learning approach for recommender systems. Their decentralized approach guarantees safe user data management while achieving performance on par with centralized systems. They showed how well federated learning worked in situations where stringent data compliance was necessary. Their method guarantees smooth model training in dispersed contexts and substantially lowers the risks connected with data breaches [13].Dynamic recommender systems that adjust to shifting user preferences over time were studied by Zhou et al. (2023). Their methodology, which incorporates temporal data, has potential for streaming services where user behavior is subject to rapid change. The study emphasizes how crucial it is to describe temporal dynamics in order to keep recommendations relevant. Furthermore, their methodology facilitates real-time upgrades, which makes it ideal for sectors that move quickly [14]. A multi-modal recommender system that integrates text, image, and audio data for comprehensive user profiling was presented by Wang et al. (2024). In platforms with a lot of multimedia, like social networks and e-commerce, their technology provides better personalization. By tackling the issues of data heterogeneity, their findings also open the door for advancements in multimedia recommendation systems [15].

II. Flow Chart

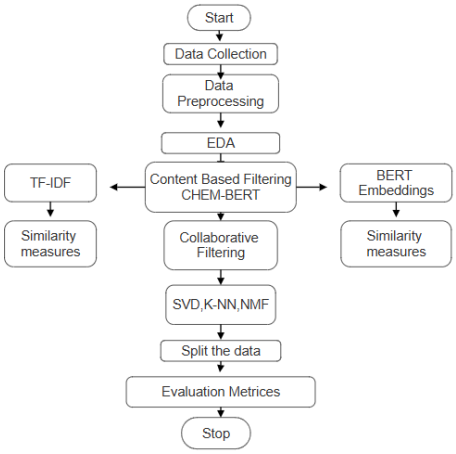

Fig. 1. Flowchart illustrating the product recommendation process

III. Methodology

A. About Dataset

The information used in this research is comprised of two main components: the **product dataset** and the **review dataset**, both gathered via web scraping.

1) Product Dataset: The product dataset contains 8494 rows and 27 attributes, offering comprehensive details about each beauty product. The key attributes include:

- **Product ID**: A distinct identifier assigned to each product.
- **Product Name**: The name associated with the product.
- **Brand ID** and **Brand Name**: Identifiers and names associated with the product's brand.
- **reviews, ratings, and loves_count**: Properties that provide insights into user interaction and feedback.
- **size, variation_type, variation_value, and variation_desc**: Attributes related to product variations, including size and color options.
- **ingredients**: Lists all components in the product.
- **price_usd, value_price_usd, and sale_price_usd**: Pricing information, detailing the product's price structure.
- **limited_edition, new, online_only, and out_of_stock**: Indicators regarding the product's availability and exclusivity.
- **primary_category, secondary_category, and tertiary_category**: Categories used to classify the product.

2) Review Dataset: The review dataset consists of 602,130 rows and 19 attributes, providing extensive information about user feedback. Key features include:

- **author_id**: Identifier for the reviewer.
- **rating**: The rating given by the user for the product.
- **is_recommended**: A field indicating whether the reviewer recommends the product.
- **helpfulness**: A metric evaluating the helpfulness of the review.
- **total_feedback_count, total_neg_feedback_count, and total_pos_feedback_count**: Metrics for evaluating user feedback on the review.
- **review_text** and **review_title**: The main content and title of the review.
- **Review Submission Date**: The date on which the review was posted.
- **Skin Tone, Eye Color, Skin Type, and Hair Color**: Attributes related to the user's physical characteristics.

B. Data Preprocessing

Dropping Irrelevant Attributes: At first, some of the columns that were thought to be less important for analysis were removed from the dataset. These comprised the following attributes: `value_price_usd`, `sale_price_usd`, `size`, `variation_desc`, `variation_value`, `variation_type`, `child_max_price`, and `child_min_price`.

C. Handling Missing Data

1) Ingredients and Highlights Imputation: To address missing values in the `highlights` and `ingredients` columns, a category-based imputation approach was employed. The dataset was grouped by the `primary_category`, and the most frequent values (mode) for `highlights` and `ingredients` within each category were determined. Missing entries in these columns were then replaced using these category-specific modes, ensuring contextually relevant imputations.

2) Tertiary Category Imputation: The overall mode of the column was used to fill in missing values for the `tertiary_category` column, which contains categorical data. This technique preserved data integrity while guaranteeing a consistent and believable replacement.

3) Imputation of Ratings and Reviews: The K-Nearest Neighbors Imputer (`KNNImputer`) was used to impute the ratings and reviews in the numerical columns. This method provides a reliable solution while preserving the statistical characteristics of the dataset by estimating missing values using the similarity between data points.

4) Removing Duplicates: Only distinct combinations of `author_id`, `product_id`, and `rating` remained after duplicates in the reviews dataset were eliminated.

D. Exploratory Data Analysis

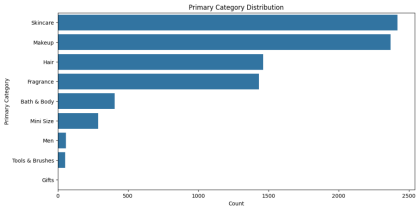

Fig. 2. primary category Distribution

Fig.2 represents the distribution of ratings, with four out of five being the most common. This indicates that buyers have largely given the product or service positive reviews, with a significant number of 4-star ratings. Although the product is highly accepted, there may be space for improvement to achieve a more consistent 5-star rating, as evidenced by the sharp decline in ratings from 4 to 5.

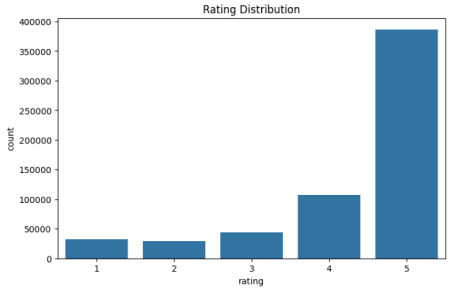

Fig. 3. Rating Distribution

Fig. 6 the distribution of ratings, with four out of five being the most common. This indicates that buyers have largely given the product or service positive reviews, with a significant number of 4-star ratings. Although the product is highly accepted, there may be space for improvement to achieve a more consistent 5-star rating, as evidenced by the sharp decline in ratings from 4 to 5.

E. Content-Based Filtering

1) Term Frequency-Inverse Document Frequency: It is a statistical metric used to determine the significance of a word within a document in comparison to a collection of documents. It considers both the term's frequency (TF) within a document and its inverse document frequency (IDF), which measures how uncommon or common a phrase is throughout the corpus of texts.

a) Term Frequency (TF): TF measures how often a word appears in a document.

$$\text{TF}(t, d) = \frac{\text{Number of times } t \text{ appears in document } d}{\text{Total number of terms in document } d}$$

b) Inverse Document Frequency (IDF): IDF evaluates the importance of a word by assessing how often it appears across all documents, with less frequent words being considered more relevant.

$$\text{IDF}(t) = \log\left(\frac{N}{\text{df}(t)}\right)$$

Where:

- N represents the total number of documents.
- df(t) refers to the number of documents that contain the term t .

c) TF-IDF Calculation: : The final score for a term t in document d is determined by

$$\text{TF-IDF}(t, d) = \text{TF}(t, d) \times \text{IDF}(t)$$

In the context of this research, we applied TF-IDF to several product information textual aspects, including *product_name*, *ingredients*, *primary_category*, and *tertiary_category*. These text features are transformed into numerical vectors by this transformation, where each dimension denotes the significance of a particular term in the corresponding text field.

F. Cosine Similarity

After employing TF-IDF is applied to convert the text input into numerical vectors, and then cosine similarity is computed between these vectors. In a multi-dimensional space, Cosine similarity measures the extent of similarity between two non-zero vectors by evaluating the cosine of the angle between them.

The cosine similarity value ranges from 0 to 1:

- A value of 1 means the vectors are identical (perfect similarity).
- A value of 0 means the vectors are orthogonal (no similarity).

It is calculated using the combined feature vectors, which are created by stacking the *primary_category*, *tertiary_category*, *ingredients*, and *product_name* TF-IDF vectors. This enables the model to compare the general similarity of products using both categorical data and textual features.

The formula for cosine similarity between two vectors A and B is:

$$\text{Cosine Similarity}(A, B) = \frac{A \cdot B}{\|A\|\|B\|}$$

Where:

- $A \cdot B$ is the dot product of vectors A and B.
- $\|A\|$ and $\|B\|$ are the magnitudes (norms) of vectors A and B, respectively.

G. Sentence-BERT

The BERT concept was expanded upon by Sentence-BERT, which was created especially to provide embeddings for sentences or brief paragraphs.It is well-suited for tasks such as measuring semantic textual similarity, clustering, and information retrieval because it offers a way to create fixed-length vector representations (embeddings) of sentences, in contrast to the original BERT model, which was mainly concerned with token-level prediction.

1) Pretrained Model: For every product description, Sentence-BERT generates dense vector embeddings that each convey the meaning of the product in a high-dimensional space. Semantically comparable descriptions are represented by vectors they are positioned closer together in the vector space, reflecting the text's inherent semantic characteristics.These embeddings' dimensionality represents the text's level of complexity, and the distance between vectors indicates how similar two products are to one another. It will be easier to compare and identify similar products based on their textual content because products with similar descriptions will have geographically closer embeddings. This procedure was successfully applied to the dataset, converting the product descriptions into comparable embeddings that may be utilized for similarity-based analysis and clustering tasks.

H. CHEMBERT

This generates embeddings for product constituents, which are represented as chemical names, using the ChemBERT model. The ChemBERT model processes the ingredients to create high-dimensional vector embeddings once they are first Converted into Simplified Molecular Input Line Entry System. Parallel processing was used to effectively manage huge datasets. These embeddings are then further analyzed to determine The similarity between items is assessed according to their chemical structure.The ChemBERT model is utilized to generate embeddings that capture the molecular-level similarities between different substances.

I. Working Mechanism Of CHEMBERT

1. Converting SMILES

The dataset's chemical names are first transformed into SMILES strings, a commonly used format for text-based molecular structure representation. Using ChemBERT, which is trained exclusively on SMILES strings, requires this conversion step. Information about molecular structure is encoded in a line notation called the SMILES string. Atoms, bonds, and their connections are all compactly represented by each SMILES.

2. Embedding Generation and Tokenization

2.1 Tokenization: The ChemBERT tokenizer is used to tokenize the SMILES strings. Each SMILES string is transformed by the tokenizer into a series of tokens that stand in for atoms, bonds, and other molecular characteristics.

2.2 Embedding Generation: The SMILES strings are run through the ChemBERT model after being tokenized. Because it is transformer-based, the model uses several levels of attention to process the token sequence. This method records the connections between atoms and their surroundings in a molecule. The result of the transformer model's processing is a collection of embeddings, each of which is a high-dimensional vector that represents the chemical features. Each chemical in the product's component list is finally represented by the mean of these embeddings.

3. Batch Processing to Increase Efficiency

3.1 Batch Size: A batch processing technique is utilized to create embeddings in smaller pieces (in this case, a batch size of 8) because processing numerous SMILES strings at once can be computationally demanding.

3.2 Efficiency: This aids in accelerating the creation of embeddings, particularly when model inference is done on a GPU.

3.3 Parallel Processing: ThreadPoolExecutor is used for parallel processing in order to further maximize performance. This allows for the simultaneous processing of several products, cutting down on calculation time overall.

J. Combination of Similarity Matrices

Following the creation of embeddings for the components of each product, the embeddings are combined to create a final similarity matrix. Based on their chemical makeup, this matrix shows how similar the goods are to one another. Weighted averages are used to combine the various similarity matrices that have been generated (for example, based on TF-IDF, BERT, and ChemBERT embeddings). This combination of similarity matrices considers various feature categories (chemical and textual) with equal weights:

$$\begin{aligned}\text{Final Similarity Matrix} = {} & 0.33 \times \text{TF-IDF Sim. Matrix} \\ & + 0.33 \times \text{BERT Sim. Matrix} \\ & + 0.34 \times \text{ChemBERT Sim.Matrix}\end{aligned}$$

By using this method, the overall similarity score is guaranteed to be equally influenced by the product description, textual attributes, and chemical constituents.

K. Collaborative Filtering

1. Singular Value Decomposition (SVD): It is a matrix factorization technique commonly used in collaborative filtering for recommender systems. In order to enable the model to predict unobserved interactions (such as ratings or preferences), SVD breaks down the original user-item interaction matrix into three matrices that capture latent components.

1.1 Decomposing the Matrix:

$$R \approx U \cdot \Sigma \cdot V^T$$

Where:

- R represents the original user-item matrix.
- U denotes the matrix corresponding to the users.
- Σ (singular values matrix): A diagonal matrix with singular values that reflect the importance of each feature in explaining the variance of the data.
- V^T (item matrix): Each row represents an item, with latent features capturing item characteristics.

These matrices capture the latent patterns in the data, such as user preferences and item characteristics. Once these factors are learned, predictions for missing ratings can be made by multiplying these matrices.

1.2 Prediction Formula: Once the matrices are decomposed, The estimated rating for a user u and an item i is calculated as:

$$\hat{r}_{ui} = \mathbf{u_u} \cdot \sigma \cdot \mathbf{v_i}^T$$

Where:

- $\mathbf{u_u}$ is the row of the user matrix U corresponding to user u,
- σ is the diagonal of the singular values matrix Σ,
- $\mathbf{v_i}$ is the column of the item matrix V corresponding to item i.

1.3 Latent Factors: Hidden characteristics that influence user-item interactions are captured by the latent factors discovered throughout the SVD process. Although these characteristics are usually not readily apparent, they can help to understand user-item interaction patterns.

1.4 Optimizing the Decomposition: SVD aims to reduce the discrepancy between the actual ratings and the predicted ratings. To do this, a cost function that calculates the discrepancy between observed and projected ratings is minimized.

1.5 Cost Function: The cost function is formulated as:

$$\text{Cost} = \sum_{(u,i) \in K} (r_{ui} - \hat{r}_{ui})^2 + \lambda \left(\|U\|^2 + \|V\|^2 \right)$$

Where:

- $(u, i) \in K$ are the observed ratings (user-item pairs),
- rui represents the true rating given by user u for item i,
- $\hat{r}_{ui}$ denotes the predicted rating,
- λ is a regularization parameter that controls overfitting by penalizing large values in the matrices U and V.

1.6 SVD with Matrix Factorization: SVD reduces the complexity of the original matrix while keeping the most significant features by approximating the user-item matrix with a lower-rank factorization. By capturing the fundamental patterns of user-item interactions, this simplified representation enables the model to produce precise predictions for unobserved ratings.

The user-item ratings are converted by this algorithm into a format that may be used for model training. The dataset is formatted into the Surprise dataset structure by loading it into the Reader class using the Surprise library. After that, the dataset is divided into training and testing sets, where 20% reserved for testing,the remaining 80% applied to training. Training data is subjected to the Singular Value Decomposition (SVD) model from the Surprise library. In order to identify latent characteristics that account for the trends in user ratings, this method factorizes the user-item interaction matrix. The algorithm can forecast ratings for interactions that are not visible since it learns the relationship between users and items.

2. K-Nearest Neighbor (K-NN):

2.1 Data Preparation and Feature Scaling: To make sure the dataset is prepared for the K-Nearest Neighbors (KNN) model, it is first preprocessed. A Surprise dataset is loaded with ratings, and standard scaling methods are used to scale the features. This guarantees that features with wider ranges do not unduly impact the similarity calculation, which is crucial for distance-based algorithms like KNN.

2.2 Product Feature Encoding: One-hot encoding is used for categorical features (like product_name and primary_category) in the product information. Through this procedure, categorical variables are converted into a format that machine learning models can utilize. To help the model better comprehend and calculate product similarities, each category is transformed into a distinct binary feature. This model determines how comparable products are based on user ratings. To gauge how similar two products are to one another, the cosine similarity metric is employed. This method makes the assumption that people will rate two products similarly if their preferences are comparable.

3. Non-Negative Matrix Factorization (NMF): Frequently used in user-item matrices, where the objective is to suggest products to users in light of their previous interactions. NMF is especially helpful because it finds hidden patterns in the data (item attributes and user preferences). The interpretability of the factorization (customers' preferences for particular product features) is guaranteed by the non-negative constraints. NMF assists in revealing hidden correlations in the data by decomposing the huge matrix into smaller, more understandable matrices, enabling tailored recommendations.

3.1 Frobenius Norm: The Frobenius norm is a measure of the matrix's error during factorization. It is computed as:

$$\|R - PQ^T\|_F^2 = \sum_{i,j} \left(R_{ij} - (PQ^T)_{ij}\right)^2$$

The user-item rating matrix R captures the interactions (ratings) between users and their corresponding products (items). The goal of NMF is to decompose R into two smaller matrices:

- P (user feature Matrix), where each row represents a user, and each column corresponds to a latent feature associated with that user.
- Q (item feature Matrix), where each row represents an item, and each column corresponds to a latent feature of that item.

3.2 Gradient Descent: Gradient descent is usually used by NMF to minimize the objective function. Iteratively updating their values is done by computing the objective function's gradients with respect to the matrices P and Q.

IV. Evaluation Metrics

1. Precision

Precision is a metric that evaluates the accuracy of the positive predictions made by the model. In a recommendation system, it is used to assess the proportion of recommended items that are relevant.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

Where:

- **True Positives (TP):** The count of items that were accurately recommended and were actually preferred by the user.
- **False Positives (FP):** The number of items that were recommended but were not liked by the user.

2. Recall

This calculates recommendation system's ability to identify all relevant items. It evaluates how well the model retrieves all the items that the user actually likes, irrespective of whether they were recommended or not.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

Where:

- **True Positives (TP):** As previously defined.
- **False Negatives (FN):** The number of items that were not recommended but were actually liked by the user (i.e., the item was relevant but not recommended).

3. F1-Score

The F1-Score is the harmonic mean of precision and recall, offering a single metric that balances both. It is particularly useful when there is a need to balance the trade-off between precision and recall.

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

4. Mean Absolute Error (MAE)

The Mean Absolute Error is a metric that measures the accuracy of predicted ratings by calculating the average absolute differences between the predicted and actual ratings.

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|Y_i - \hat{Y}_i|$$

Where Yi is the true rating, and ˆ Yi is the estimated rating for the i-th product.

5. Root Mean Squared Error (RMSE)

RMSE is an evaluation metric that gauges the model's accuracy, giving more importance to larger errors.It is determined by

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(Y_i - \hat{Y}_i)^2}$$

Where Y_i is the actual rating, $\hat{Y}_i$ refers to the predicted rating, while N indicates the total number of predictions.

V. Result and Conclusion

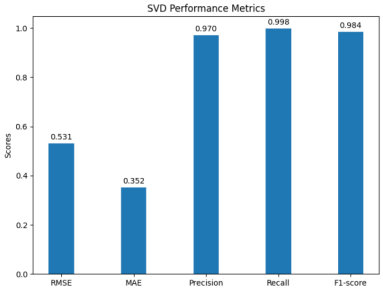

Fig. 4. SVD Performance Metrics

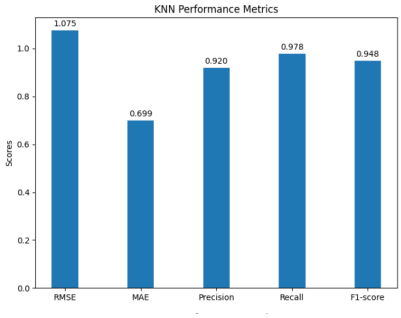

Fig. 5. KNN Performance Metrics

With the lowest RMSE (0.5315) and MAE (0.3523) among the evaluation measures, the SVD model performs better than the others, suggesting higher overall prediction accuracy. Given their larger RMSE and MAE values, the K-NN and NMF models appear to be less successful at rating prediction.

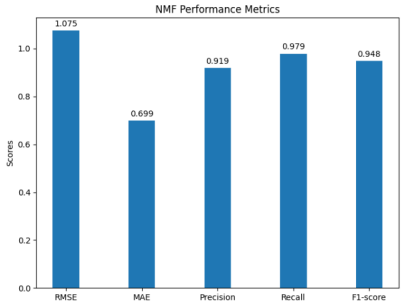

Fig. 6. NMF Performance Metrics

All three models had relatively high precision, recall, and F1-scores, with SVD marginally outperforming the others in these categories. All things considered, SVD is the most successful model since it exhibits the best balance between accuracy and recommendation quality

VI. Future Work

- **Using Hybrid Models:** By taking into account both item similarity and user preferences, combining collaborative filtering strategies with content-based filtering approaches, such as ChemBERT, will improve recommendation accuracy and produce more individualized recommendations.
- **Sentiment Analysis for Recommendations Based on Content:** Incorporating sentiment analysis into user comments or product evaluations can enhance content-based filtering and provide more profound insight into user interests. By considering sentimental nature of user reviews or ratings, recommendations could be improved by providing additional context.
- **Overcoming the "Cold Start" Issue:** A key area to future development is creating methods to tackle "Cold start" issue, which arises when there is a lack of adequate data for new users or items.Techniques such as hybrid models, transfer learning, or utilizing external data sources could be explored to provide better suggestions in these scenarios.

VII. References

[1] Aditya, P. H., Budi, I., Munajat, Q. (2016, October). A comparative analysis of memory-based and model-based collaborative filtering on the implementation of recommender system for E-commerce in Indonesia: A case study PT X. In *2016 International Conference on Advanced Computer Science and Information Systems (ICACSIS)* (pp. 303-308). IEEE.

[2] Al-Badarenah, A., Alsakran, J. (2016). An automated recommender system for course selection. *International Journal of Advanced Computer Science and Applications, 7*(3), 166-175.

[3] Al-Ghuribi, S. M., Noah, S. A. M. (2019). Multi-criteria review-based recommender system–the state of the art. *IEEE Access, 7*, 169446-169468.

[4] Chen, J., Li, X., Li, M., Huang, G. (2020). Graph neural network for recommendation. *ACM Transactions on Information Systems, 38*(4), 1-28.

[5] Chiu, M. C., Huang, J. H., Gupta, S., Akman, G. (2021). Developing a personalized recommendation system in a smart product service system based on unsupervised learning model. *Computers in Industry, 128*, 103421.

[6] Deldjoo, Y., Schedl, M., Cremonesi, P., Pasi, G. (2020). Recommender systems leveraging multimedia content. *ACM Computing Surveys (CSUR), 53*(5), 1-38.

[7] Geng, S., Liu, S., Fu, Z., Ge, Y., Zhang, Y. (2022, September). Recommendation as language processing (rlp): A unified pretrain, personalized prompt predict paradigm (p5). In *Proceedings of the 16th ACM Conference on Recommender Systems* (pp. 299-315).

[8] Lee, C. H., Chen, C. H., Trappey, A. J. (2019). A structural service innovation approach for designing smart product service systems: Case study of smart beauty service. *Advanced Engineering Informatics, 40*, 154-167.

[9] Lee, D., Hosanagar, K. (2019). How do recommender systems affect sales diversity? A cross-category investigation via randomized field experiment. *Information Systems Research, 30*(1), 239-259.

[10] Li, Y. M., Chou, C. L., Lin, L. F. (2014). A social recommender mechanism for location-based group commerce. *Information Sciences, 274*, 125-142.

[11] Liu, L., Yang, Q., Hu, X. (2022). Federated learning for recommender systems: A survey. *ACM Computing Surveys, 55*(1), 1-34.

[12] Paradarami, T. K., Bastian, N. D., Wightman, J. L. (2017). A hybrid recommender system using artificial neural networks. *Expert Systems with Applications, 83*, 300-313.

[13] Sun, Z., Wang, C., Hu, W., Feng, J. (2021). Knowledge-aware recommendation: A survey. *IEEE Transactions on Knowledge and Data Engineering, 33*(4), 1328-1348.

[14] Wang, Y., Zhang, H., Chen, X. (2024). Multi-modal recommendation: State-of-the-art and future directions. *Journal of Artificial Intelligence Research, 67*(3), 45-67.

[15] Zhou, Y., Li, X., Wang, J. (2023). Dynamic modeling in recommendation systems. *Information Systems Research, 34*(2), 321-340.