International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Conference Proceedings: Volume: 05 ACME – 2026 DOI: 10.55041/ISJEM.ACME102
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

Recommendation System using Gen AI

Mrs. D. Sirisha [1]
Assistant Professor, Department of AI&DS
Annamacharya Institute of Technology and Sciences, Tirupati – 517520, A.P.
sirishaaids@gmail.com

M Syam Prasad [2]
Department of AI&DS
Annamacharya Institute of Technology and Sciences, Tirupati – 517520, A.P.
molakasyamprasad2004@gmail.com

T Sree Lakshmi [3]
Department of AI&DS
Annamacharya Institute of Technology and Sciences, Tirupati – 517520, A.P.
tharugusreelakshmi@gmail.com

N Sukumar [4]
Department of AI&DS
Annamacharya Institute of Technology and Sciences, Tirupati – 517520, A.P.
nsukumar2212@gmail.com

S Rahul [5]
Department of AI&DS
Annamacharya Institute of Technology and Sciences, Tirupati – 517520, A.P.
surakanirahul@gmail.com

ABSTRACT **—** Recent advancements in the field of Generative Artificial Intelligence "(Gen AI) and Large Language Models (LLMs) have made it possible to improve the performance of modern recommendation systems. The conventional approaches employed for designing recommendation systems, such as collaborative filtering and content-based filtering, have some limitations, such as the cold-start problem, data sparsity, and lack of diversity in recommendations. These limitations have led to the poor performance of recommendation systems, especially in situations where there is a lack of interaction data and new users and items are being added to the recommendation system. To overcome these limitations, a new recommendation system has been proposed in this project based on Generative AI, which utilizes the advantages of the Google Gemini 2.5 Flash Large Language Model to design intelligent recommendations and improve the performance of the recommendation system.

The proposed recommendation system has been designed and developed as a web-based application using the Django framework, which allows users to interact with the recommendation system and utilize the advantages of the Google Gemini 2.5 Flash Large Language Model to design intelligent recommendations. Users can input data to the proposed recommendation system in the form of recommendation scenarios, and the proposed system can process the input data to design relevant recommendations and insights using the advantages of the Google Gemini 2.5 Flash Large Language Model. The combination of concepts related to recommendation systems and generative AI has assisted the proposed recommendation system in improving the performance, including the design of diverse recommendations and overcoming the limitations associated with the cold-start problem and data sparsity.

Keywords — Generative Artificial Intelligence, Large Language Models, Gemini Flash LLM, Recommendation Systems, Query Processing, Context Understanding

I. INTRODUCTION

In the current software ecosystem, program behavior emerges from the complex interplay of software requirements, data flow, and the users of the software. Understanding the software's functionality, data processing, and how that data can be leveraged to provide valuable outputs is the secret to developing intelligent and efficient software. Consider online shopping, entertainment, and web services—recommendation systems have become critical to assisting users in discovering information or products that are of utmost importance to them. While traditional recommendation systems such as collaborative filtering and content-based

filtering are helpful, they are not always the best at providing high-quality recommendations. Cold starts, data sparsity, and a lack of diversity have been long-standing challenges.

Gen AI and large language models are currently opening up the possibilities of creating smarter software. Gen AI can be used to analyze the search query of a user and retrieve relevant information or product recommendations and also give a recommendation based on the search query environment. It knows the context and has optimized recommendations regarding the same. This paper will refer to an online smart recommendation software. The author uses Gen AI in the proposed architecture in conjunction with conventional recommendation systems. The tech stack consists of Django and Google Gemini 2.5 Flash generative AI model which is being accessed via API. Gemini 2.5 Flash is an attention-based Gen AI model based on a transformer architecture. It helps to know the context of the user and applies a sparse mixture-of-experts approach to the data processing and make context-sensitive recommendations. With this approach, users are able to build context-sensitive search queries that capture numerous situations to build smart suggestions and derive meaningful information.

II. RELATED WORK

Recommendation systems have become very important in the digital world today, whereby they give end-users personalized recommendations depending on the interests and preferences. The collaborative and content-based filtering are the most well-liked recommendation systems. Collaborative filtering is based on determining the resemblance of users whereas content-based filtering is based on the analysis of the characteristics of the items and the preferences of the user. However, these two are popular with certain weaknesses like cold-start problem, lack of diversity and sparse user interactions that could reduce the effectiveness of the recommendation systems. To enhance the performance of the recommendation systems however, hybrid systems have been established where two or more algorithms are merged to enhance performance. In addition, machine learning and deep learning have been introduced in the last few years to improve the performance of the recommendation systems. However, most recommendation systems are based on user interactions, which might not be enough for the recommendation systems to understand the context of the query. However, with the advent of Generative AI (GenAI) and Large Language Models (LLMs), new avenues have been opened for developing smarter recommendation systems. The newly developed recommendation systems, such as Gemini AI , can interpret user queries and provide context-based recommendations. The newly developed recommendation systems do not need user interactions, as they can provide recommendations based on user queries. The newly developed recommendation systems have opened new avenues for developing smarter recommendation systems, and the newly developed system is based on the Gemini 2.5 Flash large language model and Django, which can provide intelligent recommendations and insights using GenAI.

III. METHODOLOGY

The system is designed as a Generative AI-powered recommendation and intelligent query processing platform, with a Large Language Model (LLM) for generating insightful answers to user queries. The process flow is designed in a sequence, as depicted in the diagram: user interaction, LLM processing, and output generation.

1.System Overview

The aim is to develop a recommendation system using Generative AI, designed using Django. The users interact with the system using a web interface, where they enter their contextual queries. The Django application relays these queries to the Gemini 2.5 Flash Large Language Model using the Google Generative AI API.

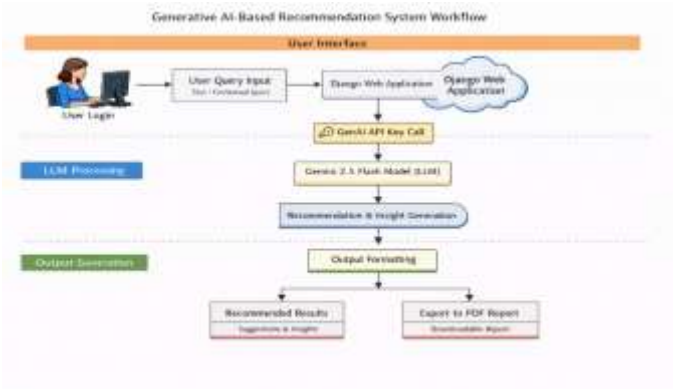

Fig 1 : System Architecture

2.User Interface and Authentication

Django is used to design the user interface web interaction. The users are able to create and log in to the recommendation system. Once they are logged in, they will be redirected to the dashboard and can make queries concerning various recommendation contexts. These questions are handled as contextual queries and accepted as natural language inputs by the Django application and the recommendation process begins.

3. GenAI API Communication

Upon receiving a user query, the Django application calls the layer of Generative AI using a GenAI API Key. The system is connected with Google Generative AI API to get access to the Gemini model. The environment variables load the API Key, which ensures the security of both Generative AI interface and the application.

This API integration allows the system to transmit user queries to the large language model and retrieve responses. The API integration allows the system to tap into the powerful features of the Gemini model without having to develop and train enormous neural networks from scratch.

4. Large Language Model Processing

The Gemini 2.5 Flash large language model, developed using the transformer model, is used for processing. The transformer model enables the model to understand contextual relationships and generate well-articulated responses to user queries. The Gemini model processes the user query and provides intelligent outputs like recommendations and insights.

Prompt-based generation is employed, where the user query and system instructions are combined to generate outputs. This method enables the system to understand the context of the recommendation scenario and provide outputs that improve the overall quality of recommendations. The Gemini model is integrated through the Generative AI API.

5. Recommendation and Insight Generation

Once the query is done, the Gemini model will produce output in the form of recommendations, insights, or well-structured answers to the context in question. These outputs are further sent back to the Django application which in turn processes them and structures them to be displayed. This step indicates how generative AI can provide smart recommendations without having to utilize the established cooperative or content-recommendation methods.

6.Output Formatting and Report Generation

The last stage is that of the Django application which formats the response that has been generated to be displayed to the user. The system also places the recommendations in a well-structured and easy to read format to enhance easy usability. The system has as well enabled the users to export the generated recommendations as PDF report that can be downloaded and stored in archives to be used in future. This adds some level of utility to the system as it can be used more efficiently in the process of analysis and documentation.

IV. PERFORMANCE ANALYSIS

The study aims to determine the efficiency of the Generative AI-based recommendation system in three significant areas: its performance, response time, and usability for generating smart recommendations according to user questions. The application is developed as a web application using the Django framework, which connects to the Gemini 2.5 Flash model using the Google Generative AI API.

A. Query Processing Accuracy

Query processing accuracy refers to the accuracy of the application in understanding the user query and generating accurate recommendations. The application, using the Gemini 2.5 Flash model, can process user queries using natural language processing, indicating its potential for generating accurate recommendations as long as the query clearly states the problem to be solved.

B. System Response Time

System response time refers to the time taken by the application to generate recommendations after receiving a query from the user. The response time depends on the Generative AI model, which is hosted in the cloud and uses the query to generate recommendations.

C. Context Understanding Capability
The application uses a transformer-based large language model to understand the context of the user query, generating accurate recommendations without relying on user behavior.

D. Usability and User Interaction
The application is designed as a user-centric application, enabling user registration, login, question posing, generating recommendations, and exporting the results as a PDF document.

E. Overall System Performance
The application aims to provide an efficient and intelligent recommendation system according to user questions, addressing some limitations associated with traditional recommendation systems.

V. RESULTS AND DISCUSSION

In the study, a Generative AI-based recommendation system was implemented and tested using various user queries to determine whether it can provide effective recommendations. The system processed the user queries using a Django web interface for natural language inputs, while it produced effective outputs using the Gemini 2.5 Flash, a large language model. The study shows that the system understands the context of the questions and provides effective recommendations for the specific situation.

From the experiments conducted, it is evident that the system can generate recommendations without relying on traditional collaborative and content-based filtering. This helps to address some of the common problems, such as cold-start and sparsity, in the system. The web interface makes it easier for the user to interact with the system and obtain structured outputs. The study shows that the proposed system can effectively generate intelligent recommendations using Generative AI.

VII. CONCLUSION

We developed a recommendation system that employs Generative AI to provide intelligent recommendations based on user queries. The project developed is a web application that employs the Django framework and integrates with the Gemini 2.5 Flash Large Language Model through the Google Generative AI API.

The results of this study indicate that the recommendation system can comprehend natural language queries and provide contextual recommendations". Unlike other recommendation systems that rely on user data, this system employs Generative AI to overcome the cold start problem and sparsity of user data. The project developed has shown that Large Language Models can be employed to enhance recommendation systems and improve intelligent decision-making.

REFERENCES

[1]. Talaei Khoei, T., & Kaabouch, N. (2023). Machine Learning: Models, Challenges, and Research Directions. Future Internet, 15(10), 332. [2]. Wang, X., Zhao, Y., Qiu, C., Hu, Q., & Leung, V. C. (2024). Socialized Learning: A Survey of the Paradigm Shift for Edge Intelligence in Networked Systems. IEEE Communications Surveys & Tutorials.
[3]. Torkashvand, A., Jameii, S. M., & Reza, A. (2023). Deep learning-based collaborative filtering recommender systems: A comprehensive and systematic review. Neural Computing and Applications, 35(35), 24783-24827.
[4]. Javed, U., Shaukat, K., Hameed, I. A., Iqbal, F., Alam, T. M., & Luo, S. (2021). A review of content-based and context-based recommendation systems. International Journal of Emerging Technologies in Learning (iJET), 16(3), 274-306.
[5]. Fkih, F. (2022). Similarity measures for Collaborative Filtering-based Recommender Systems: Review and experimental comparison. Journal of King Saud University-Computer and Information Sciences, 34(9), 7645-7669.
[6]. Çano, E., & Morisio, M. (2017). Hybrid recommender systems: A systematic literature review. Intelligent data analysis, 21(6), 1487-1524.