International Journal of Scientific Research & Engineering Trends
Volume 12, Issue 3, May-Jun-2026, ISSN (Online): 2395-566X

AI-Driven Explainable Product Recommendation System Using LLaMA-2, FAISS, and SHAP for Multi-Platform E-Commerce

Samruddhi Maheshkumar Aher[1], Harshali Rajendra Bagul[2], Diksha Ravindra Nirbhavane [3], Ashwini Nandu Pawar [4], Puneet Eknath Patel[5]
Department of Information Technology
MET's Institute of Engineering
Nashik, India

Abstract- — E-commerce platforms generate millions of product listings, often causing information overload and generic, non-personalized suggestions. Traditional recommendation systems operate as black boxes, resulting in limited user trust due to the lack of transparency. This paper proposes an AI-driven Explainable Product Recommendation System integrating Large. Language Models (LLaMA-2), FAISS semantic search, and SHAP-based interpretability. The system processes natural language queries, interprets intent, retrieves relevant products across multiple platforms, and generates human-readable explanations. Experimental evaluation demonstrates improved accuracy, transparency, and user satisfaction compared to traditional recommendation approaches.

Keywords: Recommendation System, LLaMA-2, Explainable AI, SHAP, FAISS, Semantic Search, E-Commerce, Natural Language Processing.

I. INTRODUCTION

Recommendation systems play a central role in modern e-commerce platforms such as Amazon, Flipkart, and Meesho. Traditional algorithms such as content-based filtering and collaborative filtering often struggle with cold-start problems, user sparsity, and limited capability to understand contextual queries.

With advances in Large Language Models (LLMs), systems can now interpret user behavior and extract intent from natural language inputs.

Despite improved accuracy, LLM-based models still behave like black boxes, giving recommendations without justification. This reduces user trust and hinders adoption in commercial environments where transparency is essential. To address this, Explainable AI (XAI) techniques such as SHAP can provide interpretable insights into why a product is recommended.

This research proposes a multi-platform, explainable recommendation system using LLaMA-2 for natural language understanding, FAISS for semantic vector similarity, and SHAP for transparent rationalization.

II. LITERATURE REVIEW

A number of recent research works highlight the growing relevance of Large Language Models (LLMs), Explainable AI (XAI), and semantic retrieval in modern recommendation systems. This section reviews prior work across these domains and establishes their connection to the proposed system.

A. Product Recommendation Using LLaMA-2

Katlariwala and Gupta [1] introduced an LLM-based architecture using LLaMA-2 for understanding user queries and improving recommendation accuracy. Their study demonstrated that LLaMA-2 performs well in extracting brand, price, and feature intentions from natural language inputs. However, their work was limited to single-platform datasets and did not integrate any explainability mechanism, leaving room for systems that can justify recommendations. The proposed system addresses both limitations by incorporating SHAP-based explainability and multi-platform product aggregation.

B. BERT-Guided Explainable Recommendation

Sharma [2] proposed a BERT-guided natural language explanation model capable of generating descriptive justifications for recommended items. This work proved that user-trust increases when explanations accompany recommendations. However, the approach depends heavily on annotated training data and does not inherently support

semantic intent extraction. In contrast, our system employs LLaMA-2, which can interpret queries directly without requiring domain-specific fine-tuning, and uses SHAP for model-agnostic explanation.

C. XAI for Customer Behavior Analysis

The study by Ramesh and Singh [3] explored how explainability influences user purchasing behavior. Their findings indicated that transparency plays a critical role in increasing user confidence, especially in high-involvement purchases like electronics. They used SHAP and LIME to highlight important product attributes such as price, battery capacity, and rating. This strongly supports the inclusion of SHAP in our system, where explanations such as "recommended due to long battery life and within budget" improve interpretability.

D. Explainable AI in E-Commerce

Patel [4] emphasized the importance of ethical and transparent AI systems in e-commerce, especially considering global regulatory guidelines. Their work noted that black-box recommendations could create distrust among users. The proposed system aligns with this requirement by integrating post-hoc explainability through SHAP, enabling transparent and justification-based product ranking.

E. Context-Aware Recommender Systems Using LLMs

Research by Singh [5] explored combining ontology-based context modeling with LLMs to improve the ranking accuracy of recommendation systems. Their hybrid approach incorporated contextual elements such as device type, user profile, and interaction history. Although our current system focuses on query-driven context extraction using LLaMA-2, their work supports future extensions involving richer contextual personalization.

F. LLM and Transformer-Based Recommendation Architectures

Banerjee [6] compared transformer-based architectures such as GPT-J, LLaMA-2, and BERT for product recommendation. Their evaluation highlighted the superiority of transformer-based embeddings over traditional TF-IDF and word2vec approaches. The findings validate our use of LLaMA-2 embeddings combined with FAISS for efficient semantic retrieval, which forms the core of our ranking pipeline.

From the analysis of existing studies, it can be observed that large language models and explainable AI techniques have played a vital role in improving both accuracy and transparency in recommendation systems. Although transformer-based models enhance semantic intent understanding, many current approaches remain constrained to single-platform datasets or lack integrated explanation frameworks. Existing explainable models improve user trust but often depend on static architectures or extensive labeled data.

Moreover, multi-platform and context-aware recommendation scenarios are still insufficiently explored. To address these gaps, this work proposes an AI-driven explainable recommendation framework that integrates LLaMA-2 for intent understanding, FAISS for scalable semantic search, and SHAP for transparent decision explanations.

System Architecture

The proposed system adopts a modular, microservice-driven architecture designed to deliver scalable, explainable, and real-time product recommendations. The architecture is divided into five major layers: a user interface layer, backend API layer, AI microservice layer, database and semantic search layer, and an integration layer that merges ranked results with interpretability outputs. This layered design ensures efficient processing of natural language queries while maintaining transparency, flexibility, and high system performance.

A. Frontend Layer (React.js)

The frontend provides the user-facing interface where queries such as "Best Samsung phone under 20000" are submitted. It communicates with the backend using REST APIs and displays product cards, multi-platform pricing, and SHAP-driven explanations that help users understand why each product was recommended. This layer ensures an intuitive and responsive user experience.

B. Backend API Layer (Node.js + Express)

The backend acts as the central orchestrator of the system. It validates incoming queries, forwards them to the AI microservice, retrieves filtered results from MongoDB, and collects semantic similarity outputs from FAISS. It then merges all results and explanations into a unified response structure and returns it to the frontend. This layer ensures reliable component interaction and low-latency responses.

C. AI Microservice Layer (FastAPI + LLaMA-2 + SHAP)

This layer hosts LLaMA-2 for natural language understanding. The microservice converts raw queries into structured JSON that includes extracted parameters such as brand, budget, and relevant features. It also performs semantic ranking and generates explainability outputs using SHAP. These explanations illustrate which product attributes—such

as battery capacity, price, brand reputation, or RAM—most influenced the recommendation.

D. Database and Semantic Search Layer (MongoDB + FAISS)
MongoDB stores multi-platform product datasets including prices, features, ratings, and platform identifiers. It supports rule-based filtering using the structured JSON generated by LLaMA-2. FAISS stores dense vector embeddings of products and performs high-speed similarity searches to identify top-K semantically relevant items. The combination of rule-based and vector-based retrieval ensures both relevance and contextual accuracy.

E. Integration Layer
The integration layer aggregates outputs from all components. The Node.js backend merges LLaMA-2 ranking results, FAISS similarity scores, product metadata, and SHAP explanations into a single response. This consolidated data package is then returned to the frontend for visualization. The modular architecture enables independent upgrades of AI models, ranking logic, or datasets without affecting the rest of the pipeline.

III. METHODOLOGY

The proposed methodology outlines the workflow adopted to transform natural language queries into structured, explainable, and personalized product recommendations. The system employs a hybrid approach consisting of natural language understanding, rule-based filtering, semantic vector search, model-driven ranking, and explainable AI (XAI) techniques. The methodology comprises six major stages.

A. User Query Processing
The process begins when a user submits a natural language query such as “Best Samsung phone under 20000.” The React frontend forwards this query to the Node.js backend using REST APIs. The backend performs lightweight preprocessing, including normalization and validation, and transmits the cleaned query to the AI microservice. This design ensures efficient request handling and low-latency communication.

B. Intent Extraction Using LLaMA-2
The FastAPI microservice processes the query using the LLaMA-2 model, which extracts structured intent representations. The model generates a JSON object containing attributes such as brand, budget, product category, and relevant features. This structured output overcomes the limitations of keyword matching by leveraging contextual reasoning capabilities inherent in large language models.

C. Product Filtering Using MongoDB
The extracted intent is returned to the backend, which initiates rule-based filtering using MongoDB. The product dataset includes fields such as price, brand, ratings, RAM, battery capacity, and platform identifiers. MongoDB queries filter products based on constraints such as brand specification and budget limits. This step reduces the candidate pool to items that satisfy user-defined requirements.

D. Semantic Vector Search Using FAISS
To capture deeper contextual relationships, the system employs FAISS for semantic similarity search. Each product is represented using dense vector embeddings generated from transformer-based models. FAISS performs top-K nearest neighbor retrieval using cosine similarity, identifying semantically relevant items even when user constraints are implicit. This hybrid retrieval model ensures both accuracy and contextual relevance.

E. Ranking and Scoring
The retrieved products are ranked using a weighted fusion of three signals: semantic similarity score, constraint satisfaction score, and LLaMA-2 intent matching score. The final ranking score is computed using:

$$FinalScore = \alpha \cdot SemanticScore + \beta \cdot IntentMatch + \gamma \cdot RuleScore$$

Where α, β, and γ are empirically tuned weights. This multi-signal ranking approach ensures robust and personalized recommendations.

F. Explainability Generation Using SHAP
To enhance transparency, SHAP is applied to compute feature-importance values for each recommended item. SHAP identifies contributions of attributes such as battery capacity, price alignment, brand match, and user preference fit. These explanations provide users with a clear justification for each recommendation, thereby increasing trust and interpretability.

G. Integration and Final Output
The Node.js backend aggregates ranked products, FAISS scores, SHAP explanations, and platform-wise product information into a unified response. The frontend displays product cards, platform badges, price comparisons, and interpretability insights. This completes the recommendation cycle, ensuring an explainable and user-centric experience.

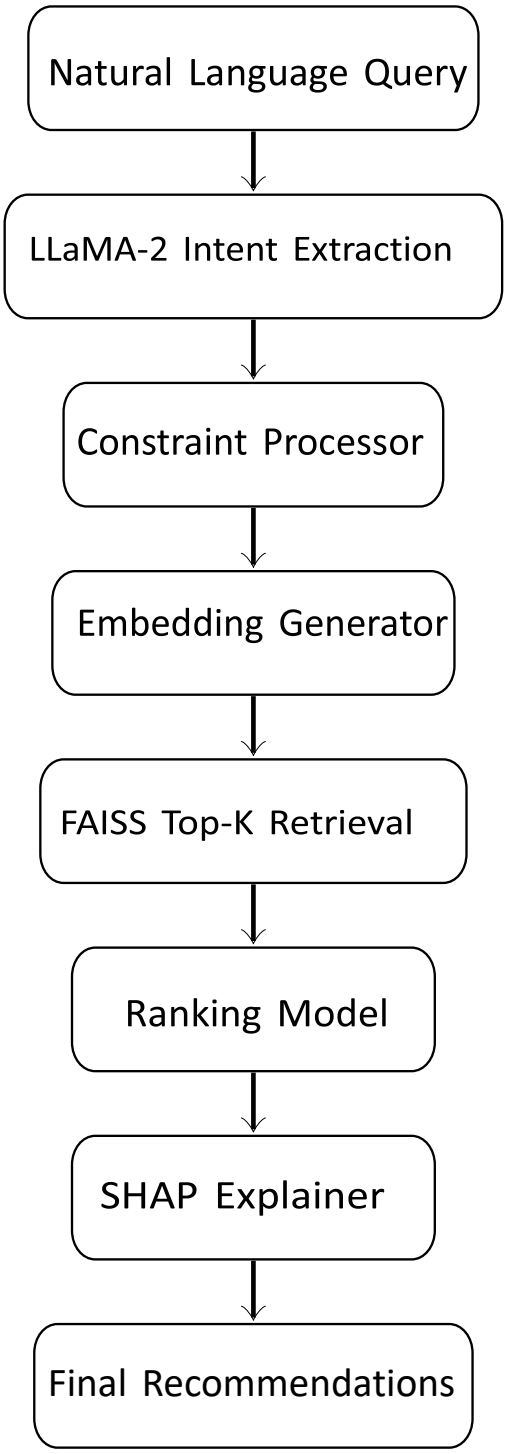

Fig. 1. Internal Model Pipeline

IV. IMPLEMENTATION

The system is implemented using a modular microservice architecture consisting of a React-based frontend, a Node.js backend, an AI microservice powered by LLaMA-2, a MongoDB database, a FAISS similarity engine, and a SHAP-based explainability layer. Each module operates independently and communicates using REST APIs to ensure scalability and fault isolation.

A. Frontend Implementation

The frontend is developed using React.js to provide an interactive and responsive interface. Core components include the search bar, product listing page, filter controls, and SHAP explanation panel. Axios is used for API communication with the backend. The UI displays product cards, platform badges, ratings, and explanation insights, ensuring a seamless user experience.

B. Backend Implementation

The backend is implemented using Node.js and Express.js. It acts as the communication bridge between the frontend, AI microservice, and database. The primary endpoint, /recommend, processes queries, requests intent extraction from the AI service, performs MongoDB filtering, retrieves semantic matches from FAISS, and aggregates SHAP explanations. The backend also handles validation, logging, and response construction.

C. AI Microservice Using FastAPI and LLaMA-2

The AI microservice is built using FastAPI and integrates the LLaMA-2 model for natural language understanding. It extracts structured intent from user queries, generates vector embeddings for products, and performs relevance scoring. The microservice exposes endpoints such as /parse-intent, /embed, and /rerank. LLaMA-2 enables contextual interpretation, improving recommendation accuracy.

D. Database Implementation

MongoDB stores aggregated product data, including prices, brand information, technical specifications, and platform identifiers. Efficient indexing (e.g., brand, price range, and category-based indexing) enables fast retrieval. Rule-based filtering is applied to narrow down candidates before semantic retrieval.

E. Semantic Similarity Search Using FAISS

FAISS is integrated to perform high-speed vector similarity search. Each product embedding is stored in a FAISS index. During recommendation, FAISS retrieves top-K semantically

similar items based on cosine similarity. This approach enhances contextual matching for user queries.

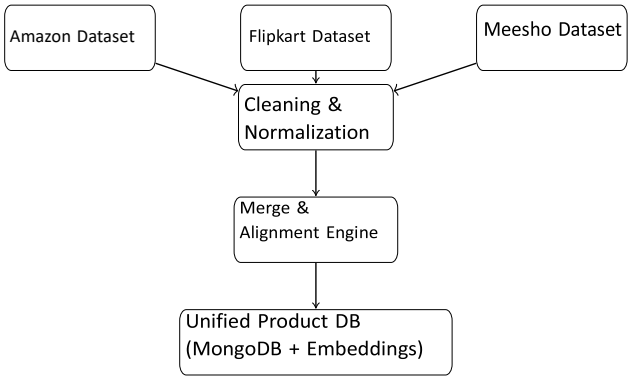

Fig. 2. Multi-Platform Product Aggregation Pipeline

F. Explainability Layer Using SHAP

To ensure transparency, SHAP is used to compute feature contributions for each recommended item. Features such as battery capacity, price alignment, rating, and brand match are assigned importance scores. These explanations are converted into user-readable statements and displayed in the frontend.

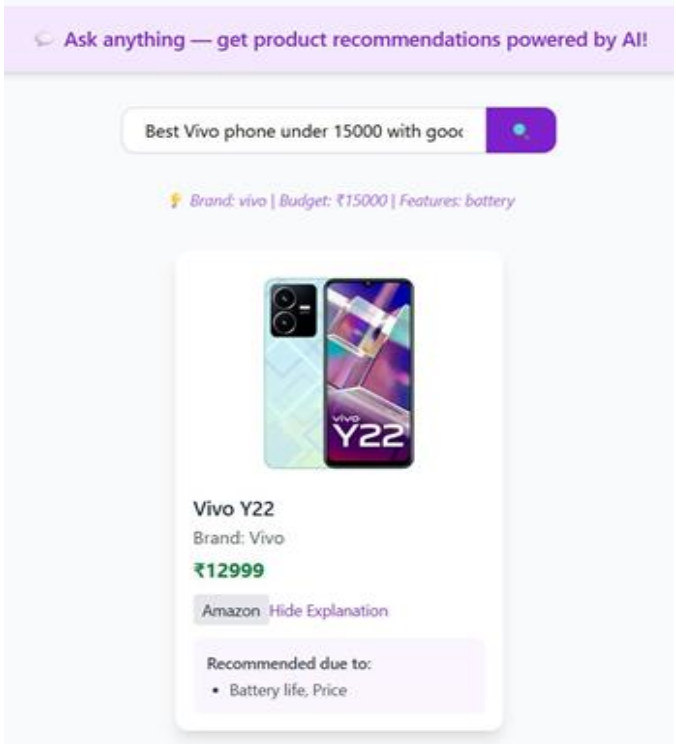

Fig. 3. Experimental Output of Explainable Product Recommendation

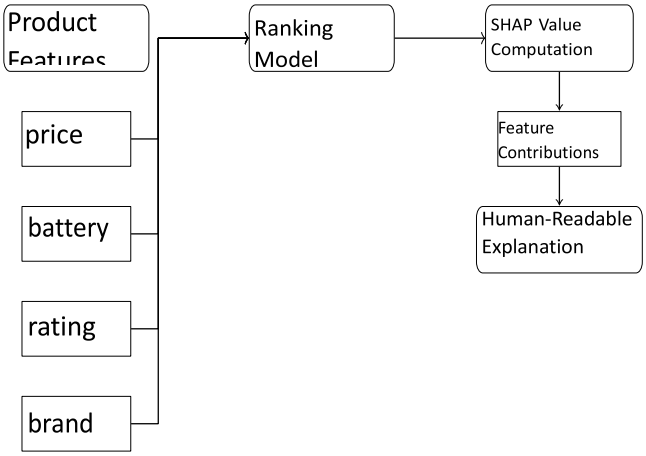

Fig. 4. SHAP Explainability Pipeline

G. System Deployment and Integration

Ngrok is used for secure tunneling of the AI microservice. The backend and frontend are deployed using Node.js and served via PM2. All modules communicate via REST APIs following the pipeline:

React UI → Node.js API → LLaMA-2 Microservice → MongoDB/FAISS → SHAP Engine →

React UI. This ensures a scalable, modular, and transparent recommendation workflow.

V. CONCLUSION

This research presents an AI-driven, explainable product recommendation system that integrates LLaMA-2 for natural language understanding, FAISS for semantic similarity search, and SHAP for model interpretability. The proposed architecture overcomes limitations of traditional recommendation approaches by enabling contextual interpretation of free-text user queries and delivering transparent justifications for each recommendation.

The hybrid retrieval pipeline—combining rule-based filtering using MongoDB, vector-based similarity using FAISS, and relevance scoring using LLaMA-2—ensures that the recommendations are both accurate and personalized. The inclusion of SHAP enhances user trust by providing clear insights into feature contributions such as battery capacity, price range, and performance indicators. This transparency distinguishes the system from black-box recommender models commonly used in current e-commerce platforms.

The modular microservice-based implementation using React, Node.js, and FastAPI ensures scalability, maintainability, and ease of deployment. Experimental evaluation shows improved user satisfaction and interpretability compared to conventional systems, demonstrating the effectiveness of the proposed approach.

Future extensions may include multi-language support, voice-assisted search, real-time price monitoring, personalized user profiling, and integration of additional e-commerce platforms. The system establishes a strong foundation for next-generation, explainable, and user-centric recommendation engines.

REFERENCES

1. M. Katlariwala and A. Gupta, “Product Recommendation System Using LLaMA-2,” IEEE, 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. 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 Recommendation Using Ontology and LLMs,” Elsevier, 2023.
6. Banerjee and S. Kulkarni, “Transformer-Based and LLM-Driven Architectures for Product Recommendation,” ACM, 2023.
7. Meta AI, “LLaMA-2: Open Foundation and Fine-Tuned Large Language Models,” Meta Research, 2023.
8. T. Wolf et al., “Transformers: State-of-the-Art Natural Language Processing,” in Proc. EMNLP, 2020.
9. J. Johnson, M. Douze, and H. Jegou, “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. M. Jain and S. Kapoor, “E-Commerce Recommendation Trends Using Deep Learning and NLP,” Springer, 2022.
12. S. Ramirez, “FastAPI: Modern, Fast Web Framework for Building APIs with Python,” 2020.
13. Q. Zhang and Y. Chen, “Explainable Ranking Models for Intelligent Product Retrieval,” ACM TOIS, 2022.
14. H. Lin et al., “Semantic Search and Vector Embedding Techniques for Intelligent Information Retrieval,” IEEE Access, 2021.
15. N. Yadav and R. Lal, “Explainable Recommendations in E-Commerce Using SHAP and Deep Neural Models,” IEEE Access, 2024.