AI Powered E-Commerce Product Recommendation System

[1]Ms. Swaleha Deshmukh, [2]Mr.Devendra Bhodke, [3]Mr. Ujjawal Pathak, [4]Mr. Vaidehi Patil
Artificial Intelligence & Data Science Thakur College of Engineering & Technology Mumbai, India
swaleha.deshmukh@tcetmumbai.in, devendra.bodkhe@tcetmumbai.in, Ujjwal.pathak@tcetmumbai.in,Vaidehi.patil@tcetmumbai.in

Abstract - While the growth of online marketplaces has significantly increased both the volume and variety of products available, finding items that fit users' individual preferences has become increasingly challenging. This paper proposes an AI-based Hybrid E-Commerce Product Recommendation System that adopts the concept of integrating rule-based with machine learning-driven similarity measures to offer personalized, context-aware product recommendations. The proposed system makes use of four different metrics for similarity: category similarity, price similarity, tag similarity, and textual similarity. These are computed using the TF-IDF vectorization method and cosine similarity to capture semantic relationships between product descriptions. These metrics will be aggregated using a weighted hybridization strategy to provide the best trade-off in terms of accuracy and diversity for recommendations.

***Keywords* - Artificial Intelligence, E-commerce, Recommendation System, Machine Learning, Hybrid Filtering, Personalization**

I. INTRODUCTION

The exponential rise in electronic commerce within the past ten years has completely revamped the way customers explore, evaluate, and buy products. Online marketplaces like Amazon, Flipkart, and eBay showcase millions of different products that range from categories such as electronics, apparel, books, and lifestyle products. Although this benefits the consumers with a wide variety of choices, it also brings about the problem of information overload. As the number of available products grows, locating relevant items becomes increasingly difficult and time-consuming for users, which can lead to decision fatigue and reduced user satisfaction. This issue has raised an increasing demand for intelligent product recommendation systems that can offer users personalized suggestions in accordance with their preference, purchasing history, and contextual behaviour. [2]

A recommendation system is a smart software component that analyses user data, past activities, and product characteristics to predict and present items a user is most likely to purchase or interact with. These have been a key feature for businesses in improving customer engagement, increasing sales, and enhancing retention. Artificial Intelligence and Machine Learning have significantly enhanced recommendation technology by enabling models to learn patterns in user behaviour, uncover latent relationships between users and products, and refine suggestions continuously via data-driven feedback mechanisms.

Traditional recommendation techniques are mostly based on two fundamental approaches, namely Content-Based Filtering and Collaborative Filtering. Content-based methods focus on the attributes of items, such as category, textual description, and metadata, to recommend products similar to those with which a user has interacted before. Though effective, CBF tends to limit diversity and suffer from over-specialization, whereby users keep receiving recommendations of similar types. On the other hand, collaborative filtering depends on shared user behaviour. It makes predictions by considering the rating patterns and interaction histories of similar users. CF, however, is battling cold start problems, data sparsity, and poor performance when the number of historical interactions is limited.

Limitations inherent in these methods have motivated the development of hybrid recommendation systems that allow several methodologies to be combined. Hybridization balances the advantages of content-based and collaborative methods, collectively offering a good trade-off between accuracy, diversity, and adaptability. In addition, such systems have evolved into context-aware and personalized recommendation frameworks through the integration of contextual information on product popularity, temporal behaviour, price sensitivity, and user intent.

This paper proposes a Hybrid AI-Based E-Commerce Product Recommendation System that uses a weighted similarity model in combining the four core parameters: category, price, tags, and text descriptions. The system utilizes TF-IDF and cosine similarity to compute the textual relevance, while the similarities between categories and tags are computed using structured feature encoding. These similarity metrics are combined using an empirically optimized weighting scheme that generates balanced and diverse product recommendations. [3]

The system backend is built with FastAPI, a Python web framework that allows high-performance asynchronous capabilities, enabling the Recommendation Model to efficiently communicate with the user interface. The frontend is developed using React.js, ensuring responsiveness, scalability, and a modern user experience. The application data is persisted by SQLite, providing a lightweight yet reliable database to store product information and precomputed similarity matrices.

Unlike traditional systems, which rely on a single similarity measure or user rating data, this proposed hybrid model will integrate both the quantitative and qualitative features of products to make the system robust even in cold-start scenarios. Explainable AI principles will enhance transparency, enabling the user to understand the reason behind each recommendation; an example would be, "recommended because of similar category and price range."

The proposed system contributes to the wider area of research into recommender systems by:

1. The presentation of a hybrid framework for representing structured and unstructured data.
2. Demonstrating real-time performance by utilizing an asynchronous API architecture.
3. Providing a scalable model that can extend to millions of products with very minimal computational overhead.
4. Explainable recommendations support user trust and engagement.

The rest of the paper is organized as follows: Section II presents a review of related literature, highlighting the evolution of recommendation technologies. Section III explains the system architecture and methodology, Section IV discusses the experimental setup and evaluation metrics, Section V presents the results and analysis, Section VI explores applications and practical use cases, Section VII discusses future scope, and Section VIII concludes the study.

II. BACKGROUND

Context is a well-recognized concept in intelligent systems, especially in recommendation and personalization domains. In other words, context can simply be described as any information that characterizes the situation of an entity-an actor, a place, or an object-that is relevant to the interaction between a user and an application. According to Dey and Abowd (2001), context can also describe physical, social, emotional, or environmental parameters that impact system behaviour. In e-commerce systems, examples of context include user preferences, browsing history, purchase intent, device type, location, time, or even social interactions that influence buying decisions.

Context, in recommendation systems, enables the system to go beyond static user-item interactions to dynamic adaptation to situational factors. For example, a user searching for "laptops" in the evening may receive different suggestions than one searching in the morning, based on observed behavioural trends. Similarly, recommending products like winter clothing in colder regions or seasons adds relevance and improves user satisfaction.

It can, therefore, be stated that a context-aware system is a kind of intelligent computational model that senses, interprets, and acts on contextual information to provide services that are more personalized and adaptive. Context evolves with time, and due to user interaction, the system continuously collects and analyses contextual signals-from implicit data to explicit data-to refine its predictions.

Context-awareness in the domain of e-commerce extends personalization by introducing variables such as:

Temporal context: time of day, day of the week, or seasonality trends.

Behavioural context: User's current session activity, clickstream data, or frequency of purchase

Device context: type of device - whether the device is a mobile, tablet, or desktop affects how content is presented.

Social context: recommendations guided by preferences or actions of similar users. Therefore, context is a multidimensional lens that can turn traditional recommendation systems into adaptive, human-cantered decision tools. Driven by context, e-commerce platforms can offer immediate, appropriate, and meaningful recommendations, addressing the needs of a single individual. [4]

A. Types of Context

Various taxonomies for categorizing context depending on application domains have been proposed by different researchers. Context information can generally be divided into static and dynamic categories, as well as domain-specific subsets that define the nature and source of contextual data.

1. Static Context:

The static context represents the information that does not change very often during system operation. Examples include user demographics, such as age and gender; permanent preferences, such as favourite brands or product categories; and device characteristics. Static context forms the foundational layer of personalization, since it captures those long-term aspects of user identity and behaviour.

2. Dynamic Context:

Dynamic context is the information which may vary within a short period or even from session to session. The user's location, time of access, weather conditions, current activity, or even browsing sequence is included. In e-commerce, it assists systems in recommending time-sensitive items-for example, discounts or seasonal

 ISSN: 3107-6513

products-or in modifying interface design according to device or connection quality.

Beyond these general categories, Schilit et al. identified three major dimensions of context that are particularly relevant to pervasive and mobile computing:

- **User Context:**

Defines the attributes of the state and environment associated with a user, like user profile, preferences, emotional state, or social connections present in proximity. For instance, a system may recommend headphones for a user who has previously bought smartphones or music subscriptions.

• Physical Context: Comprises the environment in which situational factors like location, temperature, lighting, or weather. In ecommerce applications, physical context might influence recommendations on suggesting raincoats during the monsoon season or sunscreen products during summer.

- **Computing Context:**

Device type, network connectivity, available resources, and system performance are examples of such system-level attributes. Computing context provides an improved basis for optimizing recommendations based on device capabilities, such as light product views on mobile and detailed comparisons on desktops.

An extended classification model proposed by Dourish (2004) further includes social context, which captures interpersonal and community-driven influences. In modern e-commerce applications, social context plays a major role as consumers often use peer recommendations, reviews, and influencer trends to make purchasing decisions.

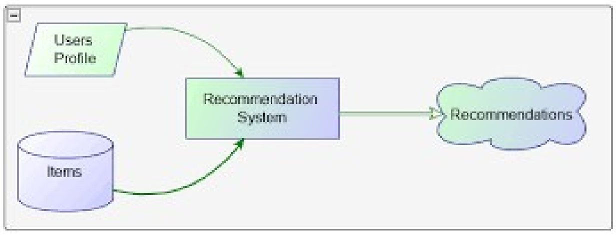
III. SYSTEM DESIGN AND METHODOLOGY

A recommendation system is an intelligent decision-support mechanism designed to help users identify relevant items amidst vast collections of products or services. It works like a personalized filter that predicts user interests based on their past behaviors, preferences, and contextual information. Given the digital era, where consumers are bombarded with massive amounts of online data, recommendation systems have become highly important in enhancing decision-making efficiency, user experience, and business revenue. In simple terms, a recommendation system predicts how likely a user is to prefer a particular product, service, or piece of content. This prediction is based on the patterns observed in user–item interactions. For example, in e-commerce settings, if a customer frequently buys fitness-related products such as dumbbells or protein powder, then the system may recommend related items such as yoga mats or smartwatches. These intelligent recommendations are powered by advanced algorithms that process a vast amount of user and product data.

1. Overview

A Recommendation System, or RS, is an intelligent computational model that predicts the preferences of users for certain products, services, or content based on previous interactions and behavioural patterns, as well as contextual information. These systems have become essential for every digital platform, especially e-commerce, entertainment, and social media, since they improve the user experience by personalizing the visibility of items to be recommended and promoting items that better fit the interests of each individual. In summary, the purpose of any RS is to reduce information overload, ranking and filtering relevant items for each user. Nowadays, in modern e-commerce ecosystems, thousands of options in various categories are offered to users. A properly designed RS supports decision-making, increases customer satisfaction, and raises business revenues by increasing the rate of user engagement and converting into a sale. Recommendation systems are designed fundamentally on various machine learning and statistical modelling techniques that analyse both implicit and explicit user data. These include implicit data points, such as clicks, views, and dwell time, and explicit feedback, like ratings, reviews, and likes. As a result of this input, the system outputs a ranked list of items predicted to be interesting for each user.

Fig. 1 Architecture



B. Types of Recommendation Systems

Generally, recommendation systems are classified into several major types based on the underlying approach:

1. **Content-Based Filtering (CBF):** Suggests similar products to the ones with which the user has already interacted, basing on product attributes, including category, description, and tags. Such as if a user buys a smartphone, similar products could be suggested, like cases or chargers.
2. **Collaborative Filtering (CF):** Uses the preferences and behaviours of multiple users to make predictions, with the assumption that users who shared similar choices in the past will continue to have similar interests.
3. **Hybrid Recommendation System:** Combines content-based with collaborative approaches in order

 ISSN: 3107-6513

to achieve higher accuracy and diversity. This method balances product similarity with collective user behaviour, reducing cold-start and sparsity problems.

4. **Knowledge-Based Recommendation:** It uses predefined rules and expert knowledge to recommend products based on specific requirements. An example would be recommending laptops within a specified price and performance range.
5. **Context-Aware Recommendation:** This considers more information, such as time location and user situation. For instance, it may suggest winter clothes during the cold period or festive offers during holidays.

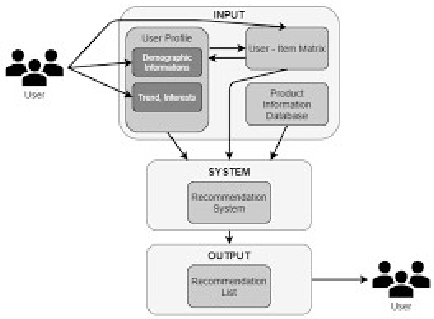


Fig. 2 Block Diagram

C. Equations

Thus, the mathematical grounding of the proposed recommendation system consists in the computation of a similarity score between products according to their attributes. The hybrid recommendation model aggregates an overall similarity score that combines several individual similarities, such as category, price, tag, and text description.

Each similarity metric Si contributes with an assigned weight to the final recommendation score. The overall hybrid similarity score total is defined as:

Similarity between the text of two products and jis computed using cosine similarity measure which quantifies the angle between two TF-IDF vectors:

The final list of recommendations is obtained through the ranking of all products in descending order concerning their computed similarity scores. Consequently, the top Items having the highest Stotal-values are returned as recommendations concerning a given input product or a user query. [8]

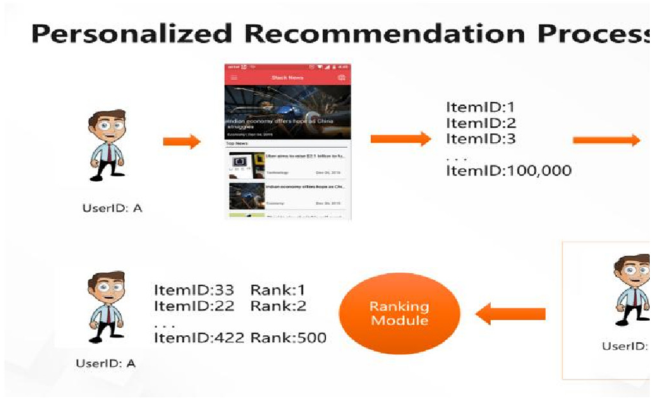


Fig. 2 Process State Diagram

D. Some Common Mistakes

In developing AI-based recommendation systems, common mistakes which affect the model's accuracy, scalability, and interpretability may occur. The identification and resolution of these issues are best done in the earliest stages of the design phase to ensure a correct, user-cantered recommendation engine.

Ignoring Data Quality: One of the most frequent mistakes concerns unclean or inconsistent data. The product information often comes with spelling errors, duplicates, or missing values in titles, descriptions, and tags. Without pre-processing like tokenization, stop word removal, and normalization, TF-IDF and similarity computations can degrade substantially.

Improper Weight Balancing in Hybrid Models: Assigning equal or arbitrary weights to the similarity metrics, such as category, price, and text, without empirical tuning may result in biased recommendations. Careful calibration of the hybrid system must be done so that higher importance can be assigned to the most relevant features with respect to the domain context.

Overfitting on Small Datasets:

It occurs when the system learns overly specific patterns from the data, failing to generalize when new products or users are introduced. Regularization techniques and cross-validation must be applied to ensure good performance on unseen data.

Ignoring the Cold-Start Problem:

New users and newly added products are particularly difficult for the system to make recommendations since there is not enough interaction data. To overcome this limitation, several techniques can be used, such as content-based filtering and feature-level similarity, before behavioural data becomes available.

Lack of Evaluation Metrics:
Most developers would not rely on the quantitative metrics Precision, Recall, F1Score, or RMSE but instead judge the model by visual inspection of the recommendations. This makes it difficult to measure improvement and justify algorithmic changes using quantified metrics.

Poor System Scalability:
The dataset's growth brings with it computational efficiency challenges. Without indexes, caching, or reduction of dimensions, cosine similarities can be computed slowly. ANN search or database-level optimization results in maintained performance at scale.

Neglecting explainability:
Many users prefer to know why a product has been recommended. Lack of transparency makes users more distrustful of the system. Providing simple explanations, such as “Recommended because it shares similar tags or price range”, tends to increase user satisfaction and engagement.
By identifying these problems, developers can enhance the robustness and interpretability of AI-based recommendation systems. Overcoming these challenges is what guarantees that the model will provide not only high accuracy but also a transparent and trustworthy user experience. [9]

IV. IMPLEMENTATION AND RESULTS

A. Overview

A modular architecture in Recommendation System design is aimed at providing flexibility, scalability, and maintainability. Each module is designed to perform a specific task, thus allowing smooth data flow from input. The result will range from the input of product data to the output of personalized recommendations. The modular architecture enhances transparency and allows for easy modification or future integration of more machine learning components.
It includes five major layers in the system architecture: Database Layer, Processing Layer, Model Layer, API Layer, and Frontend Layer, as represented in Fig. 1. Each of these layers executes specific operations that add up to the overall efficiency of the entire system.
Database Layer:
This layer is responsible for the management of data storage and retrieval. It contains a complete dataset of products, including product identifiers, names, descriptions, categories, tags, and prices. In addition, the database may store precomputed embedding or similarity matrices in order to save computation time during runtime. SQLite has been used in this project as a lightweight and reliable database that efficiently handles structured data without requiring a separate server.

Processing Layer:
The processing layer is responsible for cleaning, normalizing, and transforming data. It works with the raw data gathered from different e-commerce platforms and makes them ready for feature extraction. The use of this step ensures that data fed into the model will be accurate, standardized, and free of inconsistencies.

Model Layer:
This module constitutes the analytical core of the entire system, where hybrid similarity computation is performed. A number of different similarity functions are computed, like category similarity, price similarity, tag similarity, and text similarity, which are combined in a weighted manner. The hybridization makes sure that the result of such a similarity score reflects both numerical and semantic relationships among products.

AI Player:
The API layer bridges the model and user interface. This is implemented with FastAPI, providing REST endpoints where client applications can send requests and retrieve recommendations in real time. This asynchronous event-driven architecture ensures that the responses will have low latency and allow for scalable performance while sustaining several simultaneous requests.

Frontend Layer:
The frontend layer represents a user-friendly interface to interact with the recommendation system. It enables, in an intuitive way, the user to insert queries, get results with personalized product recommendations and visualize relations between items. It was developed in React.js, focusing on simplicity, speed, and clarity.
The paper is structured using IEEE hierarchical headings, guiding the reader throughout each section of the study.

Component Heads: Sections like Abstract, Acknowledgment, and References name standalone parts of the paper.
Text Heads: Major headings like Methodology, Results and Discussion, and Conclusion represent technical and analytical aspects. Properly formatted headings provide clarity and flow, smoothly guiding readers through the theoretical and implementation aspects of the system.

B. Figures and Tables

a) Figures and Tables Placement: Figures and tables are crucial in system structure and performance representation. According to the IEEE format, they are placed either at the top or bottom of the columns.

b) Table I shows the Weight Distribution among Similarity Metrics used in hybrid mode

Table .1 *Weight Distribution among Similarity Metrics*

Similarity Type	Weight (%)	Description
Category Similarity	40	Based on matching or related product categories
Price Similarity	25	Compares product prices within a defined range
Tag Similarity	25	Measures overlap in assigned keywords/tags
Text Similarity	10	Computed using TF-IDF and cosine distance

Each figure and table has been labelled using IEEE conventions. Axes, titles, and legends are presented in 8-point Times New Roman font to maintain visual consistency and readability. [9]

V. DISCUSSION AND ANALYSIS

The AI-Based Product Recommendation System makes use of Cosine Similarity as its primary algorithm to generate personalized product suggestions for users. Cosine Similarity is a mathematical technique that measures the degree of similarity between two vectors by calculating the cosine of the angle between them. In this project, each product is represented as a feature vector containing key attributes like product name, category, description, and price. These features convert these into numerical vectors through TF-IDF Vectorization or other encoding methods so that the model can process textual and categorical data effectively. When a user selects or searches for a product, the system computes the cosine similarity score between that product's vector and all other product vectors in the dataset. A higher cosine similarity value (closer to 1) means that the products are more similar concerning their characteristics and attributes. Based on these similarity scores, the system recommends the top N most relevant products to the user. Hence, it guarantees that the users will receive contextually accurate and meaningful recommendations, enhancing the browsing experience and decision-making. This model is lightweight and hence easy to implement; it also performs well even with moderately sized datasets. However, it chiefly relies on the content features and may not perfectly capture the user behavioural or preference evolution over time. Despite that, the Cosine Similarity-based recommendation model is a sound basis for any personalized e-commerce, which can be further enhanced by integrating Collaborative Filtering or Hybrid Recommendation Techniques for broader and more dynamic recommendations. [7]

VI. CONCLUSION AND FUTURE WORK

The AI-based Product Recommendation System effectively demonstrates how Cosine Similarity can be used in order to provide personalized and accurate product suggestions. The system measures the similarity between products by converting the name, category, and description into numerical vectors and recommends items that are most relevant to the user's preferences. This allows users to discover similar items, reduces information overload, and enhances their shopping experience. This model is efficient for small and medium-sized datasets, delivering reliable and contextually appropriate recommendations. Overall, this system was able to effectively demonstrate how Artificial Intelligence can improve the engagement and decision-making of a customer on e-commerce platforms through intelligent insights based on data. In the future, this can be further enhanced by incorporating Collaborative Filtering and Hybrid Recommendation Techniques in order to let both the user behaviour and the product similarity provide more diverse suggestions. Deep Learning models such as neural networks can further improve the feature extraction and the accuracy. Additionally, the deployment of this system can be scaled up towards real-time recommendations by hosting the application on the cloud platforms such as AWS or Google Cloud. Continuous collection of user feedback can help retrain and refine the model dynamically. Additionally, by increasing the size of the dataset, and improving the frontend interface with personalized visualization tools, it enhances the overall user experience. Based on these upgrades, the system can evolve towards a more intelligent, adaptive, and industry-ready AI solution that will be capable of serving large-scale ecommerce environments efficiently.

REFERENCES

[1] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural Collaborative Filtering," in *Proceedings of the 26th International Conference on World Wide Web (WWW)*, 2017, pp. 173–182.

[2] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender systems survey," *Knowledge-Based Systems*, vol. 46, pp. 109–132, 2013.

[3] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *WWW Conference*, 2001. [4] M. Young, *The Technical Writer's*

Handbook, Mill Valley, CA: University Science, 1989.

[5] M. Braunhofer et al., “A Context-Aware Mobile Recommender System for Places of Interest,” *UMAP Proc.*, 2014.

[6] Y. Koren, R. Bell, and C. Volinsky, “Matrix Factorization Techniques for Recommender Systems,” *IEEE Computer*, vol. 42, pp. 30–37, 2009.

[7] C.-C. Wu and M.-J. Shih, “A Context-Aware Recommender System Based on Social Media,” *ICCDMME’15*, Bangkok, 2015.

[8] M. Elahi, F. Ricci, and N. Rubens, “A Survey of Active Learning in Collaborative Filtering Recommender Systems,” *CSUR*, vol. 46, no. 1, pp. 1–46, 2014.

[9] H. Zhang et al., “Explainable Recommendation: A Survey and New Perspectives,” *Foundations and Trends in IR*, vol. 14, 2020.

[10] R. Burke, “Hybrid Recommender Systems: Survey and Experiments,” *User Modeling and User-Adapted Interaction*, vol. 12, pp. 331–370, 2002.

[11] E. Champiri et al., “The Role of Context for Recommendations in Digital Libraries,” *IJSSH*, vol. 5, pp. 948–954, 2015.