National College of Ireland

Large Language Model Driven User Cold Start Recommendations

MSc Research Project
Data Analytics

Shubham Dalvi
Student ID: x23268051

School of Computing
National College of Ireland

Supervisor: Teerath Kumar Menghwar

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Shubham Dalvi
Student ID:	x23268051
Programme:	Data Analytics
Year:	2025
Module:	MSc Research Project
Supervisor:	Teerath Kumar Menghwar
Submission Due Date:	9/15/2025
Project Title:	Large Language Model Driven User Cold Start Recommendations
Word Count:	5,131
Page Count:	27

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:	
Date:	9/15/2025

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):	

Large Language Model Driven User Cold Start Recommendations

Shubham Dalvi
x23268051

Abstract

The advent of new and powerful LLM models has given rise to innovative approaches to using large language models and recommendation systems. These models have been trained on a large corpus of data and world knowledge, which can be harnessed in recommendation systems. The user cold start problem is a prominent obstacle in recommendation systems and businesses to retain initial users. This paper presents a novel approach to addressing the user cold-start problem, generating a user profile that can be utilized to provide recommendations for cold-start users. A comprehensive evaluation framework with 6 different approaches is used in this paper to evaluate the recommendation system.

LLM(Large language models), Content Based Filtering, Collaborative Filtering, Cold Start Users, Prompting Techniques

1 Introduction

In the era where data is often treated as the backbone of making business decisions and growing the product, the volume of information generated and consumed daily continues to grow at an unprecedented pace. As user engages with online platforms, they are exposed to an overwhelming variety of content and choices. This can sometimes be a problem for the user to navigate their own preferences and intent. Recommendation systems have become essential for a personalized experience for users. Intentionally filtering and surfacing relevant items tailored to the user's preferences and intent has become an essential part of the user experience.

However, with the expansion of users and items comes a set of challenges, most notably the cold-start problem and data sparsity for new users who lack historical interaction with the platform from which they consume content. This hinders the ability of a recommendation model to infer user preferences and show relevant items to first-time users. Addressing this critical issue will not only improve user retention but also help in early engagement of users without much interaction with the platform.

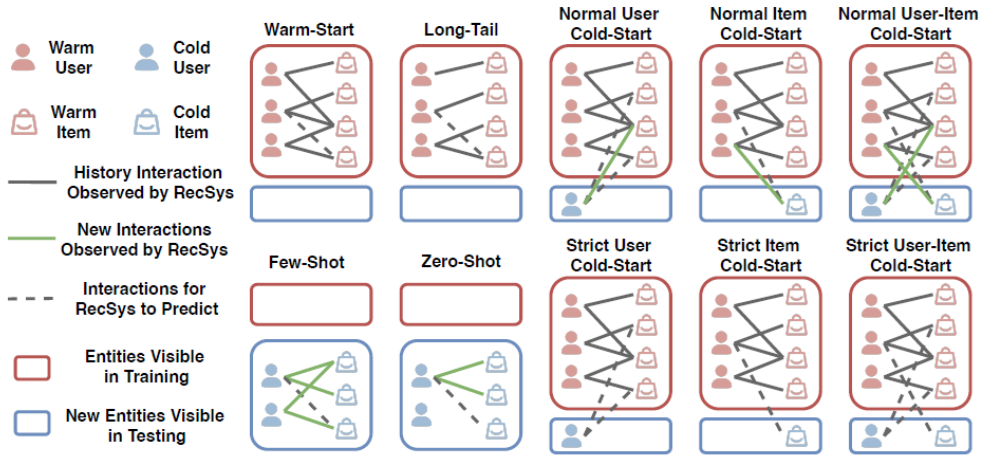

Figure 1: Cold Start Problem Zhang et al. (2025)

The above Figure 1 shows how different types of cold start problems exist. For users with no prior interaction, it's difficult to suggest items, but by using large language models to know the user preferences before the user interacts with any item is a novel way to tackle the user cold start problem. This instills the need to investigate how large language models can be used to interpret cold-start user problems. Large Language Models offer powerful means of understanding and representing the user's intent. Furthermore, addressing the cold start problem for new users brings greater value to the brand and ensures that the new users are retained and get good initial recommendations on the platform (Zhang et al.; 2025).

1.1 Research Questions

The primary research question guiding this study is:
'How can we effectively recommend relevant items to users with no prior interaction history known to the recommendation engine ?'

In addition, the following supporting research questions are also addressed:

How can LLMs be used to accurately infer true genre affinities from users' preferences?

How can users be seamlessly transitioned from content-based to collaborative filtering approaches as more rating data becomes available?

1.2 Navigation Guide

For quick navigation, this index summarizes each major section and its purpose:

1. **Abstract**: Problem, approach, and contributions in brief.
2. **Introduction**: Motivation, cold-start challenge, and research questions.
3. **Related Work**: Positioning against prior CB/CF, hybrid, meta-learning, and LLM-based methods.

4. **Research Methods**
   Specifications: Overall methodology, ethical considerations, experimental setup.
5. **Design**: System design decisions and data flow.
6. **Implementation**: Questionnaire, LLM weight generation, content-based and collaborative filtering pipelines.
7. **Evaluation**: Datasets, temporal split protocol, user profiles, and metrics (Precision@K, MRR, NDCG).
8. **Conclusion**: Findings, limitations, and future work directions.

2 Related Work

2.1 Content-based filtering and Collaborative filtering in Recommendation Systems

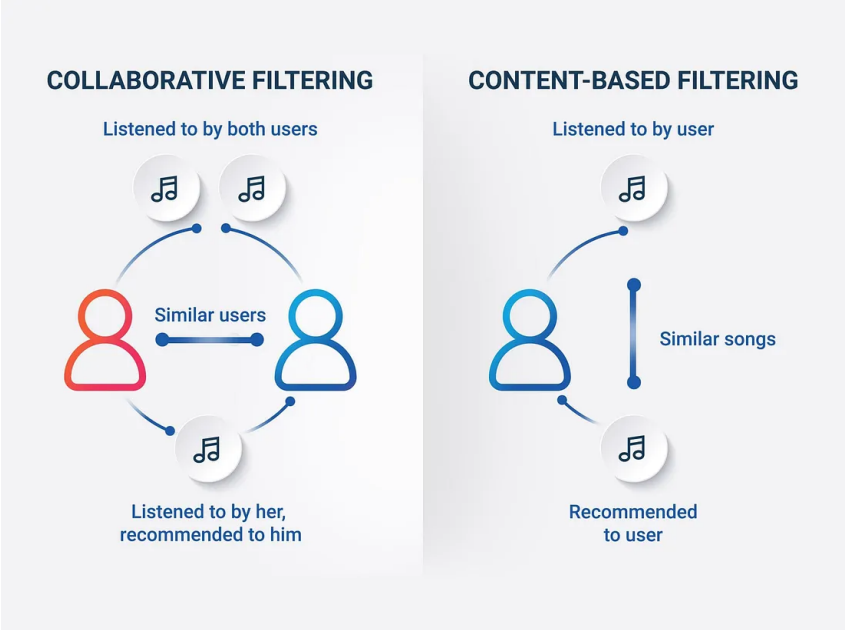

Figure 2: Content and Collaborative filtering Sheth (2023)

The two most widely used approaches to recommending items to end users are the collaborative filtering approach and the content-based filtering approach. Thorat et al. (2015) provides a comprehensive survey on both these methodologies and a hybrid approach that combines both collaborative filtering and content-based filtering. While there are some disadvantages or limitations or content-based filtering, like the cold start problem, sparsity of the rating matrix, and scalability issues with large data sets, and limitations for content-based filtering, namely over-specializations and recommending too similar items, the need for detailed item metadata with limited novelty and recommendations can be solved using a hybrid recommendation system. The hybrid strategies include choosing

between CF and CBF based on scenarios, using features from one as an input to another, adding new features from one method to improve another method and merging recommendations from both systems.

The use of a hybrid approach to tackle recommendation system problems, like using a fact-aware multi-head mixture of experts model proposed in (Liu, Zhang and Long; 2025), recommending the next sequential item. Using the multifaceted nature of items like movie genre, starring actors, etc, and capturing the complexities in sub-embeddings and multi-layered attention layers to predict the next item is proposed in this paper, which is useful for warm start users. Another similar study proposes the use of warm domain knowledge to tackle cold domain recommendations. This paper has a comprehensive summarization of existing cross-domain recommendation approaches, namely using single target CDR, multi-domain recommendations, dual target CDR, and multi-target CDR, which is achieved by using content-based transfer or embedding-based transfer, or rating pattern-based transfer (Zhu et al.; 2021).

The hybrid approach of using separate neural networks for expertise in warm and cold domains and then gradually cold to warm domains is proposed in (Zhang et al.; 2023). This approach uses two neural networks, which handle cold start users using a different neural network and handle warm users using a different neural network. Users are represented using a mix of user profile embedding, user action embedding, and user group embedding. This representation helps the cold start expert learn from the form warm start expert only when the predictions are worse using dynamic knowledge distillation. Using dynamic recommendations based on most frequently clicked and most frequently pushed items to increase user engagement and adjusting the recommendation list in real time is also a good approach to take while recommending to new users (Liu et al.; 2018).

2.2 Meta-Learning Frameworks for Recommendation Systems

Using neural networks in recommendation systems is a good way to tackle the cold start problem. Using a personalized version of a shared neural network model instead of using embeddings per user is how MeLU paper (Lee et al.; 2019) approaches solving the cold start problem. It treats each user as a separate task and uses item features only, like genre, directors, and tags, to predict preference for a particular user.

One of the problems for recommending Cold Start users is popularity bias, which is tackled in a unique way in (Luo et al.; 2024). This approach learns to reweight by dynamically adjusting how much each training sample contributes to the loss during meta learning. This helps in correcting the popularity imbalance by giving focus to less popular items during meta learning updates. This, in turn, helps in better generalization for unseen items and robustness against popularity skew.

Another meta learning framework for rapidly adapting to new users' interactions is discussed in (Lee et al.; 2019). That takes user and item metadata and predicts the ratings. Each user is treated as a separate task, and MeLU learns model parameters that can adapt fast to a new user with few updates. It performs local sub-local updates for each user's item history and global updates across tasks to generalize. Going one step further and using two neural meta learning architectures (Vartak et al.; 2017), this paper uses linear weight adaptation and nonlinear bias adaptation neural nets to tackle the cold start recommendation problem for items.

2.3 Graph Convolution network in Recommendation engines

Representing a relationship between a user and an item can be best done by using graph neural networks. One of the use cases of graph neural networks and recommendation systems is discussed in (Liu et al.; 2020). The author proposes to represent a user's items and attributes as notes in a graph and introduces an attribute-aware attention mechanism to assign different importance to each attribute and handle missing attribute values more naturally. The edges in this graph represent user-item interaction and item attribute associations. This resulted in great performance both in dense and sparse datasets due to the unified graph of user-item and attributes, which enabled richer interaction modeling. This model can perform better in gold start problems as well because of the item attributes; even though the item has a few interactions, it can still be recommended. Also, users with limited history can still benefit from indirect connections in the graph, hence boosting cold start performance for this model.

2.4 Large Language Models for Recommendation Systems

With the growing popularity of LLMs in this world of ever-growing generative pre-trained transformers (GPTs), there has been increasing interest in harnessing the power of world knowledge in LLMs to enhance recommendation systems. In contrast to traditional recommendation systems, LLM-based models capture contextual information, user queries, item descriptions, and data more effectively.

One of the ways of using LLM's to enhance a recommendation system and search engine is using a keyword-based retrieval system mentioned in Kieu et al. (2025), which uses a keyword-based retrieval and candidate list of items generation system to recommend items based on a few keywords that the user gives to the system. It also uses prompting templates like zero-shot prompting and few-shot prompting, helping it make a personalized and better ranking of the candidate list.

The paper survey on large language models provides a comprehensive guide on how LLMs can be used with recommendation systems or as recommendation systems.

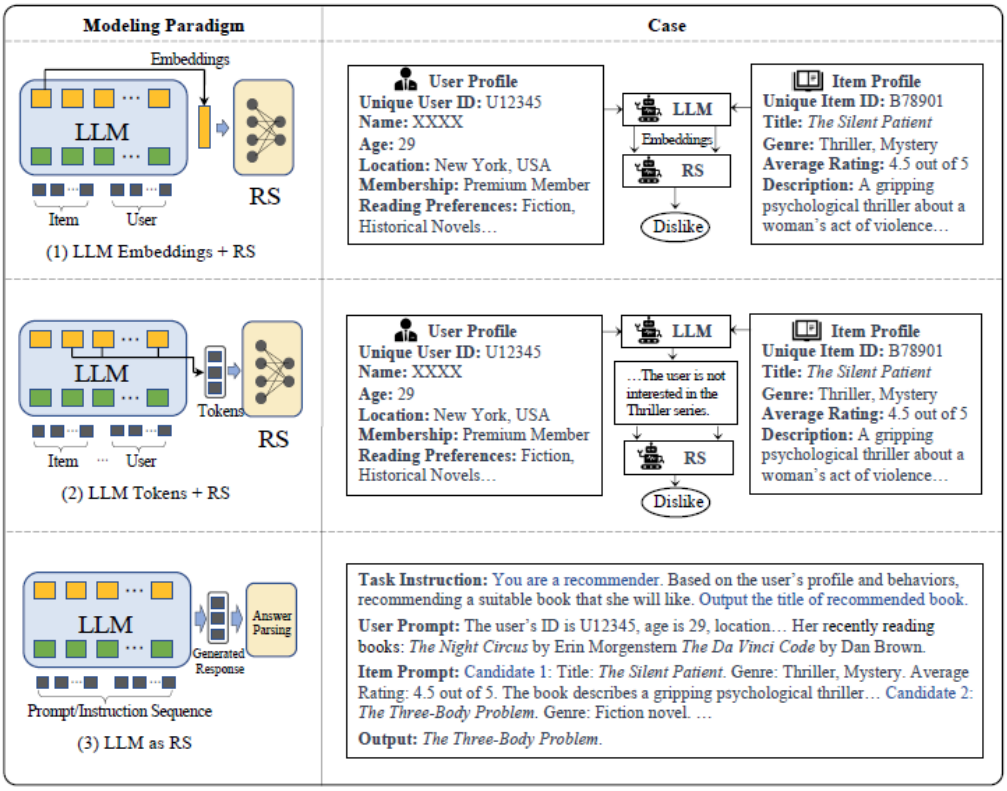

Figure 3: Three representative modeling paradigms of the research for large language models on recommendation systems. (Source: (Wu et al.; 2024))

The above figure shows how LLMs can be used in recommendation systems to enhance different aspects. One of the ways is to use LLM embeddings + RS, a paradigm that uses LLMs as a way to generate user, item feature-aware embeddings, which can be further used with the Recommendation system models to perform recommendation tasks.
Another approach of using LLM tokens + RS, which is used to predict the user's rating of an item and then use these ratings to help recommendation systems make decisions. Unlike the previous two paradigms, LLM can also be directly used as a Recommendation system to generate recommendations using profile description, behavior, and task instructions. (Wu et al.; 2024).

A unique way of identifying or generating item descriptions using LLMs instead of relying on manual content is explored in Acharya et al. (2023). LLMS are not directly used to recommend, but instead to generate a textual description for items using only item names in this approach. These descriptions are embedded via BERT and used in conjunction with the item ID to predict future user interactions. This kind of approach acts as a content-based filtering enhancer, replacing manually scraped content with automated LLM-generated metadata.

(Zhang et al.; 2025) Discusses how LLMs can be used for multiple purposes and recommendation systems, like using it as a knowledge and answer, or to use multi-step prompting to generate recommendations, or to do relational augmentation. LLMS can be tuned with an instruction set and made compatible to generate recommendations

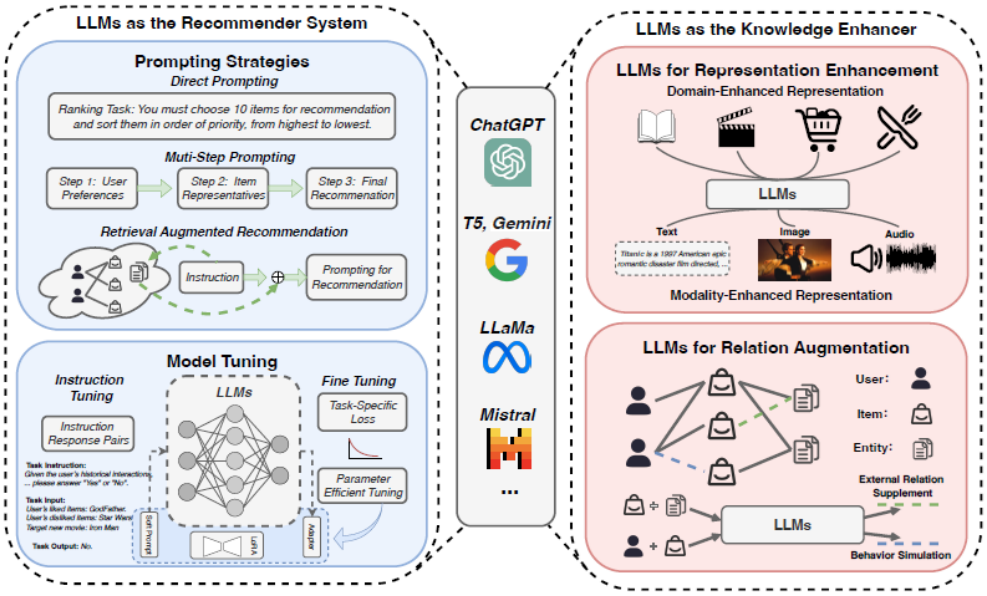

Figure 4: Different categories of methods for utilizing the world knowledge from LLMs. (Zhang et al.; 2025)

The figure shows how LLMs can be used in recommendation systems in two majorly different ways.

In another paper on recommendation systems combined with rag (Retrieval Augmented Generation) explains how the problem of popularity bias and cold start in recommendations can be effectively solved using diverse information without having strict relevance to the user preferences (Contal and McGoldrick; 2024). These results demonstrate that incorporating diversity and relevance into retrieval strategies results in the superior performance of LLMs and recommendation systems. However, adding a layer of user preferences before generating relevant and diverse recommendations supercharges the performance of the recommendations. Hence, the novel approach of using a questionnaire to determine user preferences, combined with retrieval-augmented generation and enhanced feature embedding, is worth investigating.

2.5 Evaluating recommendation systems and data splitting

Evaluating a recommendation system and having a correct data splitting strategy is the key to good recommendation system evaluation. Choosing the right splitting strategy can significantly change the model ranking and make the results across papers non-comparable, even for the same data sets. Different strategies for data splitting are explored in Meng et al. (2020). Taking inspiration from this paper, splitting strategies like country splitting for collaborative filtering leave one splitting strategy for content-based filtering testing. It's used in this so that there is no Data spillage, accidental Leakage of data in the tests.

Krishnabalan et al. (2022) Compass the strengths and weaknesses of content-based and collaborative filtering algorithms in various business contexts. Their analysis high-

lights that content-based methods excel in scenarios with limited user data and niche preferences, while collaborative filtering uses community-wide patterns to expose users to novel items. Deutschman (2023) presents a comprehensive taxonomy of evaluation matrices for recommendation systems, namely similarity-based measures, predictive accuracy metrics, and ranking-based measures. This work emphasizes the importance of aligning technical evaluation with business outcomes such as click-through rate and revenue uplift. The article also advocates incorporating novelty, diversity, and coverage for testing recommendation systems holistically.

Evidently AI (2025) Outlines 10 key metrics for assessing recommending and ranking systems, categorizing them into predictive quality, ranking quality, behavioral metrics and business metrics. This guide offers clear definitions, use cases, and limitations of each metric, such as MAP, MRR, NDCG, Diversity, and Novelty. By integrating offline metrics with online monitoring, the goal is to ensure both algorithmic performance and positive user experience. Meshram (2023) Explores how recommendation algorithms can be adapted to particular domains, assessing the tradeoffs between personalization, computational cost, and scalability. The study provides insights into how hybridized models or advanced modeling techniques can mitigate cold start and sparsity issues while improving relevance and diversity.

3 Research Methods & Specifications

The proposed novel solution combines the world knowledge of large language models to generate the user profile and use it to generate recommendations based on content-based filtering and collaborative filtering, Driven by the questionnaire the user is given before generating the recommendations.

3.1 Methodology

When the cold-start user arrives on the platform, they will be presented with a well-crafted questionnaire, which will help in generating data regarding the user preferences. This data will be fed into the LLM to categorize the user into particular categories of genres. Using this categorization and relevant information about potential items, the user will be shown recommendations. Based on the user profile, they will be shown a few potential candidates for getting explicit ratings on those candidates from the user. These ratings will then be used to generate collaborative filtering recommendations based on similar users.

3.2 Ethical Considerations of the Research

This project implementation does not require external inputs from real users. The data for this research implementation is publicly available and can be used without any special permissions. Personally identifiable information is not retrieved or asked to the user using the platform. Instead of having real users, the model is tested and trained on the data from a publicly available dataset.

4 Design

4.1 System Architecture Framework

The recommendation system architecture is designed to progressively take inputs from the user and generate recommendations based on the user's implicit and explicit signals regarding their genre preference. The architecture uses client server architecture design pattern with RESTful API communication, enabling clear separation of concerns between user interfaces and recommendation logic.

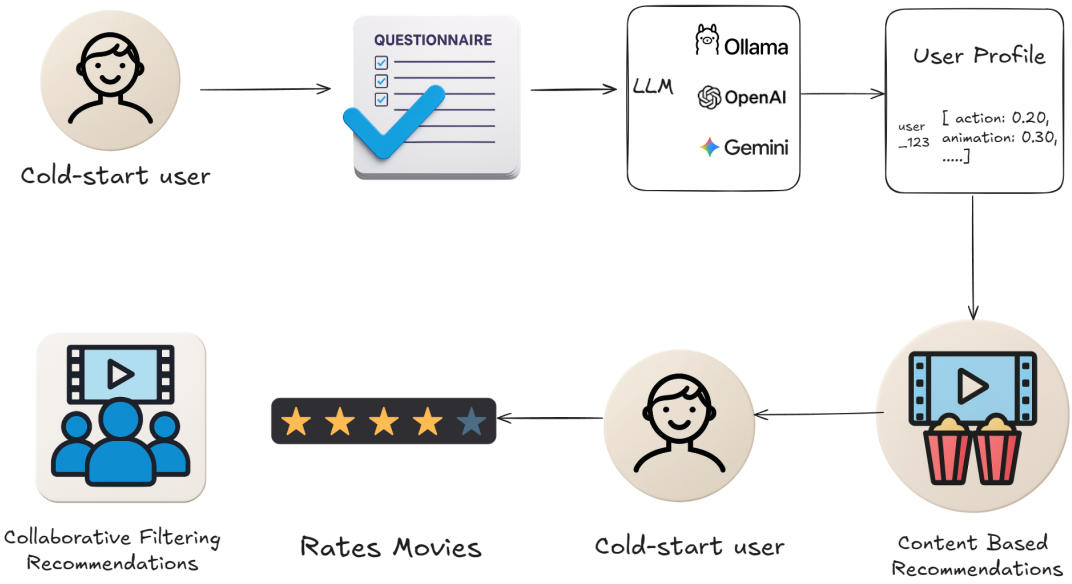

Figure 5: System Architecture Overview

The architectural framework integrates 3 core recommendation paradigms. One is user preference generation using structured questionnaires, content-based filtering for immediate cold-start recommendations, and collaborative filtering for enhanced personalization. This progressive architecture ensures continuous recommendation quality improvement As the user interacts with the system.

4.2 Requirements and Constraints

Functional Requirements

The system is designed to address multiple functional requirements for effective cold start recommendations, such as generating meaningful recommendations for users with zero prior interaction history with the items. retrieving user preferences implicitly and explicitly via strategically tailored questionnaires. Enhancing recommendations through additional user preferences signals by rating movies. The system also assures transparency in the recommendation generation by showing the explanation and similarity scores for things based on which it has given recommendations.

Technical Constraints

The key technical constraints that influenced the architectural design decision were generating real-time recommendations with an acceptable response time and limited webpage interactions. Integrating with the existing movie lens data set and also reliably integrating large language model API's for preference analysis.

4.3 Algorithmic Design Framework

Hybrid Recommendation Algorithm Design

The core design employs a novel hybrid approach combining large language model-enhanced preference inference with traditional recommendation system techniques. The design overcomes the limitations of not being able to collect explicit user preferences for cold-start users.
The design uses prompt engineering techniques to infer the user preferences based on the questions and answers to the questions. This preference inference is done using the ability of large language models to analyze the responses and to infer user preferences that align and that do not align with explicit genre selection. This approach addresses the common scenarios when users' stated preferences differ from actual consumption patterns, enabling more accurate preference modeling for cold shut scenarios.

Progressive Recommendation Strategy Design

The recommendation system strategies are designed to implement a progressive approach that transitions the user from a cold start to a warm user by using collaborative filtering scenarios. The strategy begins with cold start-based filtering using inferred genre preferences and user profiles, progresses through strategic rating collection for preference validation and generates collaborative filtering recommendations by leveraging similar users' data.
This design ensures optimal recommendation quality for singing each stage while maintaining user engagement through transparent preference elicitation My recommendation explanation. This approach helps in balancing the exploration of user preferences with the exploitation of known preferences, enabling personalization for cold-start users.

4.4 Technical Framework Selection

Web Application Framework

The system uses a modern web application framework comprising React for front-end user interface development and FastAPI for back-end API services. The framework selection is done on the basis of the observability available in the FastAPI framework for API documentation and testing. This framework enables robust backend processing capabilities for recommendation generation. And the separation of concerns between back-end and front-end components built into the code structure helps make development, testing and deployment easy.

Data Processing Framework

The data processing part utilizes pandas and numpy for efficient numerical computation and data manipulation, enabling near real-time recommendation generation and calculation with acceptable performance. The framework supports both content-based similarity computation and collaborative filtering analysis. By pre-computing the movie genre matrix by one-hot encoding helps reduce real-time processing time significantly. This framework prioritizes modularity and extensibility, enabling seamless integration of the data source and the algorithmic components.

Artificial intelligence Integration Framework

The Artificial Intelligence Framework integrates large language model capabilities through OpenAI API integration with the recommendation algorithm. This hybrid framework enables sophisticated personalized inference while maintaining computational efficiency for real-time recommendation scenarios for cold start users. The framework design includes an error-handling fallback mechanism to ensure a robust operation despite external service dependencies.

4.5 System Quality Attributes

Performance Design Considerations

The system design prioritizes response time optimization for user interaction while maintaining recommendation quality. Performance design includes efficient data structures for similarity computation, optimized database queries for movie information retrieval, and caching strategies for frequently accessed recommendation components.

Reliability and Scalability Design

The architectural design incorporates reliability mechanisms, including comprehensive error handling, graceful degradation for external service failures, and robust data validation processes. Scalability design considerations include stateless API design for horizontal scaling, efficient data access patterns, and modular component architecture supporting independent scaling of system components.

This comprehensive design framework establishes the foundation for effective cold-start recommendations while maintaining system reliability, performance, and scalability requirements essential for practical deployment scenarios.

5 Implementation

5.1 System Architecture Overview

The implemented recommendation system addresses the problem through a comprehensive web-based application. It integrates questionnaire driven preference elicitation with progressive recommendation generation. the system uses RESTful APIs back end coupled with React front end enabling seamless user interaction in real-time recommendation processing.

The implementation consists of four components: preference collection through structured questionnaires, LLM enhancement for underweight generation, Content based filtering for cold start users and collaborative filtering for users with sufficient rating history. The Movielens data set is or leveraging the movie information and collaborative signals while maintaining independent user profiles for personalized recommendations.

5.2 Preference Collection System

Structured Questionnaire Implementation

The system implements an 18-question preference elicitation framework that is designed to capture accurate user preferences beyond traditional genre selections. The questionnaire covers four critical preference dimensions, namely "pacing and visual preferences", "character complexity and thematic content", "temporal and narrative preferences", and "viewing behavior patterns". Each question has multiple responses designed to infer genre preference indirectly rather than through explicit genre selection.

The questionnaire systematically explores user preferences for character complexity (simple versus morally ambiguous), visual importance (special effects versus story focus), thematic moods (uplifting versus dark content), narrative structures (linear versus complex), temporal preferences (contemporary versus classic films) and social viewing patterns. This approach is valuable for understanding user preferences even when users themselves may not be explicitly aware of the genres they like.

Genre Selection and Validation

Prior to the questionnaire completion user is prompted to select preferred and disliked genres from a list of 19 genres taken from the MovieLens genre taxonomy. The system validates the genre selection, ensuring balanced input and preventing contradictory selections and interpretations by using correct prompting techniques to mitigate this risk.

5.3 LLM-Enhanced Genre Weight Generation

Intelligent Preference Analysis with Zero-Shot Prompting

The system uses OpenAI's GPT-3.5-turbo model with a sophisticated zero-shot prompting approach. This questionnaire responds and generates normalized genre waits for all 19 movie genres. The use of a comprehensive system prompt that instructs the model to perform holistic preference analysis by considering both explicit genre selection and implicit preferences revealed through cautionary responses that the user gives.

The system prompts inverse instructions that explicitly warn against over-relying on stated preferences, like:" Do not automatically assign the highest weight to preferred genres; instead, analyze the questionnaire response to determine if the genre truly matches the user test profile." This approach addresses the critical issues and edge cases where user-perceived preferences differ significantly from their actual viewing behavioral patterns. The prompt also includes concrete examples and detailed guidelines such as: " if a user selects documentary as preferred but the questionnaire response shows that they prefer fast-paced action, visually impressive effects, and a fantasy world, and minimal

dialogues, they likely would enjoy the action and fantasy genre more than documentary". This kind of prompting technique enables the LLM to perform nuance preference inference that goes beyond surface-level genre selection, producing normalized weight vectors that sum up to one across all genres for consistent mathematical processing.

Advanced Prompt Engineering and Response Validation

The sophisticated prompting technique to address each case, such as contradicting user inputs, helps tackle the misinterpretation problem for cold-start users. The validation framework implements three constraints on response validation from the LLM: structural validation (JSON format), mathematical validation (normalized weight summing up to 1) and semantic validation (weight distribution consistency with questionnaire responses). Failed validations trigger and retry mechanism to retry and generate a correct and valid response.

5.4 Content-Based Filtering Implementation

Genre-Based Similarity Computation

The content-based filtering system utilizes LLM-generated genre weights to compute movie recommendations through dot product similarity calculations. To make the recommendation near real-time, the system uses preprocessed MovieLens data with binary genre encoding for each movie, which indicates the presence or absence of the genre. The movie recommendation is generated by web computing the dot product between the user's normalized genre weight vector and each movie genre vector, producing a similarity score that reflects the user preference alignment.

The recommendation algorithm re-ranks these movies by similarity score and applies filtering to remove movies with insignificant genre information or low similarity scores. The system provides transparency in the recommendation process by showing the associated similarity scores on the front-end. This approach ensures that the user receives more movie suggestions that closely align with their inferred preferences, particularly effective for cold start scenarios with no reading history exists.

Recommendation Observability and Genre Analysis

The system has comprehensive observability features that provide users with detailed insight into the recommendation generation process and genre patterns. The interface displays genre analysis showing the frequency of each genre appearing in top 5, top 10, and top 20 recommendation lists, enabling users to understand how their inferred preferences translate into movie suggestions.

5.5 Movie Rating Collection System

Strategic Movie Selection for User Rating

Following the content-based recommendation system, the project implements the strategic movie rating collection phase to transition users from cold start to collaborative filtering scenarios. The system selects diverse movies from the content-based recommendation and popular MovieLenses titles, ensuring coverage across all different genres

and popularity levels. The user then rates the selected movies on a scale of one to 5 through an interactive interface that displays movie information and genre classifications.

The rating collection strategy tracks the number of ratings the user has given, because we need to satisfy the minimum threshold for the number of ratings a user has given to satisfy collaborative filtering recommendations. Ensures that sufficient ratings data for effective recommendation generation is collected while maintaining user engagement throughout the process.

Rating Data Integration and Processing

The user ratings are stored with comprehensive metadata, including rating timestamp, movie information, and user session data. Validation of rating ranges checks for duplicate entries is made to ensure data consistency across rating sessions. These processed ratings are integrated with the broader "MovieLens" data set for generating collaborative filtering analysis and similarity computations with an extensive user base.

5.6 Collaborative Filtering Implementation

User-Based Collaborative Filtering with Pearson Correlation

The system implements user-based collaborative filtering using Pearson correlation to measure user similarity on rating patterns. Pearson correlation is chosen as the similarity matrix because of its ability to handle bias in the ratings. Some users tend to give higher ratings than other users, so to mitigate this kind of skew, Pearson correlation is used. The implementation creates a user movie matrix combining the user rating received from the front end with filtered Movielens data and applying correlation analysis to identify similar user rating behaviors.

The collaborative filtering process employs configurable parameters, including the similarity threshold and the number of recommendations to generate. The system calculates Pearson correlation coefficients between the target user and all movie lens users who have rated common movies, ranking users by similarity scores above a sufficient threshold. This approach makes sure that the statistical significance in similarity measurements is satisfied, and it enables personalized recommendation generation based on authentic user behavior patterns.

Recommendation Generation and Ranking

The collaborative filtering algorithm generates movie recommendations by aggregating preferences from similar users, weighted by their correlation coefficient with the target users. The system predicts ratings for unrated movies based on similar user preferences and ranks the recommendations by predicted rating scores. Each recommendation includes a predicted rating confidence score, similar user statistics for the explainability of collaborative recommendations generated for the user.

5.7 System Integration and User Experience

Progressive Recommendation Workflow

The implemented system provides a seamless user experience through a progressive workflow that transitions users from cold-start to collaborative filtering scenarios. The web-based interface guides users through genre selection, comprehensive questionnaire completion, preference analysis results, content-based recommendations, movie rating collection, and collaborative filtering recommendations. Each phase builds upon previous user interactions, creating increasingly personalized recommendations as more user data becomes available.

The system maintains session continuity and user state throughout the recommendation process, enabling users to return to previous phases or modify their preferences. The interface provides clear feedback on recommendation rationale, displaying genre weights, similarity scores, and collaborative signals to maintain transparency in the recommendation process. This progressive approach ensures that users receive valuable recommendations immediately upon completing the questionnaire while enabling enhanced personalization through continued system interaction.

Data Management and System Reliability

The implementation maintains robust data management practices with comprehensive data validation processes and error handling mechanisms. User questionnaire responses and generated genre weights are stored with detailed metadata including processing timestamps, LLM model parameters, and validation results. The system implements proper API error handling, response validation, and automatic retry mechanisms for failed LLM calls.

The system ensures data consistency through validation checks for genre weight normalization, questionnaire response completeness, and API response integrity. User data is stored securely while maintaining compatibility with research and evaluation requirements. The robust error handling framework ensures system reliability and provides meaningful feedback to users during preference collection and recommendation generation processes.

5.8 Implementation Achievements

The implemented recommendation system successfully addresses the cost order problem through a comprehensive web-based application. It integrates questionnaire-driven preference elicitation, a large language model-enhanced analysis and progressive recommendation generation. The system also demonstrates effective integration of content-based and collaborative filtering approaches, providing immediate recommendations for new users while enabling cold user retention for real-world use cases.

Key implementation achievements include zero short prompt engineering for intelligent preference inference, LLM integration, advanced behavioral guidelines and validation mechanism, Comprehensive collaborative filtering recommendation generation using mathematical precise similarity computation and effective collaborative filtering with Pearson correlation usage. The system successfully bridges the gap between explicit user preferences and implicit behavioral patterns for cold start users, enabling accurate genre weight generation. This establishes a strong foundation both for research applications

and practical deployment scenarios in which high-performance standards and transparency in recommendation explanation is required.

6 Evaluation

6.1 Evaluation Framework Overview

The evaluation framework employs a comprehensive approach using real user interaction data and MovieLens user data as ground truth, ensuring a realistic and meaningful assessment of recommendation system performance. The framework has been enhanced with correctness fixes and addresses the challenges of sparse data, cold-start scenarios, and temporal data leakage through specialized evaluation methodologies. The implementation includes proper dimension alignment for content-based filtering, temporal train/test splits for collaborative filtering, and robust similarity computation using Pearson correlation.

6.2 Ground Truth and Evaluation Methodology

Ground Truth Definition

In recommendation systems evaluation, **ground truth** refers to the verified, actual preferences and behaviors of real users that serve as the reference standard for measuring system performance. Our evaluation framework utilizes real MovieLens user data as ground truth, ensuring that performance assessments reflect genuine user preferences rather than synthetic or simulated data.

Ground Truth in Collaborative Filtering with Temporal Split: For collaborative filtering systems, the ground truth represents the actual future behavior of the users that the system must predict. The ground truth is established through temporal data splitting, where each user's ratings history is chronologically ordered and divided into training and test sets. The system learns from the earliest 80% of the user's ratings and attempts to predict the behavior for the later 20% of ratings. This approach simulates the real-world scenarios where a recommendation system must predict future user preferences based on historical data. This splitting approach prevents data leakage by ensuring the system never sees the future ratings during training, while maintaining min overlap requirement of ($\geq$4 common movies) for statistically valid similarity calculations. In practice, this means that the user with a sufficient rating history will have an ample amount of test data ratings that serve as ground truth for evaluating prediction accuracy.

Ground Truth in Content-Based Filtering: For content-based filtering, ground truth represents the real user preferences that they have expressed through their movie ratings and questionnaire preferences. Unlike collaborative filtering, which predicts the future behavior of user ratings, content-based filtering aims to understand and replicate the user's existing preferences. The ground truth consists of three main components: the user's genre preference, which is extracted from the rated movies using 19 movie genre categories, their rating pattern, which means how the user has rated different types of movies on a scale of 1 to 5, and the alignment between our recommendations and established preferences.

The test data varies depending on what aspect of the system we want to evaluate. For recommendation quality evaluation, the system considers movies rated $\geq$2.5 by the user as relevant items that the system should be able to recommend. For preference-based evaluation, we separate movies into positive examples (highly rated movies $\geq$4.0) and negative examples (poorly rated movies $\leq$2.0) to test whether the system can distinguish between what the user likes and dislikes. For genre alignment evaluation, the system focuses on movies from genres that the user has shown a preference for, based on their rating history.
The evaluation framework uses the Movie Lens data set with users who have a sufficient rating history to ensure we have enough data to make meaningful assessments of the system's performance.

User Profile Creation Strategies

For testing, different user profile creation strategies were used to comprehensively assess the recommendation system's performance across different user modeling approaches. Each strategy is designed to balance computational efficiency with specific parameters for different user segments and evaluation scenarios. In this context, "variance" denotes the instability of user profiles across repeated constructions of different samples of user ratings used to create the user profile. lower variance implies higher stability of the user profile. Below are the six strategies :

1. **Random Strategy**: Original random sampling approach selecting exactly 5 movies from the user's rating history, providing a high-variance baseline for comparison. This strategy serves as the control condition with high expected variance rate and low stability, making it suitable for establishing performance baselines and variance analysis.

2. **Top-Rated Strategy**: Focuses on highly-rated movies ($\geq$4.0 stars) to capture user's preferred content with significantly reduced variance. Uses 10 movies by default with a minimum rating threshold of 4.0, providing high stability and reliability for preference modeling. This strategy is particularly effective for users with clear rating patterns and is recommended for production use.

3. **Adaptive Strategy**: Dynamic sample size approach that automatically adjusts based on user's total rating history availability. Uses a base of 10 movies for users with $\leq$20 ratings, scales linearly up to 25 movies for users with 100+ ratings, and caps at 25 movies maximum.

4. **Hybrid Strategy**: Combines top-rated and random sampling to balance preference capture with diversity. Uses 15 movies total with 70% selected from highest-rated movies and 30% randomly sampled from remaining movies.

5. **Weighted Strategy**: Rating-weighted sampling that gives higher probability to better-rated movies using exponential weighting. Uses 15 movies by default with sampling probability proportional to exp(rating - min_rating). Provides medium stability ($\sim$50% variance rate) and captures rating intensity while maintaining some randomness for diversity exploration.

6. **All Movies Strategy**: Utilizes complete user rating history for comprehensive preference modeling. No parameter limits, uses all available user ratings with optional temporal decay weighting. Provides highest stability but is computationally expensive and may not scale well for users with extensive rating histories. Suitable for research validation and small-scale deployments.

Strategy Selection Criteria The choice of user profile creation strategy can significantly impact the evaluation outcomes and system performance The key considerations include. Data availability, users with limited rating history benefit from strategies that maximize information extraction (adaptive, top-rated) while users with extensive histories can utilize comprehensive approaches (all movies, hybrid). Computational constraints, production systems may prefer efficient strategies (top-rated, adaptive) over computationally intensive approaches. However, for the evaluation of this system, all the approaches are evaluated at the same time for better comparison of results.

Evaluation Strategy

Our evaluation framework implements a flexible and configurable approach to recommendation system assessment through a dedicated configuration system. The framework includes a separate configuration file that enables dynamic control over evaluation parameters, allowing researchers to customize the evaluation process based on specific requirements and computational constraints.

The evaluation configuration system provides two primary dynamic capabilities:

1. **Dynamic User Selection**: The framework allows specification of the exact number of users to be tested through configuration parameters. This enables researchers to control evaluation scope from small-scale testing (10-20 users) to comprehensive evaluation (100+ users) based on available computational resources and time constraints. The system supports both random sampling and targeted user selection strategies.

2. **Preference-Based Evaluation**: Secondary evaluation that separates movies into high-rated (≥ 4.0) and low-rated (≤ 2.0) categories to test preference alignment. Measures the system's ability to recommend movies that match user's rating preferences and avoid movies the user has rated poorly.

3. **Genre Alignment Evaluation**: Secondary evaluation that measures alignment between recommendations and user's actual genre preferences using mathematically correct vector operations. Assesses how well the system captures and utilizes user's genre preferences for content-based filtering.

4. **Temporal Train/Test Split Strategy**: For collaborative filtering, uses temporal ordering of ratings (80% earliest ratings for training, 20% latest for testing) to prevent data leakage and ensure realistic evaluation conditions that mirror real-world deployment scenarios.

5. **Information Retrieval (IR) Metrics Evaluation**: Comprehensive assessment using MRR (Mean Reciprocal Rank), MAP@K (Mean Average Precision), and NDCG@K (Normalized Discounted Cumulative Gain) to evaluate ranking quality and recommendation diversity.

6. **Data Isolation Protocols**: Ensures complete separation between training and test data, with test movies strictly hidden from recommendation generation process to prevent evaluation bias and data leakage.

The evaluation framework implements comprehensive validation methodologies including recommendation quality assessment (Precision@K, Recall@K, Coverage), preference-based evaluation (high vs low rated movies), genre alignment testing, temporal train/test splits for collaborative filtering, and information retrieval metrics (MRR, MAP@K, NDCG@K). All evaluation methods maintain strict data isolation protocols to prevent evaluation bias and ensure reliable performance assessment.

6.3 Implemented Evaluation Metrics

Based on our comprehensive implementation in the evaluation framework, we focus on the following key metrics that have been thoroughly tested and validated:

Primary Metrics: Recommendation Quality

Precision@K and Recall@K For top-K recommendation evaluation, we implemented:

$$\text{Precision@K} = \frac{\text{Number of relevant items in top-K}}{\text{K}} \tag{1}$$

$$\text{Recall@K} = \frac{\text{Number of relevant items in top-K}}{\text{Total number of relevant items}} \tag{2}$$

These metrics evaluate the quality of recommendations at different positions in the ranked list, with K typically set to 5, 10, and 20 in our implementation.

Secondary Metrics: Predictive Accuracy and Preference Alignment

Mean Absolute Error (MAE) For rating prediction tasks:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|predicted_i - actual_i| \tag{3}$$

Root Mean Square Error (RMSE)

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(predicted_i - actual_i)^2} \tag{4}$$

Where MAE provides a straightforward measure of prediction error, while RMSE penalizes larger errors more heavily.

Tertiary Metrics: Information Retrieval (IR) Metrics

Mean Reciprocal Rank (MRR) We implemented MRR to measure the rank of the first relevant item:

$$\text{MRR} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{\text{rank}_i} \tag{5}$$

Where rank_i is the rank of the first relevant item for query i.

Mean Average Precision (MAP@K) For measuring ranking quality across multiple relevant items:

$$\text{MAP@K} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \text{AP}_i@K \tag{6}$$

Where $\text{AP}_i@K$ is the average precision at K for query i.

Normalized Discounted Cumulative Gain (nDCG@K) To account for the position of relevant items in the recommendation list:

$$\text{nDCG@K} = \frac{\text{DCG@K}}{\text{IDCG@K}} \tag{7}$$

Where:

$$\text{DCG@K} = \sum_{i=1}^{K} \frac{rel_i}{\log_2(i+1)} \tag{8}$$

And IDCG@K is the ideal DCG@K for the perfect ranking.

6.4 Content-Based Filtering Evaluation

Evaluation Methodology

The content based filtering evolution involves multiple approaches built upon a mathematical foundation. The implementation ensures proper vector dimension alignment through explicit construction of user preferences vector that exactly matched the movie genre matrix dimension enabling mathematical validity of dot product operations. User preferences vectors are initialized with 0 for all genre columns and then populated with user specific genre weights ensuring complete dimensional consistency.

The evaluation methodology incorporates safe pandas operation that eliminate potential index misalignment issues, replacing unsafe series operation with explicit vector construction and validation.

Content-Based Filtering Results

We evaluated the content-based filtering system across different user profile creation strategies with varying numbers of users. Here are the comprehensive results:

Table 1: Content-Based Filtering Basic Metrics (400 Users)

Strategy	P@5(%)	P@10(%)	P@20(%)	R@5(%)	R@10(%)	R@20(%)
random	6.2	5.6	5.5	0.4	0.7	1.3
top_rated	5.0	5.5	5.5	0.3	0.7	1.6
adaptive	4.5	4.8	5.2	0.3	0.6	1.3
hybrid	4.4	5.1	4.9	0.3	0.5	1.1
weighted	4.0	4.6	5.0	0.2	0.5	1.0
all_movies	4.0	3.8	4.1	0.3	0.5	0.9

Table 2: Content-Based Filtering Basic Metrics (100 Users)

Strategy	P@5(%)	P@10(%)	P@20(%)	R@5(%)	R@10(%)	R@20(%)
random	5.2	5.5	5.3	0.3	0.6	1.2
top_rated	5.2	4.7	5.1	0.3	0.5	1.3
adaptive	3.8	5.1	4.8	0.3	0.8	1.5
hybrid	5.2	4.4	4.2	0.4	0.5	0.9
weighted	5.0	4.8	5.8	0.2	0.3	0.8
all_movies	4.0	3.6	4.4	0.3	0.4	1.0

Table 4: Content-Based Filtering Ranking Quality Metrics Comparison

	400 Users		100 Users			50 Users		
Strategy	MRR	NDCG	MRR	MAP	NDCG	MRR	MAP	NDCG
random	0.149	0.237	0.133	0.003	0.222	0.195	0.003	0.272
top_rated	0.128	0.239	0.109	0.002	0.211	0.135	0.002	0.250
adaptive	0.117	0.216	0.117	0.003	0.218	0.099	0.002	0.172
hybrid	0.120	0.209	0.104	0.003	0.191	0.108	0.002	0.188
weighted	0.110	0.194	0.128	0.002	0.226	0.049	0.001	0.126
all_movies	0.113	0.194	0.112	0.003	0.193	0.176	0.004	0.268

Key observations from the content-based filtering results:

- Performance is relatively consistent across different user counts for most strategies.
- Random strategy shows surprisingly competitive performance, especially with larger user counts.
- Precision@K values are generally higher at lower K values.
- NDCG@20 scores indicate reasonable ranking quality despite lower precision/recall values.
- MRR@20 values suggest the system often places at least one relevant item in top positions.

6.5 Collaborative Filtering Evaluation

Evaluation Methodology

The Collaborative Filtering Evaluation employs user-based collaborative filtering with strong mathematical foundation. The implementation utilizes peers and correlation coefficient, replacing the previous approaches of using cosine similarity with problematic

Table 3: Content-Based Filtering Basic Metrics (50 Users)

Strategy	P@5(%)	P@10(%)	P@20(%)	R@5(%)	R@10(%)	R@20(%)
random	1.6	1.0	0.8	0.1	0.6	1.0
top_rated	1.6	1.2	1.4	0.2	0.3	0.6
adaptive	1.2	0.8	1.0	0.2	0.2	0.7
hybrid	0.8	0.8	0.9	0.1	0.3	0.6
weighted	0.4	0.4	0.7	0.1	0.2	0.6
all_movies	1.6	1.8	1.2	0.2	0.9	1.5

zero-filling strategies. The correlation-based similarity measurement ensures that the user relationships are computed only on shared preferences. This provides more accurate and meaningful similarity scores with minimal overlap requirements to ensure statistical significance.

The evaluation also implements temporal train test splitting that eliminates data leakage by using chronological ordering of user ratings, where the earliest % of ratings serve as training data and the latest 20% constitute the test set. This temporal approach ensures realistic evaluation conditions that mirror real-world recommendation scenarios. Complete data isolation is maintained throughout the process, with test movies being strictly hidden from target user profiles during recommendation generation, preventing any form of information leakage that could inflate performance metrics.

The recommendation generation process employs a weighted average prediction using similar users, incorporating normalization and mean centering techniques that preserve the directional nature of user correlation by avoiding incorrect absolute value scores.

Collaborative Filtering Results

We evaluated the collaborative filtering system with the same strategies but with a focus on both rating prediction accuracy (MAE, RMSE) and recommendation quality metrics. Here are the comprehensive results:

Key observations from the collaborative filtering results:

- Collaborative filtering consistently achieves higher precision and recall compared to content-based filtering.
- MAE and RMSE values are relatively stable across different strategies, typically ranging from 0.5 to 0.7, which is quite good.
- Performance is more sensitive to the number of available users compared to content-based filtering.
- NDCG@20 scores are significantly higher than in content-based filtering, indicating better ranking quality.
- The number of users that can be evaluated decreases significantly with smaller user bases due to minimum overlap requirements.
- Adaptive and weighted strategies generally show good balance between prediction accuracy and recommendation quality.

6.6 Evaluation Results Interpretation

Content-Based Filtering Performance Analysis

The content-based filtering results reveal several important patterns and insights:

Impact of User Base Size Performances seem to generally increase over large user bases. Precision@5 shows the most stability across different user counts.

Strategy Effectiveness The random user profile creation strategy achieves highest precision (6.2% at P@5) with 400 users, Top rated strategy shows consistent performance across metrices, Adaptive and hybrid strategies show balanced performance but lower precision. All movie strategy generally performs worst suggest using complete rating history might introduce noise.

Metric Analysis Precision metrics (P@K) generally decrease it as K increases indicating better performance for shorter recommendation lists. Recall matrix shows very low value suggesting limited coverage of user relevant items. NDCG@20 scores (0.19-0.27) indicate moderate ranking quality despite low precision/recall. MRR@20 values (0.11-0.15) suggest that relevant items, when present, often appear in higher positions.

Collaborative Filtering Performance Analysis

The collaborative filtering results demonstrate distinct characteristics and advantages:

Prediction Accuracy MAE values consistently range from 0.5 to 0.7, indicating reasonable prediction accuracy. RMSE values closely follow MAE patterns, suggesting uniform error distribution. The adaptive strategy achieves the best prediction accuracy with (MAE: 0.528) with 400 users.

Recommendation Quality Collaborative filtering shows higher precision as compared to content-based filtering (15-20% vs 4-6%), and it also has better recall values, especially at higher K values (R@20 reaching 8-13%). NDCG@20 scores consistently above 0.45, indicating good ranking quality. MAP@20 values also show better overall recommendation relevance

Scalability and User Base Impact The number of evaluable users decreases significantly with smaller user bases due to overlap requirements. The performance remains relatively stable, even with a few users suggesting robustness of the algorithm.

Comparative Analysis

When comparing both approaches, several key findings emerge:

Recommendation Quality Collaborative filtering consistently outperforms content-based filtering in precision and recall. Content based filtering shows more stability across different user base sizes Also maintaining more consistent coverage across different strategies. collaborative filtering shows better ranking quality indicated by higher NDCG scores.

Strategy Effectiveness Simple strategies like random and top rated user profile generation perform surprisingly well in content based filtering. Adaptive and weighted studies show better balance in collaborative filtering evaluations and all-movies strategies generally underperforms in both approaches.

Practical Implications Collaborative filtering is preferred when significant data is available and content based filtering is more reliable for cold start scenarios. Hybrid approaches might benefit For a more robust recommendation generation.

This enhanced evaluation framework ensures that our LLM-enhanced recommendation system is thoroughly assessed using mathematically correct implementations and realistic evaluation protocols, providing confidence in both the technical performance and practical applicability of the proposed approach.

7 Conclusion and Future Work

This research has provided comprehensive insights into the effectiveness of different recommendation strategies across varying user base sizes, with particular focus on content-based and collaborative filtering approaches. Our systematic evaluation reveals several key findings and opens up promising directions for future research.

7.1 Key Findings

Recommendation Strategy Performance

The evaluation demonstrated distinct performance characteristics between content-based and collaborative filtering approaches:

- **Collaborative Filtering Superiority**: It consistently achieved higher precision (15-20%) compared to content-based filtering (4-6%), particularly with larger user bases.
- **User Base Sensitivity**: Collaborative filtering showed more sensitivity to user base size, while content-based filtering maintained more stable performance across different user counts.
- **Strategy Effectiveness**: Simple strategies like random and top-rated showed surprisingly competitive performance in content-based filtering, challenging assumptions about strategy complexity.

Scalability and User Base Impact

The research revealed important insights about system scalability:

- **Collaborative Filtering**: Performance significantly improved with larger user bases, but required more computational resources.
- **Content-Based Filtering**: Maintained consistent performance even with smaller user bases, making it suitable for cold-start scenarios.
- **Hybrid Potential**: Results suggest complementary strengths that could be leveraged in a hybrid approach.

Evaluation Metrics Insights

Our comprehensive evaluation metrics provided nuanced understanding:

- **Ranking Quality**: NDCG scores showed better ranking quality in collaborative filtering (0.45-0.54) compared to content-based approaches (0.19-0.27).
- **Precision-Recall Trade-off**: Both approaches showed better precision at lower K values, suggesting optimal recommendation list lengths.

Potential Limitation of LLMs usage

While the research demonstrates the achievements and results Of the LLM-driven user genre generation for the recommendation system, the notable limitations lie in addressing the potential of hallucination where the system generates user genre mat based on inaccurate interpretation or biased interpretation. This issue can arise from model tendencies to synthesize information rather than retrieve it directly. This issue can be resolved in future work by integrating a retrieval-augmented generation (RAG) framework to mitigate the limitation.

7.2 Research Contributions

This work has made several significant contributions to the field, namely, trying to solve the user cold-start problem using LLM's and a novel approach. The robust evaluation framework incorporates multiple evaluation methodologies and metrics, offering practical insights into systems behavior across different user base sizes.

7.3 Future Work

There are several promising directions for future work for research which can build on top of this research paper. The LLM usage can be enhanced to use retrieval-augmented generation for candidate selection for a large dataset. LLMs can be used for sophisticated reranking for the final recommendations as well. More features can be added like people associated with the movies, etc. Exploring deep learning techniques for better feature extraction and representation can also be incorporated.

7.4 Concluding Remarks

This research has demonstrated the complex interplay between recommendation strategies, large language models, and evaluation metrics for cold-start users. While collaborative filtering shows superior performance in ideal conditions, content-based filtering proved more realistic for user base limitations. The findings suggested that the Use of large language models to interpret the code-start a user's profile is a beneficial strategy to tackle the user cold start problem.

The evaluation framework and metrics developed in this research provide a solid foundation for future work and recommendation systems. The insights gained about a strategy's effectiveness and scalability considerations offer practical guidance for system implementation, while the identified future work direction points towards promising areas for continued research and development.

References

Acharya, A., Singh, B. and Onoe, N. (2023). Llm based generation of item-description for recommendation system, *Proceedings of the 17th ACM Conference on Recommender Systems*, ACM, pp. 1204–1207.

Contal, E. and McGoldrick, G. (2024). Ragsys: Item-cold-start recommender as rag system, *arXiv preprint arXiv:2405.17587* . https://doi.org/10.48550/arXiv.2405.17587.

Deutschman, K. (2023). Recommender systems: Machine learning metrics and business metrics. Medium. Available at: `https://neptune.ai/blog/recommender-systems-metrics`.

Evidently AI (2025). 10 metrics to evaluate recommender and ranking systems. Available at: `https://www.evidentlyai.com/ranking-metrics/evaluating-recommender-systems`.

Kieu, H. D., Nguyen, M. D., Nguyen, T. S. and Le, D. D. (2025). Keyword-driven retrieval-augmented large language models for cold-start user recommendations, *Companion Proceedings of the ACM Web Conference 2025*, ACM, pp. 2717–2721.

Krishnabalan, S., Devi, R. and Janarthanan, S. (2022). Evaluating performances of content-based and collaborative filtering in business settings, *IAEME Publication* . Available at: `https://www.oxjournal.org/content-based-collaborative-filtering/`.

Lee, H., Im, J., Jang, S., Cho, H. and Chung, S. (2019). Melu: Meta-learned user preference estimator for cold-start recommendation, *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1073–1082. https://doi.org/10.1145/3292500.3330859.

Liu, D., Chen, K., Chou, Y. and Lee, J. (2018). Online recommendations based on dynamic adjustment of recommendation lists, *Knowledge-Based Systems* **161**: 375–389. https://doi.org/10.1016/j.knosys.2018.07.038.

Liu, F., Cheng, Z., Zhu, L., Liu, C. and Nie, L. (2020). An attribute-aware attentive gcn model for attribute missing in recommendation, *IEEE Transactions on Knowledge and Data Engineering* **34**(9): 4077–4088. https://doi.org/10.1109/TKDE.2020.3040772.

Liu, M., Zhang, S. and Long, C. (2025). Facet-aware multi-head mixture-of-experts model for sequential recommendation, *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining*, pp. 127–135. https://doi.org/10.48550/arXiv.2411.01457.

Luo, Y., Jiang, Y., Jiang, Y., Chen, G., Wang, J., Bian, K., Li, P. and Zhang, Q. (2024). Online item cold-start recommendation with popularity-aware meta-learning, *arXiv preprint arXiv:2411.11225* . https://doi.org/10.48550/arXiv.2411.11225.

Meng, Z., McCreadie, R., Macdonald, C. and Ounis, I. (2020). Exploring data splitting strategies for the evaluation of recommendation models, *Proceedings of the 14th ACM Conference on Recommender Systems*, ACM, pp. 681–686.

Meshram, A. (2023). Evaluation of collaborative filtering model. Medium. Available at: https://medium.com/@azadmeshram/evaluation-of-collaborative-filtering-model-6559b1a49b77.

Sheth, H. (2023). *Recommendation Engine Algorithm: Content-Based Filtering.* https://medium.com/intro-to-artificial-intelligence/recommendation-engine-algorithm-content-based-filtering-92297632e77f.

Thorat, P., Goudar, R. and Barve, S. (2015). Survey on collaborative filtering, content-based filtering and hybrid recommendation system, *International Journal of Computer Applications* **110**(4).

Vartak, M., Thiagarajan, A., Miranda, C., Bratman, J. and Larochelle, H. (2017). A meta-learning perspective on cold-start recommendations for items, *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17).* https://dl.acm.org/doi/10.5555/3295222.3295434.

Wu, L., Zheng, Z., Qiu, Z., Wang, H., Gu, H., Shen, T., Qin, C., Zhu, C., Zhu, H., Liu, Q. and Xiong, H. (2024). A survey on large language models for recommendation, *World Wide Web* **27**(5): 60. https://doi.org/10.48550/arXiv.2305.19860.

Zhang, W., Bei, Y., Yang, L., Zou, H., Zhou, P., Liu, A., Li, Y., Chen, H., Wang, J., Wang, Y. and Huang, F. (2025). Cold-start recommendation towards the era of large language models (llms): A comprehensive survey and roadmap, *arXiv preprint arXiv:2501.01945* . https://doi.org/10.48550/arXiv.2501.01945.

Zhang, X., Kuang, Z., Zhang, Z., Huang, F. and Tan, X. (2023). Cold & warm net: Addressing cold-start users in recommender systems, *International Conference on Database Systems for Advanced Applications*, Springer Nature Switzerland, pp. 532–543. https://doi.org/10.48550/arXiv.2309.15646.

Zhu, F., Wang, Y., Chen, C., Zhou, J., Li, L. and Liu, G. (2021). Cross-domain recommendation: Challenges, progress, and prospects, *arXiv preprint arXiv:2103.01696* . OK https://doi.org/10.48550/arXiv.2103.01696.