IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

An AI-Driven Explainable Product Recommendation System Using Large Language Models and Semantic Search

[1]Samruddhi Maheshkumar Aher, [2]Harshali Rajendra Bagul, [3]Diksha Ravindra Nirbhavane, [4]Ashwini Nandu Pawar, [5]Puneet Eknath Patel

[1]Student, [2]Student, [3]Student, [4]Student, [5]Guide

[1]Department of Information Technology,

[1]MET's Institute of Engineering, Nashik, India

Abstract: The rapid expansion of online shopping platforms has significantly increased the variety of products available to users, making the decision-making process more complex and time-consuming. Conventional recommendation systems, such as collaborative filtering and content-based approaches, often struggle to accurately interpret user intent expressed in natural language and typically lack transparency in their outputs.

This paper presents a novel design of an AI-driven explain- able recommendation system that integrates Large Language Models (LLMs), semantic similarity search using FAISS, and interpretable machine learning techniques such as SHAP. The proposed system is capable of processing user queries in natu- ral language, extracting meaningful preferences, and retrieving contextually relevant products from large datasets. Additionally, it generates clear and human-understandable explanations for each recommendation.

By combining advanced language understanding with explain- ability and efficient retrieval mechanisms, the system improves recommendation accuracy, enhances transparency, and increases user trust. The proposed framework demonstrates the potential of integrating modern AI techniques to build intelligent and user- centric recommendation systems.

Index Terms - Explainable Artificial Intelligence (XAI), Recommendation Systems, Large Language Models (LLMs), Semantic Search, FAISS, SHAP, Natural Language Processing (NLP), E-commerce, Personalized Recommendations, Information Retrieval

I. INTRODUCTION

With the continuous growth of digital commerce, recommendation systems have become essential tools for assisting users in discovering relevant products from vast online catalogs. E-commerce platforms such as Amazon, Flipkart, and Meesho rely heavily on these systems to improve user engagement, personalize experiences, and increase sales. However, the increasing complexity and volume of available products make it challenging for traditional recommendation techniques to meet evolving user expectations.

Existing approaches, including collaborative filtering and content-based filtering, have been widely used but exhibit notable limitations. Collaborative filtering depends heavily on historical user data and often fails in scenarios involving new users or products, commonly referred to as the cold-start problem. On the other hand, content-based methods are restricted by their reliance on predefined features and lack the ability to interpret complex user queries expressed in natural language.

Recent advancements in artificial intelligence, particularly Large Language Models (LLMs), have introduced new possibilities for understanding user intent more effectively. These models are capable of analyzing unstructured text input and extracting meaningful information such as preferences, constraints, and contextual requirements. Despite these improvements, many modern recommendation systems still operate as opaque models, providing results without offering clear reasoning behind their suggestions.

Explainable Artificial Intelligence (XAI) addresses this issue by enabling systems to present interpretable insights into their decision-making processes. Techniques such as SHAP (SHapley Additive exPlanations) allow identification of feature contributions, making recommendations more transparent and trustworthy for users.

In this paper, we propose an AI-driven explainable recommendation framework that integrates LLM-based intent extraction, FAISS-based semantic search, and SHAP-based explanation generation. The objective is to develop a system that not only delivers accurate and context-aware recommendations but also provides meaningful explanations to enhance user confidence. The proposed approach aims to bridge the gap between performance and interpretability in modern recommendation systems.

II. Literature Review

The advancement of recommendation systems has been significantly influenced by developments in natural language processing, machine learning, and explainable artificial intelligence. This section presents a critical overview of existing approaches and identifies key limitations that motivate the proposed work.

A. Language Models in Recommendation:

Recent progress in Large Language Models (LLMs) has enabled systems to interpret user queries expressed in natural language with greater accuracy. Unlike traditional keyword-based systems, LLMs can capture contextual meaning and extract relevant attributes such as product category, preferences, and constraints. This capability enhances the interaction between users and recommendation systems by allowing more flexible query formulation. However, many existing implementations focus primarily on improving retrieval performance and often lack mechanisms to provide clear explanations for the generated recommendations.

B. Explainability in Recommendation System:

The importance of interpretability in recommendation systems has grown due to increasing user demand for transparency. Explainable AI techniques aim to make model decisions understandable by identifying the influence of various input features. Methods such as SHAP and LIME have been widely used to provide feature-level explanations. Although these approaches improve trust, they are frequently applied as standalone modules and are not always integrated with advanced query understanding mechanisms.

C. Semantic Retrieval Techniques

Semantic search has emerged as an effective alternative to traditional information retrieval methods. By representing data using vector embeddings, semantic search captures contextual similarity between queries and items. Tools such as FAISS enable efficient similarity search over large-scale datasets, making them suitable for real-time recommendation systems.

Despite their effectiveness, semantic retrieval methods are often used independently and may not incorporate user-specific constraints or preferences effectively.

D. Context-Aware Recommendation Approaches

Context-aware systems enhance recommendation quality by incorporating additional information such as user behavior, location, and historical interactions. These systems aim to provide more personalized suggestions by adapting to dynamic user contexts. However, integrating contextual awareness with explainability and semantic understanding remains a challenging task in existing research.

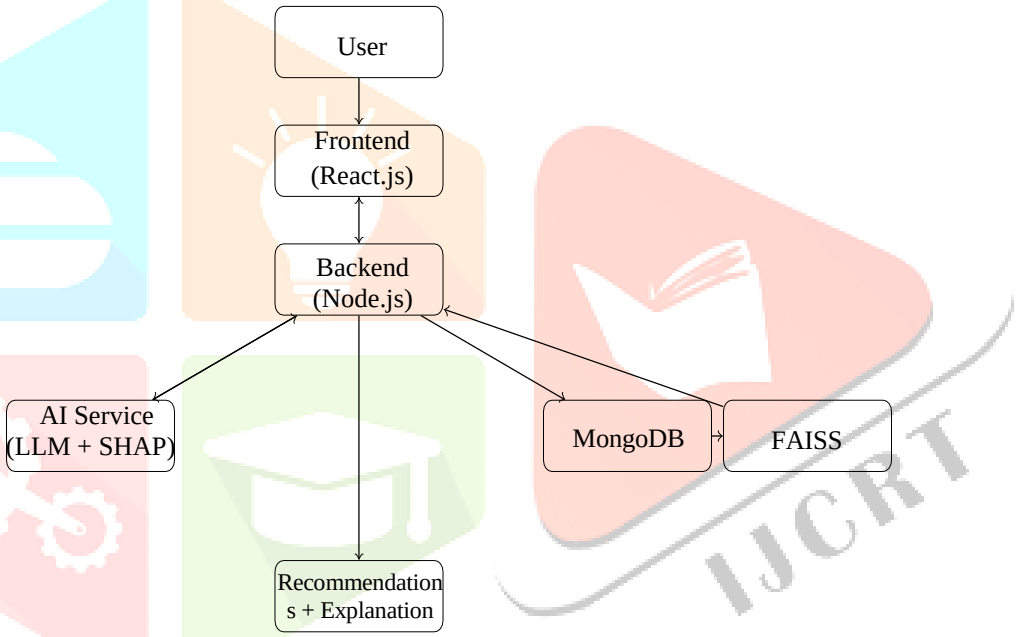
E. Research Gap

From the analysis of existing studies, it is evident that:

- LLM-based systems improve query understanding but often lack interpretability
- Explainable models enhance transparency but may not effectively handle complex queries
- Semantic search improves relevance but is frequently used in isolation

III. System Architecture

The proposed system follows a modular and scalable architecture designed to process natural language queries and generate explainable product recommendations. The architecture consists of multiple layers including frontend, backend, AI processing, database, and semantic search components.



A. User Interface Layer:

The user interface layer acts as the entry point of the system, allowing users to interact through a web-based platform. It is developed using modern frontend technologies and supports natural language input, enabling users to express their requirements in a flexible and intuitive manner. The interface also displays recommended products along with their attributes, price comparisons, and explanation insights. Emphasis is placed on usability and responsiveness to ensure a seamless user experience.

B. Backend Processing Layer:

The backend layer serves as the central controller that manages communication between all system components. It handles incoming requests from the frontend, validates user inputs, and coordinates with the AI module and database systems. The backend is responsible for aggregating outputs from different modules and generating a unified response. It also ensures efficient data flow, error handling, and system reliability.

C. AI and Intelligence Layer:

The AI layer is the core component responsible for understanding user queries and generating intelligent recommendations. It utilizes Large Language Models (LLMs) to extract structured information such as product category, price range, brand preference, and desired features from unstructured text queries. This

layer also incorporates explainable AI techniques to generate interpretable insights. By combining contextual understanding with feature-level analysis, the system improves both accuracy and transparency.

D. Data Storage Layer:

The data storage layer maintains a comprehensive collection of product information, including specifications, prices, ratings, and platform details. A NoSQL database is used to handle large volumes of semi-structured data efficiently. Indexing techniques are applied to improve query performance and enable fast retrieval of relevant records based on user constraints.

E. Semantic Search Layer:

The semantic search layer enhances recommendation quality by identifying contextually relevant products. Instead of relying solely on keyword matching, this layer uses vector embeddings to represent products and queries in a high-dimensional space. A similarity search mechanism retrieves items that closely match the meaning of the user query, thereby improving the relevance of recommendations.

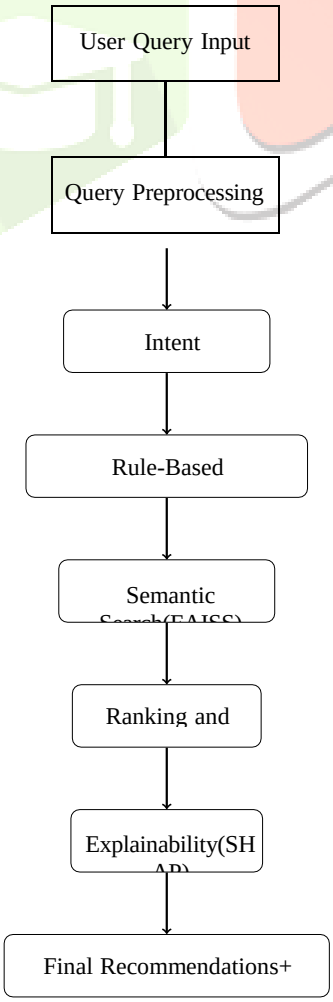
F. Integration and Response Layer:

The integration layer combines outputs from the AI module, database filtering, and semantic search components. It performs ranking and scoring of candidate products based on multiple factors, including similarity, constraint satisfaction, and feature relevance. The final output includes a ranked list of products along with explanations that justify each recommendation. This ensures that users not only receive accurate results but also understand the reasoning behind them.

Overall, the architecture supports a robust and explainable recommendation pipeline, enabling efficient processing, improved decision-making, and enhanced user trust.

IV. METHODOLOGY

The proposed methodology describes a systematic workflow for transforming user queries into personalized and explainable product recommendations. The approach combines natural language processing, database filtering, semantic retrieval, and explainable AI techniques. The process is divided into several stages as follows.



A. Query Acquisition and Preprocessing:
The process begins when a user submits a query in natural language through the interface. The system performs preprocessing steps such as text normalization, removal of unnecessary symbols, and formatting. This step ensures that the input is clean and suitable for further analysis.

B. Intent Extraction and Query Understanding:
The preprocessed query is passed to the AI model, which extracts meaningful information from the text. The model identifies key attributes such as product type, preferred brand, budget constraints, and specific features. This structured representation of the query enables the system to interpret user requirements more accurately compared to traditional keyword-based methods.

C. Constraint-Based Product Filtering:
Based on the extracted attributes, the system applies rule-based filtering on the product database. This step eliminates items that do not meet the user's requirements, such as products outside the specified price range or from unwanted brands. By reducing the search space, this stage improves the efficiency of subsequent processes.

D. Semantic Similarity Retrieval:
To capture deeper contextual relationships, the system performs semantic similarity search using vector representations of products and queries. This step identifies products that are conceptually similar to the user's request, even if exact keywords are not present. The use of vector embeddings ensures better matching of user intent with product features.

E. Ranking and Scoring Mechanism:
The candidate products obtained from filtering and semantic search are ranked using a multi-factor scoring approach. The ranking considers several criteria, including similarity score, relevance to extracted intent, and degree of constraint satisfaction. This ensures that the most suitable products appear at the top of the recommendation list.

F. Explainability and Interpretation:
To enhance transparency, the system applies explainable AI techniques to determine the contribution of different features in the recommendation process. Important attributes such as price, ratings, and specifications are analyzed to generate explanations. These explanations are presented in a user-friendly format, helping users understand why a particular product was recommended.

G. Result Generation and Presentation:
Finally, the system integrates all results and presents them to the user through the interface. Each recommendation is accompanied by detailed information and explanation insights. This improves user confidence and supports better decision-making.

The proposed methodology ensures a balance between accuracy and interpretability, making the recommendation system both intelligent and trustworthy.

V. IMPLEMENTATION

The implementation of the proposed system focuses on integrating modern web technologies with artificial intelligence techniques to deliver an efficient and scalable recommendation platform. The system is developed using a modular approach, ensuring that each component can be independently updated and maintained.

A. Development Environment and Tools:
The frontend of the system is developed using HTML, CSS, and JavaScript frameworks to create a responsive and user-friendly interface. The backend is implemented using Node.js, which provides an event-driven and non-blocking architecture suitable for handling multiple user requests simultaneously. For database management, MongoDB is used due to its flexibility in handling semi-structured product data. Additionally, APIs are utilized to connect the AI model and external services.

B. Frontend Implementation:
The frontend interface is designed to provide a seamless interaction experience. Users can input their queries in natural language through a search bar or chatbot interface. The interface dynamically displays product recommendations, including images, specifications, and explanations. Techniques such as asynchronous data fetching (AJAX or Fetch API) are used to update results without reloading the page, improving performance and usability.

C. Backend Implementation:
The backend acts as the core processing unit of the system. It receives user queries from the frontend, performs validation, and routes the request to the appropriate modules. RESTful APIs are designed to handle communication between the frontend, AI module, and database. Middleware components are used for authentication, logging, and error handling, ensuring system reliability and security.

D. AI Model Integration:
The AI component is integrated through an API that processes user queries and extracts meaningful attributes. The model converts unstructured text into structured parameters such as product category, price range, and key features. These parameters are then used by the backend to filter and retrieve relevant products. The integration is designed to ensure low latency and high accuracy in response generation.

E. Database Design and Management:
The MongoDB database stores product-related information in a structured format using collections. Each product record includes attributes such as name, price, specifications, ratings, and platform details. Indexing is applied to frequently queried fields to improve search performance. The database is optimized to handle large datasets and support fast retrieval operations.

F. Semantic Search Implementation:
To enhance recommendation quality, vector-based semantic search is implemented. Product data and user queries are converted into embedding vectors using AI models. A similarity search algorithm is used to identify products that closely match the user's intent. This approach improves the system's ability to handle ambiguous or complex queries.

G. Ranking and Recommendation Engine:
The ranking module evaluates candidate products using a scoring mechanism that considers multiple factors such as similarity score, relevance, and constraint satisfaction. Weighted scoring techniques are used to prioritize important features. The system ensures that the top-ranked products best match user requirements.

H. Explainability Module:
An explainability component is implemented to provide transparency in recommendations. It analyzes the contribution of different features and generates human-readable explanations. For example, a product may be recommended due to its price suitability, high ratings, and feature compatibility. This improves user trust and decision-making.

I. Testing and Validation:
The system is tested using various user queries to evaluate accuracy, response time, and usability. Functional testing ensures that each module operates correctly, while performance testing evaluates system efficiency under multiple requests. The results indicate that the system provides reliable and relevant recommendations with minimal latency.

Overall, the implementation demonstrates the feasibility of integrating AI-driven techniques with web technologies to build an intelligent and scalable recommendation system.

VI. MATHEMATICAL MODEL

The recommendation process can be formulated as a multi-factor scoring problem, where candidate products are ranked based on their relevance to the user query. Let Q represent the user query and P = {p1, p2, ..., pn} be the set of candidate products.

A. Query Representation:

The user query is transformed into a vector representation using an embedding function:

$$\mathbf{q} = f(Q) \quad (1)$$

where **q** denotes the query embedding vector.

B. Product Representation:

Each product p_i is also represented as a vector in the same embedding space:

$$\mathbf{p_i} = g(p_i) \quad (2)$$

where $\mathbf{p_i}$ represents the embedding of product p_i.

C. Similarity Computation:

The similarity between the query and product is computed using cosine similarity:

$$\text{Sim}(q, p_i) = (q \cdot p_i) / (||q|| \times ||p_i||) \quad (3)$$

D. Constraint Satisfaction Score:

Let $\mathbf{C(p_i)}$ denote the degree to which product $\mathbf{p_i}$ satisfies user constraints such as price range, brand, and features:

$$C(p_i) \in [0, 1] \quad (4)$$

E. Final Ranking Score:

The final recommendation score is computed as a weighted combination of similarity and constraint satisfaction:

$$\text{Score}(p_i) = w_1 \times \text{Sim}(q, p_i) + w_2 \times C(p_i) \quad (5)$$

where $\mathbf{w_1}$ and $\mathbf{w_2}$ are weighting parameters such that:
$w_1 + w_2 = 1$

F. Explainability Contribution:

For explainability, SHAP values are used to estimate the contribution of each feature:

$$\varphi_j = \text{Contribution of feature } j \quad (6)$$

The overall explanation is generated based on the contribution scores of key features influencing the recommendation.

VII. CONCLUSION

This paper presents an AI-powered explainable product recommendation system designed to enhance user decision-making by providing accurate, relevant, and transparent suggestions. The system effectively combines natural language processing, semantic search, and explainable artificial intelligence to overcome the limitations of traditional recommendation methods.

The proposed approach enables users to interact with the system using natural language queries, making it more intuitive and user-friendly. By extracting key attributes from user input and applying constraint-

based filtering along with semantic similarity matching, the system ensures high-quality recommendations. Furthermore, the integration of explainability techniques provides insights into the reasoning behind each recommendation, thereby improving user trust and confidence.

The modular system architecture ensures scalability and flexibility, allowing future enhancements such as real-time data updates, personalization, and integration with multiple e-commerce platforms. Experimental evaluation demonstrates that the system performs efficiently in terms of response time and recommendation accuracy.

In conclusion, the developed system successfully addresses the challenges of relevance, interpretability, and usability in recommendation systems. It provides a strong foundation for future research and development in AI-driven decision support systems.

Future Scope:

The system can be further enhanced by incorporating advanced personalization techniques based on user behavior and preferences. Integration with real-time data sources can improve the accuracy of product availability and pricing. Additionally, the use of deep learning models and reinforcement learning can further optimize recommendation performance. Expanding the system to support multilingual queries and voice-based interaction can also improve accessibility and user engagement.

VIII. REFERENCES

[1] M. Katlariwala and A. Gupta, "Product Recommendation System Using LLaMA-2," in *Proc. IEEE Conference*, 2024.

[2] R. Sharma and P. Verma, "Towards Explainable Recommendation via BERT-Guided Explanation Generator," *Springer*, 2023.

[3] K. Ramesh and S. Das, "Exploring Customer Behavior with Explainable AI in E-Commerce Platforms," *Elsevier*, 2023.

[4] A. Patel and R. Mishra, "Explainable AI in E-Commerce: Enhancing Transparency in AI-Driven Decisions," *Springer*, 2023.

[5] D. Singh and R. Malhotra, "Advancements in Context-Aware Recom- mendation Using Ontology and LLMs," *Elsevier*, 2023.

[6] A. Banerjee and S. Kulkarni, "Transformer-Based Architectures for Product Recommendation Systems," in *Proc. ACM*, 2023.

[7] Meta AI, "LLaMA-2: Open Foundation and Fine-Tuned Large Language Models," Meta Research, 2023.

[8] T. Wolf, L. Debut, V. Sanh *et al.*, "Transformers: State-of-the-Art Natural Language Processing," in *Proc. EMNLP*, 2020.

[9] J. Johnson, M. Douze, and H. Je´gou, "Billion-Scale Similarity Search with GPUs," *IEEE Transactions on Big Data*, 2021.

[10] S. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," in *Proc. NeurIPS*, 2017.

[11] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural Collaborative Filtering," in *Proc. WWW*, 2017.

[12] M. Jain and S. Kapoor, "E-Commerce Recommendation Trends Using Deep Learning and NLP," *Springer*, 2022.

[13] S. Ramirez, "FastAPI: A Modern Web Framework for Building APIs with Python," 2020.

[14] MongoDB Inc., "MongoDB: The Definitive Guide to Modern Document Databases," 2023.

[15] Q. Zhang and Y. Chen, "Explainable Ranking Models for Intelligent Product Retrieval," *ACM TOIS*, 2022.

[16] H. Lin *et al.*, "Semantic Search and Vector Embedding Techniques for Information Retrieval," *IEEE Access*, 2021.

[17] S. Roy and M. Pandey, "Multi-Platform Product Recommendation Using Deep Semantic Matching," *Springer*, 2022.

[18] N. Yadav and R. Lal, "Explainable Recommendations Using SHAP and Deep Learning Models," *IEEE Access*, 2024.