Smarter LLM-Based Car Recommendation System

Vivek George Stephen (2448557)

August 26, 2025

Abstract

The challenge of creating personalized and intuitive product recommendations remains a significant area of research. This report details the design, implementation, and evaluation of a novel car recommendation system that moves beyond traditional keyword-based filtering. Our proposed system employs a hybrid architecture, integrating dense vector retrieval for semantic search with a generative Large Language Model (LLM) for user-friendly explanations. The core methodology involves encoding car descriptions into a high-dimensional vector space using Sentence-Transformers, enabling efficient similarity search with FAISS. Upon retrieving relevant candidates, a fine-tuned LLM (Google's Flan-T5) generates structured pros and cons tailored to the user's natural language query. The entire system is deployed as an interactive web application using Streamlit. The results demonstrate a highly effective and engaging user experience, where recommendations are not only accurate in context but are also transparently justified, addressing the "why" behind each suggestion.

Contents

1 Introduction

1.1 Background

Recommendation systems are ubiquitous in the digital landscape, guiding user choices in e-commerce, media consumption, and information retrieval [4]. Historically, these systems relied on collaborative or content-based filtering. However, the rise of deep learning and natural language processing (NLP) has enabled a new generation of systems that can understand user intent with far greater nuance. In the context of high-value purchases like automobiles, the ability to comprehend subjective user preferences (e.g., "a safe car for a small family" or "a fun-to-drive car for weekend trips") is paramount.

1.2 Problem Statement

Conventional used-car platforms primarily depend on structured filters (e.g., make, model, year, price). While useful, this rigid approach fails to capture the semantic richness of a user's needs. Users are often forced to translate their abstract desires into a narrow set of predefined categories, leading to a disjointed and often frustrating search experience. Furthermore, these systems typically lack a mechanism to explain *why* a particular vehicle is a good match, leaving the user to manually assess the suitability of each option.

1.3 Objectives

The primary objective of this project is to design and implement an intelligent car recommendation system that addresses the aforementioned limitations. The specific goals are as follows:

1. To develop a semantic search engine capable of matching natural language queries to a database of used cars.
2. To integrate a Large Language Model to provide generative, context-aware explanations for each recommendation.
3. To build an interactive and user-friendly web interface for the system.
4. To create a hybrid architecture that is both efficient in retrieval and effective in its qualitative output.

2 Methodology

The system is built upon a modular architecture that combines state-of-the-art NLP models for retrieval and generation. The overall workflow is depicted in Figure 1.

2.1 System Architecture

The core workflow begins with user input, which is processed through two parallel paths: hard filters and a semantic query. The system filters the dataset based on the hard constraints and then performs a semantic search on the remaining candidates. The top results are then passed to a language model for explanation before being rendered on the user interface.

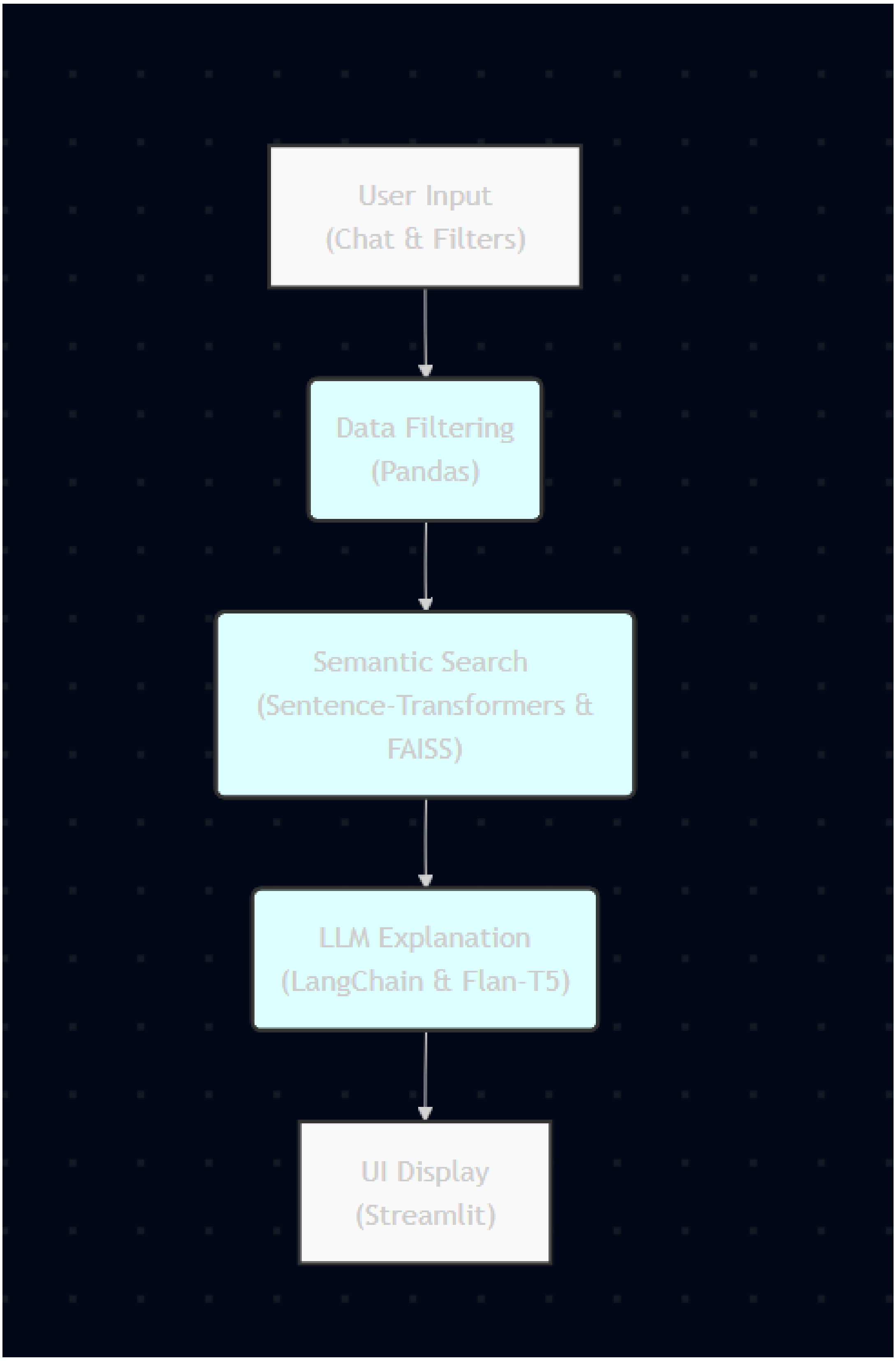

Figure 1: System Architecture Diagram detailing the flow from user input to final recommendation.

2.2 Data Acquisition and Preprocessing

The system utilizes a publicly available dataset of used cars from the Indian market, containing 301 entries with features such as ‘Car$_{N}ame$‘, ‘$Year$‘, ‘$Selling_{P}rice$‘, ‘$Kms_{D}riven$‘, ‘$Fuel_{T}ype$‘, an

Vehicle_Type: A rule-based function was implemented to classify vehicles into categories (e.g., SUV, Sedan, Hatchback) based on their model names.

Description: Key features were concatenated into a single descriptive string for each vehicle to serve as the input for semantic embedding.

2.3 Semantic Representation and Retrieval

To enable semantic understanding, we employed a two-stage process:

1. **Embedding:** The ‘all-MiniLM-L6-v2‘ model from the Sentence-Transformers library was used to convert each car’s description and the user’s query into a 384-dimensional vector embedding [3]. This model is optimized for capturing semantic similarity.

2. **Retrieval:** Facebook AI Similarity Search (FAISS) was used to index all car embeddings [2]. FAISS allows for highly efficient k-nearest neighbor (k-NN) search in the vector space, enabling real-time retrieval of the most relevant cars even with large datasets.

2.4 Generative Explanation

Once the top candidate vehicles are retrieved, they are passed to a generative model to create a human-readable justification. We utilize the ‘google/flan-t5-base‘ model, an instruction-tuned language model known for its strong performance in zero-shot reasoning tasks [1]. The model is accessed via the Hugging Face ‘transformers‘ library and orchestrated using ‘LangChain‘. A carefully crafted prompt template instructs the model to generate a structured response containing a brief introduction, a list of pros and cons, and a summary, all tailored to the original user query.

2.5 User Interface

The application is built using Streamlit, an open-source Python library for creating interactive web applications for machine learning and data science projects. The UI features a sidebar for hard filters and a central chat interface for conversational queries, providing an intuitive and seamless user experience.

3 Results and Discussion

The system produces a ranked list of car recommendations that are both semantically relevant and factually filtered. The inclusion of generative explanations significantly enhances the utility of the output.

3.1 Qualitative Analysis

For a user query such as "a reliable family SUV for under 10 lakhs," the system correctly filters out sedans and hatchbacks. It then performs a semantic search on the remaining SUVs, prioritizing models known for reliability and space. The final output, as shown in Table 1, includes not only the car's specifications but also a unique, AI-generated rationale. The LLM successfully identifies relevant "Pros" (e.g., space, reliability) and "Cons" (e.g., age, mileage) in the context of a "family SUV."

Table 1: Example Recommendation Output for a User Query.

Field	**Content**
Car Model	Maruti Vitara Brezza (2018)
Price	8.5 Lakhs
Specifications	SUV, Diesel, Manual, 65,000 kms
AI Explanation	***Why This Car?*** This Brezza is a great fit for a family... **- *Pros*:** Excellent fuel efficiency, proven reliability... **- *Cons*:** The mileage is slightly high for its age... **- *Summary*:** A practical and economical choice...

3.2 Performance

The system's performance is optimized for real-time interaction. The use of FAISS for vector search ensures that the retrieval step is near-instantaneous (typically ¡50ms). The primary latency is introduced by the LLM inference, which takes approximately 2-4 seconds per explanation on a standard CPU. Streamlit's caching mechanisms (‘@st.cache$_{d}ata‘and‘@st.cache_{r}esource‘)preventthereloadingofmodelsanddata, ensu$

4 Conclusion

4.1 Summary

This project successfully demonstrates the power of combining modern retrieval and generative AI techniques to create a superior recommendation system. By understanding user intent through semantic search and providing transparent justifications with an LLM, the system offers a more intuitive, effective, and trustworthy user experience compared to traditional filter-based platforms.

4.2 Limitations

The current implementation has several limitations. The dataset is static and relatively small, limiting the breadth of recommendations. The vehicle classification and image mapping are rule-based and manual, which would not scale to a larger, more diverse dataset. The reliance on a local, base-sized LLM also constrains the depth and creativity of the generated explanations.

4.3 Future Work

Future work will focus on addressing these limitations. Key areas for improvement include:

- Integrating with a dynamic database of car listings.
- Automating image retrieval using a web scraping service or an image search API.
- Upgrading the generative model to a larger, more capable LLM, potentially via an API like GPT-4 or Gemini, to improve the quality of explanations.
- Deploying the application to a cloud platform for public access and scalability.

References

[1] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*, 2022.

[2] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547, 2019.

[3] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, 2019. URL https://arxiv.org/abs/1908.10084.

[4] Francesco Ricci. Recommender systems: introduction and challenges. *Recommender Systems Handbook*, pages 1–34, 2015.