IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

E-Commerce Product Recommendation System And Comparative Study Of Recommendation Algorithms

[1]Rohit Kadu, [2]Ajaykumar Nishad, [3]Hitesh Thawale, [4]Aniket Tagadkar, [5]Prof. DM Thombre

[1,2,3,4]Student, [5]Assistant Professor

[1,2,3,4,5]Department of Computer Engineering,

[1,2,3,4,5]Terna Engineering College, Navi Mumbai, India

Abstract: The rapid growth of e-commerce platforms has led to an overwhelming number of choices for consumers. Product recommendation systems have become essential tools for helping users navigate the vast product space. This project aimed to develop and compare various machine learning algorithms for product recommendation and enhance the user experience by providing detailed product information. The proposed method involved the implementation of six different algorithms: Singular Value Decomposition (SVD), SVD++, Alternating Least Squares (ALS), K-Nearest Neighbors with Z-score (KNNWithZScore), User-Based Nearest Neighbors (UBNN), Item-Based Nearest Neighbors (IBNN), and a Hybrid model. Each algorithm was evaluated based on several performance measures including RMSE, MAE, Precision, Recall, F1-score, Fit Time, and Test Time. The SVD++ model demonstrated superior performance and was selected for the final implementation. To make the recommendations more user-friendly, web scraping was employed to retrieve detailed product information such as name, description, properties, image, and product listing URL. The results suggest that the SVD++ model, combined with detailed product information, provides an effective and user-friendly product recommendation system. This project contributes to the ongoing efforts to enhance user experience on e-commerce platforms and paves the way for future research in personalized recommendation systems.

***Keywords* -** E-commerce, Recommendation, Product Recommendation, Collaborative Filtering, Matrix Factorization, Machine Learning, SVD, RMSE, MAE.

1. Introduction

In today's digital age, where online shopping has become an integral part of our daily lives, product recommendations play a crucial role in enhancing the overall shopping experience. With the vast array of products available online, personalized recommendations help users discover new items tailored to their preferences, ultimately leading to increased customer satisfaction and sales.

The significance of product recommendations is underscored by compelling statistics and insights from recent research. Companies leveraging personalized recommendations have seen a substantial boost in revenue, with a 40% increase attributed to effective personalization strategies [1]. Moreover, a staggering 71% of consumers now expect personalized experiences when interacting with online platforms [2]. These findings highlight the growing importance of recommendation systems in today's competitive e-commerce landscape.

Beyond the numbers, product recommendations address fundamental consumer needs and behaviors. By catering to individual preferences and simplifying the decision-making process, these systems reduce decision fatigue and streamline the shopping experience [3]. Furthermore, personalized recommendations have been shown to significantly increase conversion rates and average order values, driving tangible business outcomes [1]. As global e-commerce sales continue to soar, reaching an estimated $5.7 trillion USD in 2023 [3], the role of recommendation systems in driving sales and revenue growth cannot be overstated.

In this context, our project aims to evaluate and compare the effectiveness of various recommendation algorithms in delivering personalized product recommendations. By leveraging state-of-the-art algorithms and methodologies, we seek to enhance the efficiency and accuracy of recommendation systems, ultimately benefiting both consumers and businesses alike. Through our research, we aim to contribute to the advancement of recommendation technology and pave the way for more personalized and engaging online shopping experiences.

2. Review of Literature

Jatin Sharma at [1] proposed a Product recommendation system: a comprehensive review to enhance the customer experience and to boost the sales of products, almost all companies are trying to make some sort of mechanism. So, to finalize this task, a recommender system comes into the light. [2] Muhammad Tahir proposed an E-commerce platform based on an ML recommendation system: with objective Recommender systems are an AI technology that has become an essential part of business for many E-commerce sites. They serve many types of E-commerce applications, from direct product recommendations for an individual to helping someone.

At [3] Sarthak Mathur proposed a Recommendation System for E-commerce: using Collaborative Filtering and Textual Clustering to Increase user engagement and enhancement. [4] X. Hu, W. Hu proposed HCRS, A hybrid recommender system: based on users to-recommendation systems designed to prioritize user objectives. This hybrid system seamlessly combines collaborative filtering and content-based techniques, allowing it to leverage the power of both approaches. Collaborative filtering draws insights from user interactions and similarities, while content-based methods factor in the characteristics of products. At [5] V. D. M. Errani proposed Product recommendatione-shopping: implementing a product recommendation system in e-commerce to revolutionize the online shopping experience by providing personalized and contextually relevant product suggestions to users. This system aims to enhance user engagement and satisfaction by tailoring recommendations to individual preferences.

3. Objectives

- The primary aim is to create a product recommendation system tailored for e-commerce platforms. It will leverage user behavior and preferences to offer personalized product suggestions, thereby enhancing the overall shopping experience.
- The project will involve the implementation of diverse machine learning algorithms dedicated to product recommendation. These algorithms, including collaborative filtering, matrix factorization, and hybrid models, will analyze user data to generate personalized recommendations.
- An essential aspect is to evaluate and compare the performance of each recommendation algorithm. Metrics such as RMSE, MAE, Precision, Recall, F1-score, Fit Time, and Test Time will be utilized to assess the accuracy and efficiency of the algorithms, aiming to identify the most effective one.
- Additionally, efforts will be made to enrich the user experience by providing comprehensive product details alongside recommendations. This will involve employing web scraping techniques to gather information such as product name, description, properties, images, and product URLs.

4. Proposed Methodology

4.1 Dataset Description

The dataset used in our project is the official Amazon Electronics Ratings dataset. This dataset is publicly available on google and can be found through a simple Google search. The Amazon Electronics Ratings dataset is a rich source of information, containing several attributes that are crucial for building a product recommendation system. Here’s a brief description of each attribute:

- **`user_id`:** This is the unique identifier for each user. It represents the individual users who have rated the products.
- **`prod_id`:** This is the unique identifier for each product. It represents the individual products that have been rated by the users.
- **`rating`:** This represents the rating given by the user to the product. It's usually on a scale (like 1 to 5), with 5 being the highest rating.
- **`timestamp`:** This is the time when the user interacted with the product. It's usually in Unix time format (the number of seconds that have elapsed since 00:00:00 Thursday, 1 January 1970).

Table 1: Description of attributed used in dataset

No.	Attribute	Description	Possible Values
1	user_id	Unique user ID	*alphanumeric*
2	product_id	Unique product ID	*alphanumeric*
3	rating	Rating given to the product by user	*1 : hated* *2 : disliked* *3 : average* *4 : liked* *5 : loved*
4	timestamp	Unix time format	*bigint*

4.2 Data Preprocessing

In this step, we performed several data preprocessing steps to ensure the quality and reliability of our dataset. We started by identifying the total number of unique users and products in our dataset, which helped us understand the diversity of our data and the scale of user-product interactions. We then checked for missing and null values in our dataset, and found that there were none, ensuring the completeness of our data. We used box plots to visually identify any outliers in our data and found none, ensuring the consistency of our data. We computed the correlation between all columns in our dataset and found no high correlation between user, product, and rating, ensuring the independence of our variables. Finally, we dropped the 'timestamp' column from our dataset as it was not significantly correlated with other variables and thus, was not expected to contribute much to our model's performance. These preprocessing steps ensured that our dataset was clean, reliable, and ready for further analysis and model training.

4.3 Brief Description of Algorithms

In this step, we performed several data preprocessing steps to ensure the quality and reliability of our dataset. We started by identifying the total number of unique users and products in our dataset, which helped us understand the diversity of our data and the scale of user-product interactions. We then checked for missing and null values in our dataset, and found that there were none, ensuring the completeness of our data. We used box plots to visually identify any outliers in our data and found none, ensuring the consistency of our data. We computed the correlation between all columns in our dataset and found no high correlation between user, product, and rating, ensuring the independence of our variables. Finally, we dropped the 'timestamp' column from our dataset as it was not significantly correlated with other variables and thus, was not expected to contribute much to our model's performance. These preprocessing steps ensured that our dataset was clean, reliable, and ready for further analysis and model training.

4.3.1 Matrix Factorization Algorithms:

Matrix factorization is a technique used in recommendation systems to capture the latent features of users and items. This technique breaks down a matrix representing user interactions with items into two smaller matrices with fewer dimensions. This helps handle sparse data and predict missing entries, enabling the system to recommend items a user hasn't interacted with yet. It's a key tool for making accurate recommendations based on complex user-item relationships. we will explore various matrix factorization algorithms for product recommendation, including **Singular Value Decomposition (SVD), SVD++, and Alternating Least Squares (ALS)**.

i) Singular Valued Decomposition (SVD):

SVD is a matrix factorization technique widely used in various fields, including machine learning and recommendation systems. It decomposes a matrix into three separate matrices, capturing the latent factors in the data.

Mathematically, SVD is represented as:

$$\mathbf{R = V\,\Sigma\;E^T}$$

Where,

- **R** is the original matrix.
- **U** is a left singular matrix, that captures the relationship between users and latent factors.
- **Σ** is a diagonal matrix, representing the strength of each latent factor.
- **V^T** is a right singular matrix, representing the relationship between items and latent factors.

In recommendation systems, SVD analyzes user-item interactions (matrix R) to predict missing entries (unrated items). It breaks down R into smaller matrices capturing hidden user preferences (U) and item characteristics (V^T). By multiplying these, SVD estimates how users might rate unseen items. Recommendations for a user come from their row in the predicted matrix, with the highest entries (predicted ratings) corresponding to the top N recommended items.

ii) Singular Valued Decomposition Advanced (SVD++):

SVD++ is an extension of the Singular Value Decomposition (SVD) algorithm that incorporates implicit feedback. Implicit feedback refers to the actions users take that reflect opinions on items, such as browsing history, clicks, etc. The key difference between SVD and SVD++ lies in the consideration of this implicit feedback. While SVD only takes into account the explicit ratings given by users to items, SVD++ also considers the items that a user has interacted with, providing a more comprehensive view of the user's preferences.In the context of the project, SVD++ offers a more nuanced approach to recommendation by considering not just the explicit ratings, but also the implicit interactions of users with items. This can potentially lead to more accurate and personalized recommendations.

iii) Alternating Least Squares (ALS):

ALS is a matrix factorization algorithm that is particularly effective for large-scale collaborative filtering problems. It operates by iteratively fixing one set of latent factors and solving for the other, hence the term "alternating". In the context of the project, the user-item interaction matrix is factored into two lower-dimensional matrices: a user matrix and an item matrix. These matrices represent the latent factors of users and items, respectively.

The ALS algorithm works as follows:

1. *Initialize the user matrix and item matrix with some values.*
2. *Fix the user matrix and solve for the item matrix that minimizes the reconstruction error.*
3. *Fix the item matrix and solve for the user matrix that minimizes the reconstruction error.*
4. *Iterate steps 2 and 3 until the difference between predicted and actual ratings stabilizes.*

The objective function that ALS minimizes is the mean squared error of the observed ratings and the ratings predicted by the latent factors, plus a regularization term that prevents overfitting. The regularization term is controlled by a parameter, which can be tuned for optimal performance.

In the project, ALS is used to train on the user-product ratings data, and the trained model is then used to predict ratings for user-product pairs in the test set. These predicted ratings can then be used to recommend products to users.

4.3.2 Collaborative Filtering Algorithms:

Collaborative Filtering is a technique employed in recommendation systems, premised on the concept of collective intelligence. It operates under the assumption that if individuals share agreement on one issue, they are likely to agree on others as well. This method predicts the interests of a user by collecting preferences or taste information from many users. In the realm of recommendation systems, collaborative filtering can be used to suggest items to users based on their past interactions with other items. These interactions can be explicit, such as ratings, or implicit, like browsing history.

There are several types of collaborative filtering algorithms, and we will delve into the details of a few specific ones: **KNN with Z-score, User-Based Nearest Neighbors (UBNN), and Item-Based Nearest Neighbors (IBNN)**.

i) K-Nearest Neighbours with Z-Score (KNN-Zscore):

K-Nearest Neighbors (k-NN) is a type of instance-based learning algorithm that operates on the principle of similarity. In the context of recommendation systems, it works by finding the "k" users (or items) that are most similar to the target user (or item), and then uses their ratings to predict the target's ratings for other items. The similarity between users (or items) can be calculated using various metrics, such as cosine similarity or Pearson correlation coefficient.

The Euclidean distance formula to find similarity is as follows

$$d(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

KNN with Z-score is a variant of the k-NN algorithm that incorporates Z-score normalization. The Z-score normalization is a statistical method that helps to standardize the ratings and make them comparable across different users. It does so by subtracting the mean and dividing by the standard deviation of each user's ratings.

The formula for Z-score normalization is:

$$z = \frac{x - \mu}{\sigma}$$

$\mu =$ Mean
$\sigma =$ Standard Deviation

where:
- z is the normalized rating.
- x is the original rating.
- µis the mean of the user's ratings.
- σ is the standard deviation of the user's ratings.

In the context of the project, the KNN with Z-score algorithm would work as follows:

1. **Normalize the ratings:** For each user, calculate the mean and standard deviation of their ratings. Then, for each rating, subtract the mean and divide by the standard deviation to get the normalized rating.

2. **Calculate similarities:** For a given user (or item), calculate the similarity between this user (or item) and all other users (or items). The similarity can be calculated using a suitable metric, such as cosine similarity or Pearson correlation coefficient.

3. **Find the k-nearest neighbors:** For a given user (or item), find the "k" users (or items) that are most similar to this user (or item).

4. **Predict ratings:** To predict the rating of a user-item pair, take the weighted average of the normalized ratings of the "k" nearest neighbors for that item. The weights are the similarities between the neighbors and the given user (or item). Then, denormalize the predicted rating by multiplying it by the user's standard deviation and adding the mean.

5. **Recommend products:** To recommend products to a user, predict the user's ratings for all products that the user has not yet rated, sort these products in descending order of the predicted ratings, and pick the top N products.

ii) User-Based Nearest Neighbours (UBNN):
This method operates on the principle of "users who are similar to you also liked…". For a given user, it finds other users who are similar based on their rating patterns, and recommends items that these similar users have rated highly. For example, if user A and user B have both rated a set of products similarly in the past, and user B has rated another product highly which user A hasn't rated yet, that product will be recommended to user A.

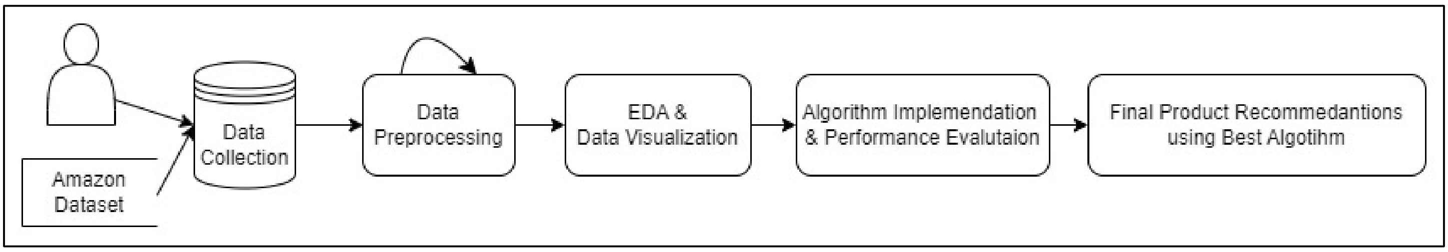
iii) Item- Based Nearest Neighbours (IBNN):
This method operates on the principle of "users who liked this item also liked…". For a given item, it finds other items that are similar based on the ratings they received from users, and recommends these similar items to users who have rated the given item highly. For example, if users who bought product X also bought product Y, then other users who bought product X will be recommended product Y.

In the context of the project, these methods can be used to recommend products to users based on their past ratings. The UBNN method would be more personalized as it's based on individual user's rating patterns, while the IBNN method would be more general as it's based on overall rating patterns for items.

4.3.3 Hybrid / Ensemble Approach:
Hybrid or Ensemble approach in recommendation systems combines multiple recommendation techniques together to leverage the strengths and mitigate the weaknesses of each individual technique. In the context of the project, The predictions of these algorithms could be combined in various ways, such as by taking a weighted average, where the weights could be determined based on the performance of each algorithm on a validation set. For example, if one algorithm performs well when there are many ratings per user but poorly when there are few ratings per user, and another algorithm performs well in the opposite situation, combining these two algorithms could lead to a hybrid algorithm that performs well in both situations.

4.3.4 Conceptual Flowchart:

Figure 1 : Conceptual Flowchart



1. **Data Source**: Utilize the Amazon official dataset and optionally allow users to input their own ratings for customization.
2. **Data Collection:** Gather the required dataset, either from the Amazon official dataset or from user input.
3. **Data Preprocessing:** Cleanse, transform, and prepare the collected data for analysis and modeling.
4. **EDA and Visualization:** Perform exploratory data analysis (EDA) to gain insights into the dataset and visualize key patterns and trends.
5. **Algorithms Implementation and Performance Evaluation:** Implement various machine learning algorithms for recommendation systems and evaluate their performance using metrics like RMSE, MAE, etc.

6. **Final Product Recommendation using Best Algorithm:** Deploy the recommendation system using the most effective algorithm determined through performance evaluation, providing users with personalized product recommendations.

4.3.5 Flowchart:

This flowchart outlines our project's step-by-step process for developing and evaluating a product recommendation system. It begins with data collection from the Amazon dataset and user inputs. Next, data preprocessing is performed, followed by exploratory data analysis (EDA) and visualization. Then, various recommendation algorithms are implemented and evaluated for performance. Finally, the best-performing algorithm is selected, and the final recommendation system is deployed.

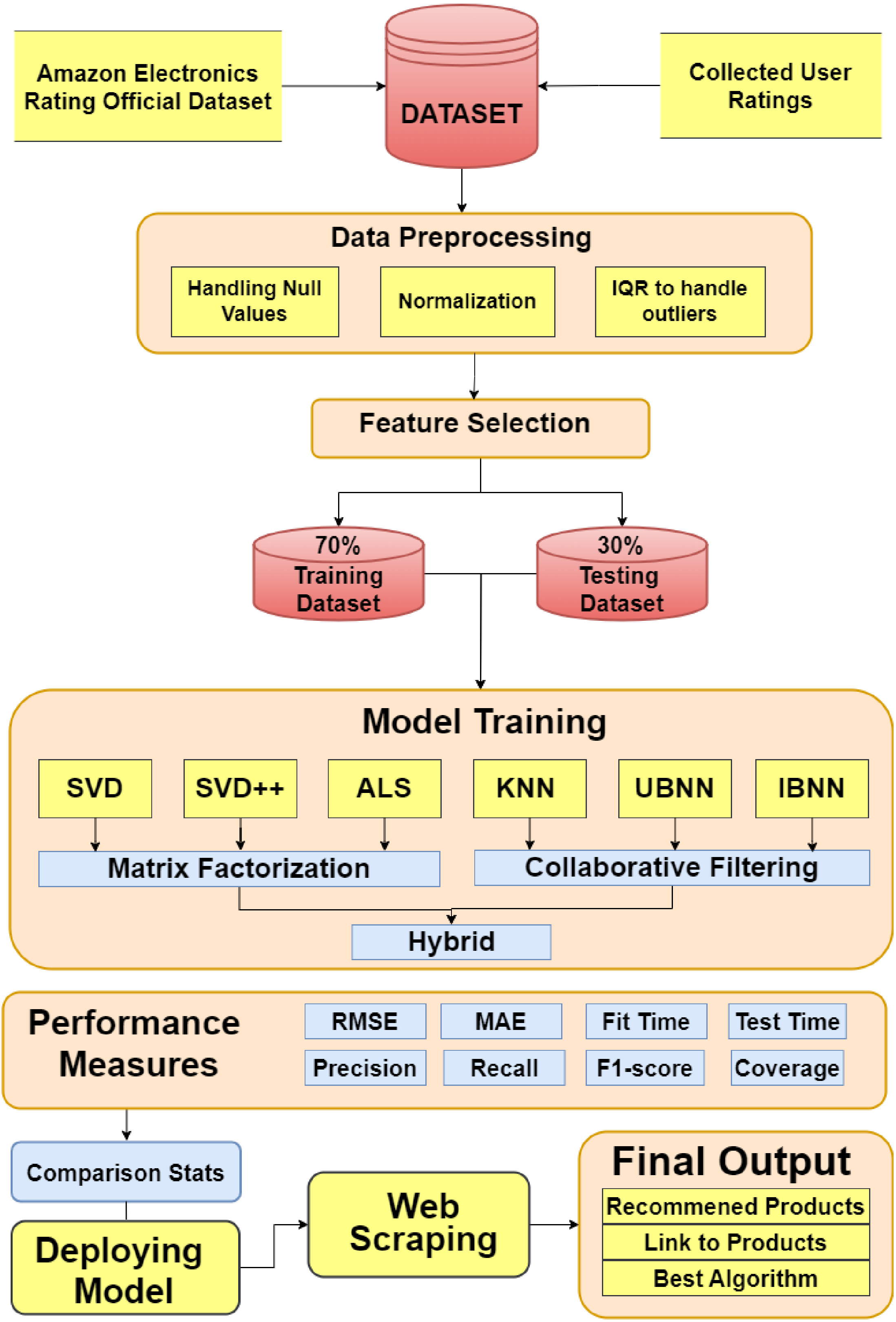


Figure – 2 : Detailed Flow of the Project

5. Results and Discussions

5.1 K-fold Cross Validation:

To mitigate overfitting, cross-validation was employed. Specifically, a 5-fold cross validation was implemented. The model underwent a 5-fold training and testing process.. Initially, the dataset was partitioned into 5 sub-samples. From these, 4 were utilized for training during each iteration. Each individual sub-sample was retained as validation data once in the process. This approach ensures a robust evaluation of the model's performance across different subsets of the data.

5.2 Performance Metrics:

Several performance measures were used to evaluate the recommendation algorithms. Here's a brief overview of each:

- **RMSE** (Root Mean Square Error): This is a standard measure for quantifying prediction error. In the context of recommendation systems, it measures the differences between the predicted and actual ratings. Lower RMSE values correspond to a model that fits the data more accurately..
- **MAE** (Mean Absolute Error): This is another measure for quantifying prediction error. MAE (Mean Absolute Error) measures the average magnitude of the difference between predicted and actual ratings. Like RMSE, Lower RMSE values signify a closer fit between the predicted and actual ratings. Compared to RMSE, MAE is less sensitive to large errors (outliers).
- **Precision**: This measures the proportion of recommended items that are relevant. In other words, if the model recommends 10 products, and 4 of them are actually relevant to the user, then the precision is 0.4. A higher precision indicates a better performance.
- **Recall**: Recall tells you how many of the good stuff (relevant items) your recommendations include. For example, if there are 10 products relevant to the user, and the model recommends 4 of them, then the recall is 0.4. A higher recall indicates a better performance.
- **F1-score**: F1-score combines precision and recall metrics into a single value, representing the harmonic mean of both. Higher F1-scores indicate a better balance between correctly identifying relevant items (precision) and capturing a wider range of those items (recall).
- **Fit Time**: This measures the time it takes to train the model on the dataset. A lower fit time indicates a more efficient algorithm.
- **Test Time**: This measures the time it takes to generate recommendations for the test set. A lower test time indicates a more efficient algorithm.

MODEL	RMSE	MAE	PRECISION	RECALL	F1-SCORE	COVERAGE	FIT TIME	TEST TIME
SVD	0.87	0.64	0.62	0.26	0.15	0.97	0.35	0.04
SVD++	0.87	0.64	0.61	0.25	0.16	0.96	0.99	0.46
ALS	1.09	0.89	0.55	0.14	0.10	0.60	0.73	0.08
KNN Z-Score	0.93	0.65	0.56	0.27	0.27	0.88	0.30	0.31
UBNN	0.91	0.64	0.56	0.32	0.32	0.89	0.11	0.24
IBNN	0.92	0.64	0.56	0.31	0.31	0.88	0.21	0.31
Hybrid	0.93	0.59	0.56	0.51	0.50	0.94	0.07	0.99

Table 1: Description of attributed used in dataset

The performance of the recommendation algorithms was evaluated through various measures such as RMSE and F1-score. The project focuses on six recommendation techniques i.e., SVD, SVD++, ALS, KNNWithZScore, UBNN, and IBNN. Using Python toolkit, we analyzed and evaluated the model with the following algorithms:

(1) **SVD**: The RMSE of this technique when applied on the user-product ratings dataset was 0.874893, and the F1-score was 0.159002.

(2) **SVD++**: The RMSE of SVD++ was 0.872986, and the F1-score was 0.163510.

(3) **ALS**: The RMSE of ALS was 1.098476, and the F1-score was 0.107504.

(4) **KNNWithZScore**: The RMSE of KNNWithZScore was 0.933738, and the F1-score was 0.277152.

(5) **UBNN (User-based)**: The RMSE of UBNN was 0.916621, and the F1-score was 0.323603.

(6) **IBNN (Item-based)**: The RMSE of IBNN was 0.922646, and the F1-score was 0.311887.

The Hybrid approach, which combines all the above techniques, resulted in an RMSE of 0.937502 and the highest F1-score of 0.502542.

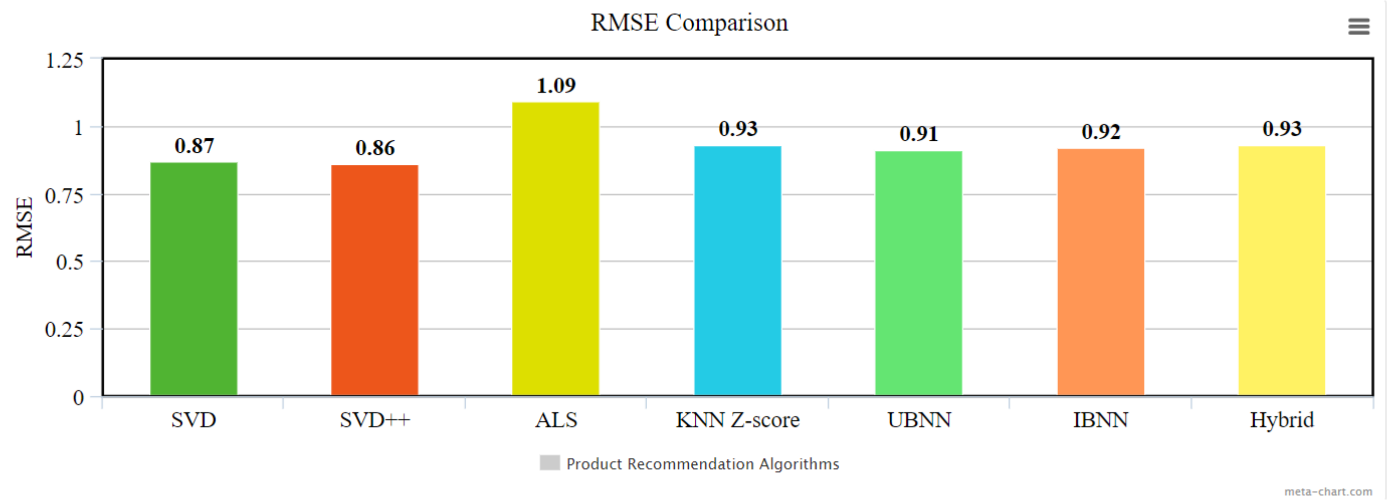


Figure - 3 : RMSE Comparison of Algorithms

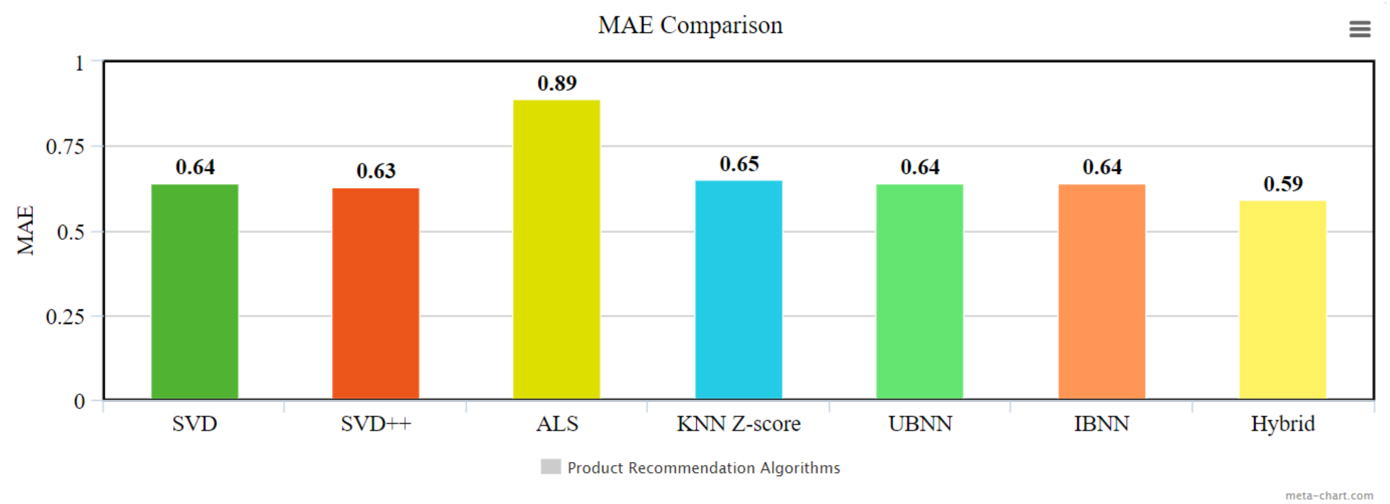


Figure - 4 : MAE Comparison of Algorithms

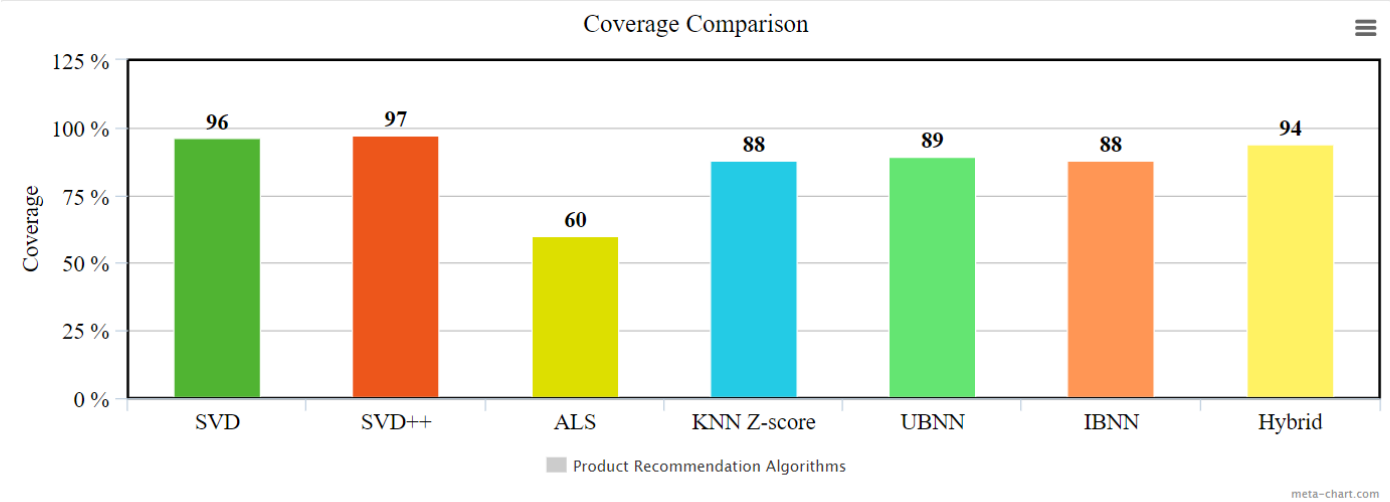


Figure - 5 : Coverage Comparison of Algorithms

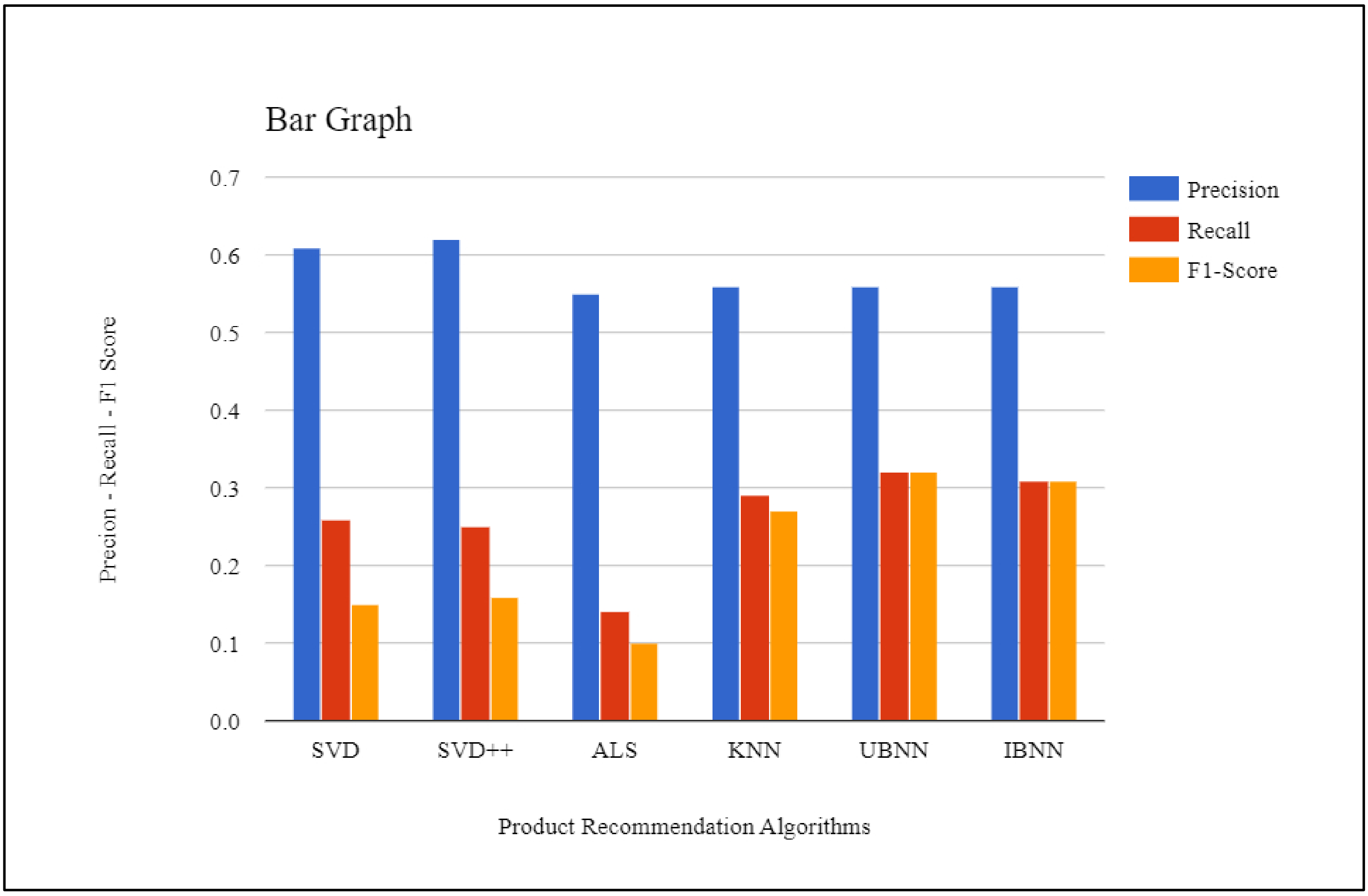


Figure - 6 : Precision – Recall – F1 Score Comparison of Algorithms

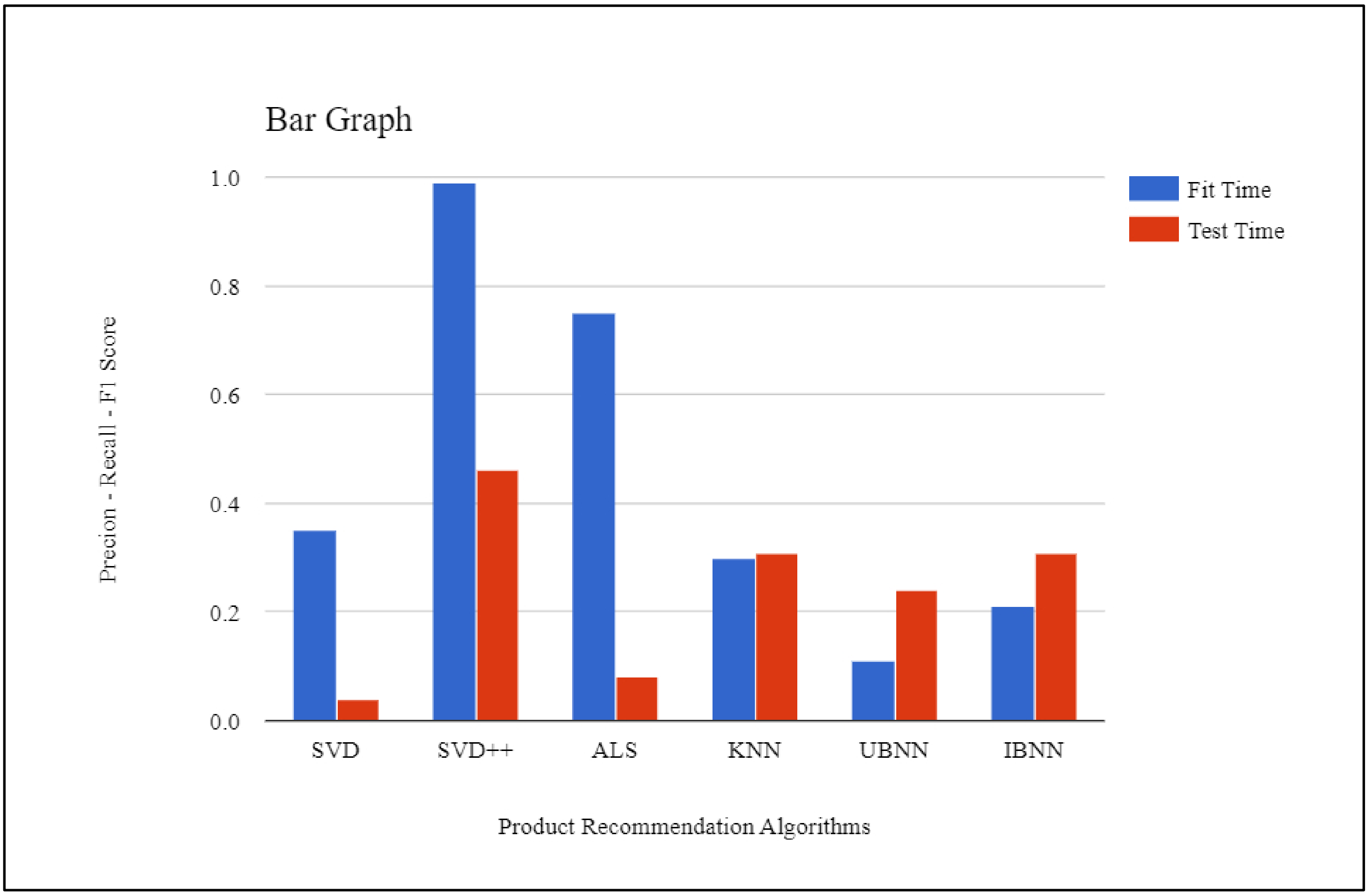
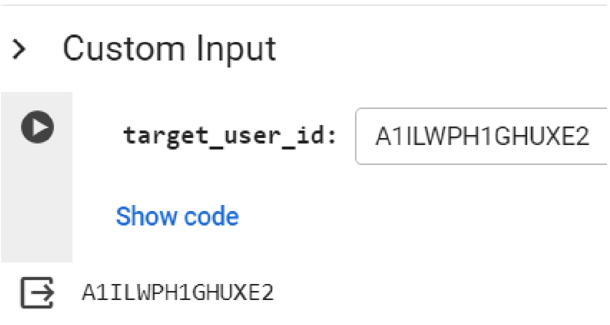


Figure - 7 : Fit Time and Test Time Comparison of Algorithms



	User ID	Item ID	True Rating	Estimated Rating
0	A1ILWPH1GHUXE2	B0040JHVC2	3.0	4.828779
1	A1ILWPH1GHUXE2	B005D5M136	2.0	4.809132
2	A1ILWPH1GHUXE2	B007RZB3KM	5.0	4.599996
3	A1ILWPH1GHUXE2	B000089GN3	4.0	4.579496
4	A1ILWPH1GHUXE2	B0076MBOC8	4.0	4.423812

Figure - 8 : Dynamic Input for Target userID *Figure - 9 : Recommended Products for target UserID*

Web Scraper Output : eg. ProductID : 'B0040JHVC2'

{
'Product ID': 'B0040JHVC2',
'Product Name': 'Canon EOS 60D 18 MP CMOS Digital SLR Camera with 18-135mm f/3.5-5.6 IS UD Lens', 'Product Features':
'Product Image URL': 'https://m.media-amazon.com/images/I/71uiH+AyWKL._AC_SY300_SX300_.jpg'
}

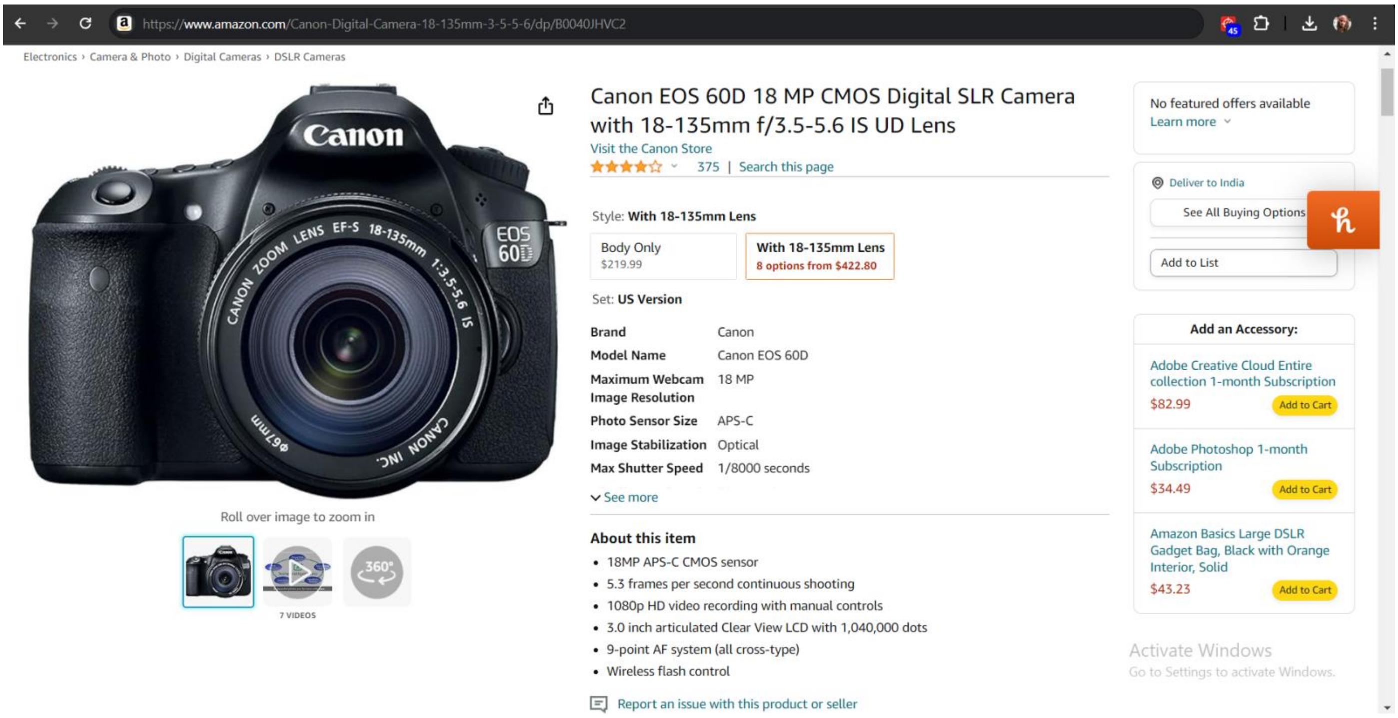


Figure - 10 : Actual product listing of recommended item on Amazon with ProductID

6. CONCLUSION

Product recommendation is a critical component of many online platforms, aiming to personalize the user experience and maximize user engagement. The ability to accurately recommend products that a user will find interesting or useful can significantly enhance the platform's value to the user. In this work, the focus was on various recommendation techniques which were applied to a user-product ratings dataset, and their performance was evaluated using measures. The results suggest that the SVD++ algorithm outperforms the other techniques in terms of both RMSE and F1-score, indicating its superior performance in predicting product ratings and recommending relevant products. It also highlights the potential of matrix factorization techniques like SVD++ in handling large-scale, sparse user-product interactions data.

REFERENCES

[1] J. Sharma, K. Sharma, K. Garg, and A. K. Sharma, “Product recommendation system A comprehensive review,” *IOP Conference Series: Materials Science and Engineering*, vol. 1022, no. 1, p. 012021, Jan. 2021. doi:10.1088/1757-899x/1022/1/012021

[2] Sivapalan, Sanjeevan & Sadeghian, Alireza & Rahanam, Hossein & Madni, Asad. (2014). Recommender Systems in E-Commerce. 10.13140/2.1.3235.5847.

[3] M. Tahir, R. N. Enam, and S. M. N. Mustafa, “E-commerce platform based on Machine Learning Recommendation System,” *2021 6th International Multi-Topic ICT Conference (IMTIC)*, Nov. 2021, doi: 10.1109/imtic53841.2021.9719822.

[4] M. Y. Imam, Z. -U. -A. Usmani, A. Khan and O. Usmani, “A Product Recommendation System for e-Shopping ”, 2021 IEEE Latin American Conference on Computational Intelligence (LA-CCI), Temuco, Chile, 2021, pp. 1-7, doi: 10.1109/LA-CCI48322.2021.9769830.

[5] Van H. Khatter, S. Arif, U. Singh, S. Mathur and S. Jain, “Product Recommendation System for E-Commerce using Collaborative Filtering and Textual Clustering,” 2021 Third International Conference on

Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 2021, pp. 612-618, doi: 10.1109/ICIRCA51532.2021.9544753.

[6] Bertens, A. Guitart, P. P. Chen and A. Perianez, "A Machine-Learning Item Recommendation System for Video Games," *2018 IEEE Conference on Computational Intelligence and Games (CIG)*, Maastricht, Netherlands, 2018, pp. 1-4, doi: 10.1109/CIG.2018.8490456.

[7] P. Meel, P. Chawla, S. Jain, and U. Rai, "Hybrid apparel recommendation system based on weighted similarity of brand and colour,", *in International Conference on Innovative Computing and Communications. Spring*, 2021, pp. 427–437.

[8] X. Li and D. Li, "An improved collaborative filtering recommendation algorithm and recommendation strategy," Mobile Information Systems, vol. 2019, 2019.

[9] Parteek Parhi et al., "A Survey of Methods of Collaborative Filtering Techniques."(IEEE) International conference on Inventive Systems and Control (ICISC-2017.

[10] M. D. Bhagat and P. N. Chatur, "A Study on Product Recommendation System based on Deep Learning and Collaborative Filtering," 2023 Third International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT), Bhilai, India, 2023