National College of Ireland

Generative-Agentic AI (GA-AI) Framework for Product Recommendation

MSc Research Project
Masters in Artificial Intelligence

Durga Nagendra Prasad Gonugunta
Student ID: 23285524

School of Computing
National College of Ireland

Supervisor: Paul Stynes

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	DURGA NAGENDRA PRASAD GONUGUNTA		
Student ID:	23285524		
Programme:	Masters in Artificial Intelligence	**Year:**	2025
Module:	Practicum		
Supervisor:	Paul Stynes		
Submission Due Date:	15-09-2025		
Project Title:	Research Paper		
Word Count:	5847 **Page Count 14**		

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: DURGA NAGENDRA PRASAD GONUGUNTA

Date: 15-09-2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Generative-Agentic AI (GA-AI) Framework for Product Recommendation

Durga Nagendra Prasad Gonugunta
23285524

Abstract

Availability of millions of products online has made the product recommendation systems crucial. Product recommendation systems suggest relevant products to the users based on their requirement. Current research in product recommendations rely on Collaborative Filtering and Deep Learning models but these lack conversational ability. Building a product recommendation system that understands user intent, retrieves the product and generates conversational response is a challenge. This research proposes a Generative-Agentic AI framework to deliver product recommendations through conversation. The proposed framework combines Intent Classification Agent, Retrieval Agent, Generation Agent and Controller Agent for product recommendation through conversation. Amazon 5 core product review dataset covering six categories of Toys and Games, Musical Instruments, Cellphones and Accessories, Appliances, All Beauty and Amazon Fashion has been used for this research. Data Analysis techniques have been applied on this dataset to build Embeddings from it. Logistic Regression for Intent classification, Facebook AI Similarity Search (FAISS) index with MiniLM embeddings for Retrieval Agent and Quantized OpenHermes Language Model for Generation Agent are combined to build an Agent based recommendation framework. Results are evaluated using classification accuracy for Intent Classification Agent, Mean Reciprocal Rank, Hit Rate and Relevancy score for Retrieval agent and BLEU, BERTScore, ROUGE-L score, faithfulness and relevancy for Generation Agent. The research shows promise for a conversation based Product Recommendation using Generative Agentic AI Framework. It benefits the community by demonstrating competitive accuracy and reliability, while reducing computational cost.

1 Introduction

Recommender systems have become crucial for helping the users navigate an overwhelming number of products and content in the present era. Over the past decades, recommendation techniques have evolved significantly, from rule-based and content-based approaches to collaborative filtering methods, which infer user preferences by leveraging patterns of similar users or items. Collaborative filtering emerged as a dominant approach by the 2000s. In recent years, the field has witnessed a revolution through deep learning. Advanced neural models have been integrated into recommender systems to capture complex user-item relationships and address data sparsity. These advancements improved the accuracy and personalization of recommendations. However, Traditional and Deep Learning based recommenders faced challenges like the cold start problem and a lack of explainability. These limitations motivate the exploration of new methods that can provide more robust, interpretable and user-friendly recommendation experiences.

To overcome the above limitations, researchers have begun to explore Generative AI and Agentic Frameworks as the next level for recommender systems. The rise of Large Language

Models (LLMs) opens the door for conversational recommender systems that can interact with users in natural language. Unlike Traditional Recommenders that just produce list of items, a conversational system can ask questions, clarify needs, and explain recommendations through a conversation providing a richer and more engaging user experience. Studies have integrated LLMs into recommendation pipelines in two main ways either by pairing an LLM-based dialogue module with a separate traditional recommender engine or by using an LLM alone to both converse and generate recommendations. Thus, Generative AI promises to address the explainability gap by producing human-like justifications for recommendations and solve cold start issues by feeding on semantic knowledge. At the same time, the concept of Agent AI frameworks has emerged, advocating a modular architecture where agents handle different sub-tasks in a coordinated manner. Recent works suggest that complex tasks which overwhelm a single monolithic model can be tackled by decomposing them into sub-tasks handled by multiple agents. In the recommendation field, an Agentic Framework involves one agent for intent understanding, another for retrieving relevant product information, and another for generating the recommendation response. This design is modular and extensible, which improves scalability and maintainability of the system. Current Research lacks a unified system that combines **semantic retrieval, generative reasoning, and a modular agentic design.** Some studies have augmented LLM recommenders with external knowledge retrieval to ground the model's reasoning in product data and others have proposed multi-agent strategies to manage recommendation dialogues and planning. However, an integrated framework that brings together semantic retrieval, LLM-driven generation and multi-agent coordination into one recommendation system remains largely unexplored in the literature. This gap highlights the importance of developing this Generative Agentic Framework approach that fills the research void and meets the expectations for recommender systems that not only perform well but also communicate and reason about their recommendations in an understandable way.
Motivated by the above gap, **this research asks the question:**

How accurately can a Generative Agentic AI Framework produce faithful, relevant and conversational product recommendations in response to natural language queries?

In other words, investigation of whether an approach that integrates semantic retrieval of product information, generative reasoning via an LLM, and a multi-agent modular design can deliver recommendations that are both accurate to the data, relevant and conversational in manner.

To address the research question, four primary **objectives** have been established and each aligned with a critical component of the proposed system.

- **Intent Classification:** To develop a module for interpreting user's natural language query and identify their intent.
- **Semantic Retrieval:** Implement a Semantic Retrieval mechanism that can fetch relevant product data from the knowledge base using the query's meaning.
- **Generative Recommendation:** Design a Generative Response Module that uses a Large Language Model to compose the recommendation output through conversation.
- **Evaluation and Refinement:** Establish an Evaluation framework to assess the system's performance on all modules.

The **Major Contribution** of this research is the implementation of a novel Generative Agentic AI Framework for Product recommendation by integrating intent classification, semantic retrieval, and natural language generation.

Minor contributions include construction of a cleaned, product-level dataset from Amazon Reviews, Integration of a Large Language Model for low-resource deployment and the introduction of agent-level evaluation metrics to assess the performance of each agent individually.

Several **limitations** of the Research have been Acknowledged. The Evaluation is conducted on a limited set of test queries. Due to time and resource constraints, the number of queries used to validate the system is modest. The Framework does not incorporate long term personalization as each query has been treated in isolation. The Framework focuses on Single-turn interactions. Reliance on a quantized LLM may result in the richness of the generated text compared to a full size model. The Evaluation has been done with simulated queries.

This paper discusses Product Recommendation systems in section 2 Related Work. The Research Methodology is discussed in section 3. Section 4 discusses the design specification for the Generative Agentic AI Framework. The implementation of this research is discussed in section 5. Section 6 presents and discusses the evaluation results. Section 7 concludes the research and discusses future work.

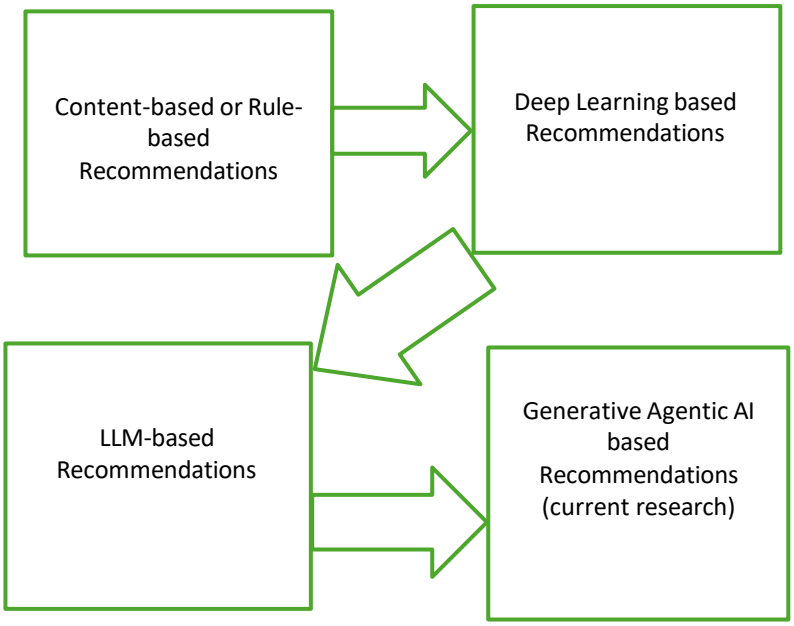
2 Related Work

Latha and Rao developed an Amazon product recommender using a modified Convolutional Neural Network(MCNN) and performed sentiment analysis on user reviews. They utilized the Amazon product reviews dataset containing 60,000 reviews. TF-IDF vectorization and text processing has been applied. The MCNN, combined with Skip-gram and GloVe word embeddings was trained to classify review sentiment. The model achieved a mean accuracy of 97.4%. Naz conducted a comprehensive survey of product recommendation methods. Review of Traditional approaches like content-based filtering, collaborative filtering, and modern hybrid approaches has been done. The research stated that hybrid recommender systems can mitigate challenges like the cold-start problem and improve personalization. Common Evaluation metrics like accuracy, RMSE, precision/recall have been discussed. Kinkar presented a systematic literature review of product recommendation algorithms. Various techniques like content-based, collaborative, knowledge-based and hybrid models have been surveyed along with the performance measures used to evaluate them. Emphasis on personalization and scalability has been observed in this research. Ahmed built an Amazon product recommender that blends collaborative filtering with sentiment analysis. Amazon product review dataset of Musical Instruments category has been used. Sentiment classification has been done using XGBoost ensemble classifier and accuracy of 93% has been achieved. These sentiment scores were then integrated with a traditional item-based collaborative filtering module. Cosine similarity on user rating vectors has been used. The research gave an RMSE of 52% for predicted ratings, outperforming a baseline CF model. Li proposed ARAG, a multi-agent Retrieval-Augmented Generation framework for personalized recommendation. ARAG has four specialized LLM-based agents. A user understanding agent to summarize user preferences, a Natural Language Inference agent to ensure retrieved items match the inferred intent, a Context Summarization agent, and an Item Ranking agent which work together. This framework was evaluated on three categories of

clothing, electronics, home using conversational recommendation tasks. It achieved up to a 42% improvement in NDCG@5 and a 35.5% improvement in Hit@5 over the best baseline in top-5 recommendation performance. This research indicated that the agentic design better captures user intent, yielding more relevant results. Li Wu and Tang's research introduced Chat-Rec, an interactive recommendation system that combines a traditional recommender with an LLM for conversational interaction. MovieLens 100K dataset has been used. Evaluation metrics of Precision@5, Recall@5, NDCG@5 for recommendation; RMSE, MAE for rating predictions have been used. Chen's research proposed a Contrastive Quantization-based Semantic Tokenization method (CoST). It is for generative recommender models that treat recommendation as an auto-regressive item generation task rather than ranking. The research introduces a learned vector quantization module to convert items into discrete semantic tokens that an LLM can generate. CoST uses contrastive learning to ensure that items with similar user engagement patterns are mapped to semantically close token codes. MIND news recommendation has been used as dataset. It achieved up to a +43% improvement in Recall@5 and about +44% in NDCG@5 on MIND compared to the best baseline method.

Wang and Zhang's research compares Large Language Models and Classical collaborative filtering for recommendation tasks. Matrix Factorization model represented collaborative filtering baseline and ChatGPT represented LLM. This research stated that LLMs are not inherently good collaborative filters, supplying structured interaction data allows them to make use of collaborative signals. Kim's research AgentRecBench is the first comprehensive benchmark to evaluate LLM-based agentic recommender systems. Interactive Simulation environment with rich user-item metadata has been built. Three scenarios: a classic static scenario, an evolving-interest scenario, and a cold-start scenario have been tested for. This research offered a valuable evaluation framework for agent-based recommendation systems. Lee's research proposed LC-Rec, a novel framework to adapt LLMs for direct recommendation generation. The core problem addressed is that LLMs operate over words instead of IDs. The research solved this by assigning each item a learned semantic identifier through residual vector quantization. Amazon product reviews containing categories of Musical Instruments, Arts, Video Games has been used as dataset. LC-Rec achieved about a 25.5% improvement in full-ranking Hit Rate/NDCG.

Other Researches involved Ahmed with collaborative filtering techniques, showed that incorporating review sentiment can improve product recommendation accuracy. Liu's Research investigated ChatGPT's capabilities as a recommender and found that while it can generate excellent natural-language explanations and summaries of user preferences, its out-of-the-box recommendation accuracy only reached *baseline levels* on certain tasks. Huang's research introduced InteRecAgent, a modular system where an LLM is used as the brain for reasoning and dialogue and is backed by traditional recommender models as tools. This approach achieved satisfying conversational recommendation performance. Meng's research proposed KERAG_R, which combined an LLM recommender with a knowledge graph. The model reduced recommendation errors due to LLM hallucination. These recent developments state a trend towards hybrid systems that combine the strengths of Large Language Model for conversation and reasoning and the strengths of agent based approach in the next generation recommender systems. The below figure shows the trend of the recommendation systems.

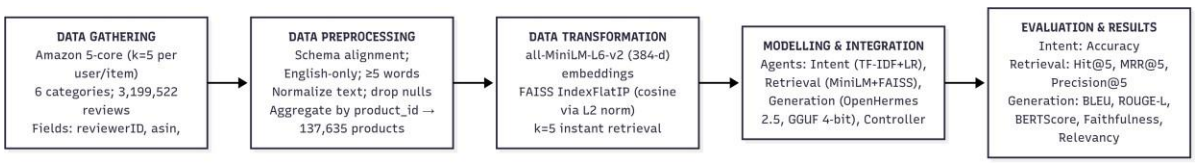


2.1 Research Gap

The literature showed strong progress from traditional models focus on structured similarity, deep learning improves prediction accuracy, and generative models enable conversational experiences. However, most current works focus on either retrieval accuracy or conversational generation, not both in one modular framework. Few solutions use multi-agent architectures for **intent classification, fast retrieval, and natural language generation** together. This research addresses these gaps by building a Generative Agentic AI Framework consisting of Intent Classification Agent for sentiment-based intent detection, Retrieval Agent through FAISS for efficient product search. LLaMA based Generation Agent for conversational Recommendations and a Controller Agent.

3 Research Methodology

The research methodology consists of five key steps: **Data Gathering**, **Data Pre-processing**, **Data Transformation**, **Data Modeling and Integration**, and **Evaluation and Results**, as illustrated in the below diagram. Each step is detailed below.



3.1 Data Gathering

The first step, Data Gathering, focused on choosing the proper dataset for the research. Amazon 5-core product review dataset has been used as the dataset. The dataset contained six categories: Toys and Games, Musical Instruments, Cell Phones and Accessories, Appliances, All Beauty, Amazon Fashion. The files was downloaded in the form of JSON. 5-core means that every user and every product has more than five reviews in the dataset. The dataset contained 3,199,522 reviews across six categories and it contained reviewer id, asin, review text, summary, overall rating as the features. This dataset is perfect for the current research as it provides rich, abundant reviews and 5-core filter guarantees a minimum strength per product.

3.2 Data Pre-processing

The second step, Data Pre-processing, focused on cleaning the raw dataset and making it into a clean, consistent and analysis ready dataset suitable for semantic embedding and retrieval. This step involved Schema alignment, language and content filtering, text normalization, handling missing values. The dataset across different categories though shared the same basic JSON structure it varies slightly in additional fields. Schema alignment made sure that all the subsets of the data maintained a unified schema by renaming some fields. To maintain linguistic consistency and relevance language and content filtering was applied. It ensured that the reviews retained were only English language reviews. Reviews that are shorter than five words were filtered as they provide insufficient context for meaningful embeddings. To improve tokenization and embedding quality text normalization was done. It involved converting all text to lowercase, removing HTML tags and special characters, removing extra whitespaces, standardizing punctuation, preserving numbers and common units as they are important in product descriptions. Missing values were handled. Rows with missing product id, review text or rating were dropped as the dataset records were high. The dataset after cleaning was reduced to **2,630,050** rows at review-level. This review-level dataset was aggregated into product level dataset. All reviews for the same product id were grouped together. The aggregation speeds up FAISS search and produces richer semantic profiles. After this aggregation at product level the dataset was reduced to **137,635** rows.

3.3 Data Transformation

The third step, Data Transformation, involved converting the aggregated product level data into formats suitable for semantic retrieval and downstream modeling. Each product's combined review text was transformed into a fixed length embedding (numeric vector) that captures its semantic meaning. Pre-trained transformer based sentence embedding model **all-Minilm-L6-v2** from the SentenceTransfomers library has been used to map each product text into a 384-dimensional vector space. These vector embeddings enable semantic similarity comparisons between products.

After generating embeddings for 137,635 unique products, an efficient similarity search index using Facebook's FAISS library was built. Using FAISS, the product embeddings were indexed in an IndexFlatIP structure, which allows cosine similarity to be computed via inner

product on L2 normalized vectors. This indexing makes sure for instant retrieval. This stage ensures that the generation model in the next step works only with relevant retrieved candidates.

3.4 Data Modelling and Integration

The fourth step, Data Modeling and Integration, consists of an agent-based recommendation pipeline and the deployment of a retrieval-augment generation system. Pre-trained components were integrated into agents that are responsible for a specific stage in the recommendation process. The pipeline begins with the Intent classification agent that uses TF-IDF vectorization and a Logistic Regression classifier to identify the intent of the user through the query. Following this agent, the Retrieval Agent uses the FAISS index to retrieve the top five most similar products to the query. This stage is designed for exact inner product search using IndexFlatIP that ensures that the system always returns the most relevant products. The retrieved products are then passed to the Generation Agent, that is powered by the OpenHermes 2.5 LLM, built on the Mistral 7B architecture. This model was best suited for the research as it generates strong conversational and reasoning abilities with given space and time constraints. To enable efficient deployment in the environment the model was loaded in a GGUF quantized format. This quantization reduced memory usage while maintaining generation quality. The controller agent begins the flow between components. After receiving query, it first calls the intent classification agent, routes the query to the retrieval agent, formats the retrieved product information into a structure prompt, and invokes the Generation agent to produce the response. The end result of this phase is a fully implemented Generative Agentic AI Framework that recommends products to the user.

3.5 Evaluation and Results

The fifth step, Evaluation and Results, involved evaluating the performance of each agent in the Generation Agent based recommendation framework. The Intent classification agent was evaluated using classification accuracy. The retrieval agent was evaluated using Hit, Mean Reciprocal Rank and Precision to assess the relevance and ranking of retrieved products. The Generation Agent was evaluated using BLEU, ROUGE-L and BERTScore to measure linguistic similarity to a reference response, manual annotation was used to assess Faithfulness and Relevancy.

4 Design Specification

The proposed Generative Agentic Product Recommendation Framework is a modular, multi-agent framework that is designed to deliver conversational product recommendations. It is implemented in three Python Notebooks for better understanding. The first notebook focuses on data preprocessing and exploratory data analysis. The second notebook focuses on Embedding and FAISS index creation. The third notebook focuses on Agentic pipeline and evaluation. The architecture consists of four agents. Intent classification agent, Retrieval Agent, Generation Agent and Controller Agent. The design specifications are discussed below.

Intent Classification Agent classifies the user's query to determine its intent. This agent uses a text classification model. Combination of TF-IDF and Logistic Regression are used to decide how the query should be handled.

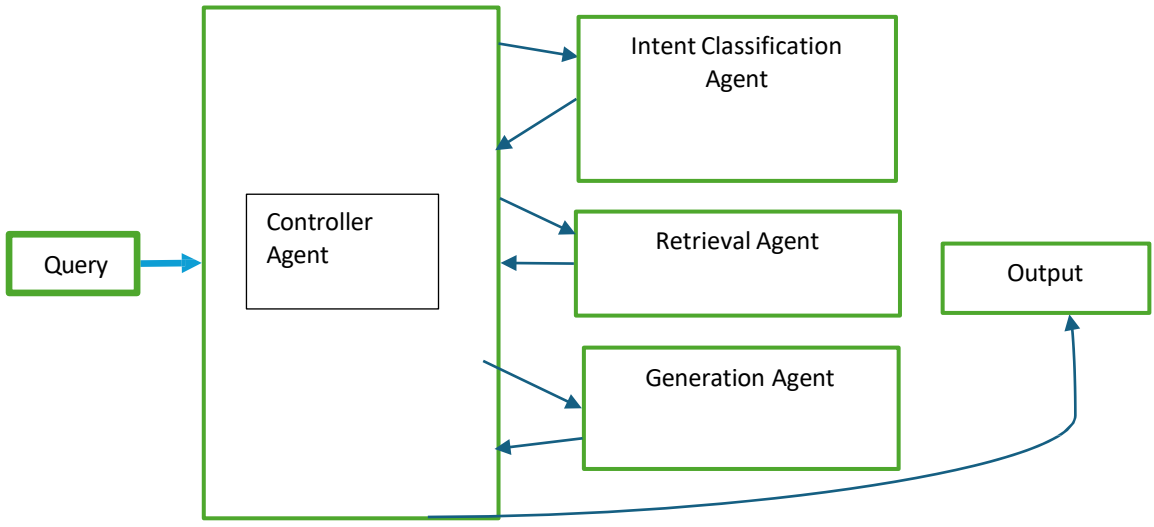
Retrieval Agent handles retrieval of relevant product data. It generates a semantic embedding of the query and searches the product level dataset for similar items.

Generation Agent deals with generating a message explaining why the retrieved product suits the user's query. OpenHermes LLM is used in this.

Controller Agent deals with passing the query from intent classifier agent to retrieval agent to generation agent and produces the output.

4.1 Data flow

Query from the user is processed through the following stages
The first stage is Intent Analysis. Controller agent passes the raw query to the intent classification agent. This agent transforms the query text into a feature vector using Term Frequency-Inverse Document Frequency (TF-IDF) and applies a Logistic Regression classifier to predict the query's intent category. The second stage is Semantic Embedding. After confirming that the user is looking for a product, the controller engages the Retrieval Agent. The third stage is Product Retrieval. The Retrieval Agent returns the top five product recommendations based on the user's query. The fourth stage is prompt construction. The controller agent now engages the generation agent and provides a suitable prompt. The fifth stage is Generative Response. The Generation Agent now generates a human-like response explaining the returned product suits the user's query. The final stage is Result Delivery. The controller collects the output from the generation agent and displays the final result.
The below figure shows the outline of the data flow in the framework.



4.2 Software and Hardware Requirements

This framework can run entirely on modest hardware. The model is designed to be executed on Google or a similar environment. A standard Colab instance was sufficient to perform this model. Access to a GPU is great for embedding generation. The OpenHermes LLM is implemented in a quantized model which took around 1.5GB. The code runs on Python 3. Required libraries include pandas, scikit-learn, sentence-transformers, faiss-cpu and ctransformers. The MiniLM and OpenHermes model files should be downloaded from hugging face.
In conclusion, this framework is flexible because of its modular design, manageable resource requirements and clear logical structure. Thereby supporting ongoing innovation and scalability of the solution.

5 Implementation

The framework has been implemented in three colab notebooks in python. The first notebook contained data preprocessing and exploratory data analysis. Data cleaning has been performed to ensure the dataset quality. Data cleaning involved removing records with null fields, reviews were all normalized into lower case for better embeddings, special characters, extra whitespaces, non-English words were removed. Data visualization has been performed. The original dataset was grouped on review-level. It was regrouped into product-level to ensure optimization of space and time complexity.

The next notebook involved the next phase. Embeddings were built in this phase. The product level dataset from the before notebook was utilized. MiniLM model was used for embedding generation. Embeddings were generated in batches to manage memory usage and runtime. GPU was used in this phase to reduce the time complexity by significant percentage. Embeddings were normalized for cosine similarity search. Cosine is more accurate for retrieval and works the best on the selected dataset. Indexing was done with Facebook's AI Similarity Search. IndexFlatL2 index type has been chosen which is an exact Euclidean distance nearest-neighbor index without compression. Flat index was used to ensure high recall and because the scale of data was manageable in memory. After index construction, the index and accompanied data structures are saved to the local file system to reuse in the next notebook. A sample query has been tested and it returned relevant product IDs based on the query. The output of this phase were an faiss index file and a pickle file storing the list of product IDs in the same order as the embeddings were added to the index.

The final notebook involved the Generative Agentic AI Framework. It handles the user queries for product recommendations. This framework contains agents working in sequence that is coordinated by a controller. All agents were implemented using python. The agents and their roles are as follows:

Intent Classifier Agent: It classifies the user's query to determine the intent. It's primary job is to decide whether the query is asking for a product recommendation. This agent was implemented as a simple machine learning classifier using textual features of the query. This agent informs the controller whether to proceed with retrieving products or handle the query differently.

Retrieval Agent: It is invoked by the controller agent, after the recommendation intent is confirmed. It takes the user query and embeds it using the same SentenceTransformer model

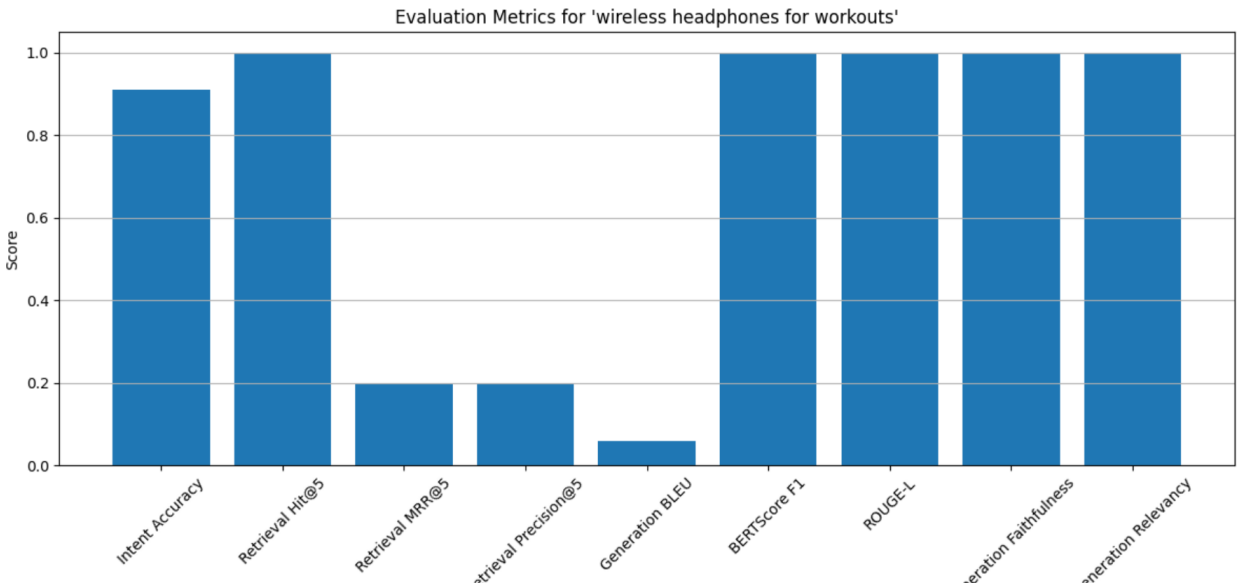
used for products. The query embedding is then submitted to the FAISS index to perform a nearest-neighbor search. Search is configured to retrieve the top five most similar product vectors. The agent then maps these back to their product IDs and retrieves the corresponding product details from the product dataset. The output of this agent is a set of top five products that are semantically related to the query. This agent effectively serves as the memory lookup, ensuring that the generative model has relevant, factual grounding information for the recommendation.

Generation Agent: It is responsible for generating the final output to the user. It is given the query and retrieved product information as input. A quantized OpenHermes model has been used as the Large Language Model. The quantized model has been loaded in the environment using the ctransformers library. The agent constructs a prompt that includes the user's query and the relevant details of the top retrieved products. The prompt is designed to encourage the model mention the retrieved product as suggestions and to just the recommendation using facts from the reviews. The agent ensures the output stays on topic and factual by ground the prompt in actual product data, thereby reducing hallucinations.

Controller Agent: It controls the entire pipeline. It is implemented as a simple rule-based controller. It receives the query and invokes the intent classifier agent. Based on the predicted intent, the controller decides the next steps. For Product recommendation intent, it calls the retrieval agent to obtain products, then passes the query and the products into the generation agent. The agent then returns the output of the generation agent as the final answer to the user.

This design allows each agent to operate independently, which implies improvements can be made to one component without changing the others. The use of separate agents for classification, retrieval, and generation also adds transparency to the framework's decision-making process.

6 Results and Discussion



The above figure summarizes the performance of each agent in the Generative Agentic AI Framework for product recommendation. The framework was tested on the sample "wireless headphones for workout". The results demonstrated high accuracy in understanding user intent, effective retrieval of relevant products, and strong quality in the generated recommendation response. Each Agent's performance is discussed below in detail.

Intent Classification Agent performance

The agent achieved an accuracy of 0.909. It indicates that it interpreted the user's intent in about 91% of the test instances. The high accuracy reflects a robust understanding of the sample query. Future improvements for this agent could further improve the accuracy. It includes using Neural Networks for classification.

Retrieval Agent Performance

The metrics used for this agent are hit, mean reciprocal rank and precision. The agent's results demonstrate a mix of strengths and limitations. It obtained a perfect Hit@5 of 1 meaning that at least one truly relevant product was present among the top five retrieved results. The agent never missed the correct recommendation. The ranking of the retrieval results was suboptimal. The Mean Reciprocal Rank (MRR@5) is only 0.2, which implies that the first relevant result was, on average, at the fifth position. Precision@5 is 0.2, indicating that only 20% of the items in the top five list were relevant. Together these metrics reveal that while the correct item was always retrieved, it was usually in the bottom of the list. The Generation Agent can compensate for imperfect ranking by leveraging the relevant content wherever it appears in the retrieval set. Improving the retrieval ranking could be done.

Generation Agent Performance

The agent's results shows strong performance in producing a high-quality, helpful recommendation. It obtained BERT Score and ROUGE-L score of 1.0. It indicates that the generated recommendation captured all the key information present in the reference answer. The generation was rated with Faithfulness and Relevancy of 1.0. It means that the agent was entirely faithful to the retrieved evidence and fully relevant to the user's query. No hallucination were found. Contrastingly the BLEU score of the generated output was 0.0596, which is very low. BLEU is a precision-oriented metric that measures exact n-gram overlap with a reference text. The agent likely used different sentence structure than the reference answer leading to low n-gram overlap. This indicated that BLEU was not appropriate indicator of quality in this context. Overall, the generation agent demonstrated that it can produce a coherent, informative recommendation that is both relevant to the user and supported by evidence.

The results highlight the advantages of the generative agentic approach compared to traditional recommendation systems. Conventional recommender systems output a list of item suggestions, with no explanation provided to the user. This lack of interactivity and explainability has been recognized as a major limitation of traditional recommenders. This opacity can reduce user trust and engagement. Studies have shown that making recommendations explainable and conversational can greatly improve user satisfaction and trust in the system. The research produces a natural language recommendation with relevant explanation. This approach aligns with emerging conversational recommender systems in literature. Recent work has pointed out that traditional recommender systems struggle with

poor interactivity and explainability. The research is a concrete proof of this shift, leveraging a generation agent to deliver recommendations in conversational form. It is worth noting that LLM based recommenders have been reported to excel at producing explainable recommendations. This ability to generate relevant and fluent explanations is a new strength that traditional recommenders do not possess.

7 Conclusion and Future Work

The research was set out to determine how accurately can a Generative Agentic AI Framework produce faithful, relevant and conversational product recommendations by combing intent classification, retrieval and generation components. To answer this question, A novel Generative Agentic AI Framework was built with four agents. An Intent Classification Agent that was built on Logistic Regression, A retrieval agent with FAISS vector retrieval on MiniLM embeddings, A generation agent built on Quantized OpenHermes model. Preprocessing and Exploratory Data Analysis has been done on the Amazon 5-core product dataset. Each agent in the framework was evaluated. The intent classification agent attained 90.9% accuracy indicating excellent capability in correctly identifying the user's intent. The retrieval agent also performed effectively. It attained Hit@5 of 1.0, MRR@5 of 0.2, Precision@5 of 0.2. The generation agent generated recommendations that are both faithful and relevant to the user's query. Metrics gave a BERT Score of 1.0, Faithfulness of 1.0, relevancy of 1.0. The BLEU score for the agent was low. The research addressed several limitations of traditional recommender systems. This approach mitigates the cold start issue. The successful integration of intent classification, retrieval and generation agents demonstrates a promising proof of concept: an agent based AI framework can indeed generate faithful and relevant product recommendations in conversational form offer a richer user experience than conventional recommendation systems.

However, some **limitations** were observed. The product coverage was limited to a subset of Amazon categories, restricting the diversity of recommendations. Additionally, while OpenHermes was selected due to space and resource constraints, its generation quality was lower than that of larger state-of-the-art LLMs. The current Ranking Agent was implemented as a placeholder and did not influence retrieval ordering beyond FAISS similarity scores.

While the results are encouraging, there are several paths to further improve and extend this framework. One immediate extension is to enable **multi-turn conversational interactions**. In the current framework the user receives a recommendation based on a single query turn. Retrieval Ranking can be improved. This could involve enhancing the vector representations. Incorporating sophisticated retrieval techniques such as neural retriever models can be done. This will improve the hit rates and MRR metrics. Future enhancements can focus on personalization. This involves integrating user profiles, past interaction history and feedback into the agent's decision making. Such user adaptive behaviour would leverage the agentic framework's modularity. This framework can be deployed in real time scenarios using the Agentic AI framework such as LangChain.

In summary, the **Generative Agentic AI framework for product recommendation** shows considerable promise, and the planned future improvements aim to refine its accuracy, interactivity, and user-centric adaptability. By extending the dialogue capabilities, enhancing retrieval precision, and incorporating personalization and human feedback, this approach envisions moving closer to deployment-ready conversational recommender systems that can

significantly improve user experience. The work lays a foundation for next-generation recommenders that combine the strengths of structured retrieval algorithms with the intelligence of generative AI.

References

Ahmed, M.Z., Singh, A., Paul, A., Ghosh, S. and Chaudhuri, A.K., 2022. Amazon product recommendation system. *International Journal of Advanced Research in Computer and Communication Engineering*, 11(3), pp.333–351.

Ahmed, M.Z., Singh, A., Paul, A., Ghosh, S. and Chaudhuri, A.K., 2022. Amazon product recommender system using collaborative filtering and sentiment analysis. *International Journal of Advanced Research in Computer and Communication Engineering*.

Chen, Z., Liu, L. and Sun, Y., 2024. CoST: Contrastive quantization based semantic tokenization for generative recommendation. *arXiv preprint* arXiv:2404.14774.

Huang, X., et al., 2023. Recommender AI Agent: Integrating large language models for interactive recommendations. *arXiv preprint* arXiv:2308.16505.

Kim, J., et al., 2025. AgentRecBench: Benchmarking LLM agent-based personalized recommender systems. *arXiv preprint* arXiv:2506.21931.

Kinkar, K., Zore, A. and Kulkarni, P.V., 2021. Product recommendation system: A systematic literature review. *International Journal for Research in Applied Science and Engineering Technology*, 9(7), pp.3330–3338.

Latha, Y.M. and Rao, B.S., 2024. Amazon product recommendation system based on a modified convolutional neural network. *ETRI Journal*, 46(4), pp.633–647.

Lee, S., et al., 2023. LC-Rec: Integrating collaborative semantics into LLMs for recommendations. *arXiv preprint* arXiv:2505.19623.

Li, H., et al., 2025. ARAG: Agentic retrieval-augmented generation for personalized recommendation. In: *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*.

Li, W., Wu, K. and Tang, J., 2023. Chat-Rec: Towards interactive and explainable LLMs-augmented recommender system. *arXiv preprint* arXiv:2303.14524.

Liu, J., et al., 2023. Is ChatGPT a good recommender? A preliminary study. In: *Proceedings of the 1st Workshop on Recommendation with Generative Models*, ACM CIKM.

Meng, Z., Yi, Z. and Ounis, I., 2024. KERAG_R: Knowledge-enhanced retrieval-augmented generation for recommendation. *arXiv preprint* arXiv:2507.05863.

Naz, A., Khan, U.A., Sodhar, I.H., Buller, A.H. and Sodhar, J., 2022. Product recommendation using machine learning: A review of the existing techniques. *International Journal of Computer Science and Network Security*, 22(5), pp.523–530.

Wang, Y. and Zhang, M., 2025. What LLMs miss in recommendations: Bridging the gap between LLM and matrix factorization models. *arXiv preprint* arXiv:2505.20730.

Johnson, J., Douze, M. and Jégou, H., 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3), pp.535–547. doi:10.1109/TBDATA.2019.2921572. *(FAISS foundational paper)*

Reimers, N. and Gurevych, I., 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. doi:10.18653/v1/D19-1410. *(Sentence-Transformers model family)*

Izacard, G., et al., 2022. Towards a unified agentic framework for retrieval-augmented generation. *arXiv preprint* arXiv:2208.03299.

Jianmo Ni, Jiacheng Li, Julian McAuley
Empirical Methods in Natural Language Processing (EMNLP), 2019