Hotel Recommender System

Final Project Report of Recommendation Systems

Teacher: Fatima Rodrigues

2024 / 2025

Cristiane Santos nº1241928

Filip Orlikowski nº 1242389

Olivia Puig nº 1242352

Abstract

This project introduces a hybrid hotel recommendation system that combines Collaborative Filtering and Content-Based Filtering techniques to deliver personalized suggestions tailored to individual user preferences. The report begins with a systematic literature review of hotel recommendation systems following the **PRISMA methodology**, providing a solid foundation for the proposed approach. The system computes user and item similarities using sparse matrices, incorporating user attributes such as service quality, cleanliness, location, and cost-benefit considerations. A dedicated cold-start module ensures relevant recommendations for new users or items with no historical data, while a non-personalized recommendation feature ranks hotels based solely on aggregated user ratings. To mitigate popularity bias and enhance diversity, a penalty mechanism reduces the influence of overrepresented cities. The entire architecture is implemented with **FastAPI**, ensuring modularity, performance, and scalability for real-world deployment scenarios.

Keywords:

Hybrid Recommendation System, Collaborative Filtering, Content-Based Filtering, Cold-Start, Recommendation, Non-Personalized Recommendations, Hotel Recommendation

Table of Contents

List of Figures

Índice de Tables

1 Introduction

As part of the Recommender Systems course, this report aims to develop a hotel recommendation system based on user preferences. The goal is to provide relevant and personalized suggestions that enhance the accommodation search experience, taking into account a wide range of criteria specific to each user.

1.1 Context

We currently live in an increasingly globalized world where the need or desire to be in other places has become part of everyday life. The reasons for traveling may include professional matters, studies and research, specialized medical treatments, curiosity, well-being, quality of life, and many others relevant to modern society.

Thus, the need arises to search for accommodations that align with everyone's goals, expectations, financial capacity, and preferences. For some, it is crucial that the daily rate does not exceed a certain budget; for others, having breakfast with various options and/or proximity to the beach is of utmost importance.

Hence, recommender systems emerge to facilitate this search and provide options with the highest potential to meet the user's needs.

1.2 Problem Description

The vast number of available hotel options—considering attributes such as geographical location, room types, and amenities—combined with the diversity of user profiles, highlights the significant relevance of employing an intelligent system to facilitate the matching process between hotels and user preferences.

Therefore, within the context of the Recommender Systems course, the hospitality domain was selected due to its growing technological demands in the market. The objective is to develop a system capable of recommending hotel options that align with individual user profiles. In addition to achieving compatibility, the system aims to address challenges such as the cold-start problem, particularly in scenarios where no prior user information is available, by employing a non-personalized recommendation strategy.

A hybrid approach is adopted, combining collaborative filtering based on hotel attributes provided by suppliers with content-based analysis derived from user-generated ratings and reviews.

1.3 Report Structure

This dissertation addresses the development of an intelligent hotel recommendation system, motivated by the increasing complexity involved in matching a wide variety of accommodation options with increasingly diverse user profiles. Given the large volume of available hotels—characterized by distinct features such as geographic location, room categories, and amenities—and the heterogeneous nature of user preferences, it becomes evident that traditional search mechanisms are insufficient. This highlights the importance of implementing intelligent systems capable of identifying relevant and compatible options for each individual user.

The main goal of this project is to design and implement a recommender system that assists users in selecting hotels that best match their preferences and requirements. Furthermore, the system aims to address the cold-start problem, where recommendations must be generated even when there is no prior information available about a specific user. To that end, a hybrid recommendation approach will be adopted, combining collaborative filtering techniques—using attributes provided by hotel suppliers—with content-based analysis derived from user-generated ratings and reviews.

Chapter 1 presents the introduction to the research topic, outlining the motivation behind the study, the relevance of recommender systems in the hospitality industry, and the objectives pursued throughout the project.

Chapter 2 provides an in-depth analysis of the state of the art in hotel recommendation systems. It surveys existing methodologies, frameworks, and technologies applied in the development of personalized search and recommendation engines in the tourism and hospitality sectors.

Chapter 3 focuses on the technical specifications of the system, documenting the selection and preprocessing of the dataset, the definition of system functionalities, the architectural decisions, and the algorithms used during implementation.

Chapter 4 presents the experimental evaluation of the system. It discusses the experimental setup, performance metrics, and results obtained, demonstrating the effectiveness of the implemented solution in meeting the initial requirements and revealing potential limitations identified during development.

Finally, the Conclusion summarizes the main findings of the study, critically analyzes the techniques employed, and discusses future work, including trends and opportunities for enhancement in the field of hotel recommender systems..

2 State of Art

To understand the relevance of the topic and the technologies involved, a **systematic literature review** was conducted using the **B-on (Biblioteca do Conhecimento Online)** scientific database aggregates multiple reputable and high-impact scholarly databases and IEEE database.

2.1 PRISMA Methodology

To ensure a structured and transparent approach in constructing the state of the art on hotel recommender systems, the PRISMA methodology (Preferred Reporting Items for Systematic Reviews and Meta-Analyses) was adopted. Widely recognized for its rigor in systematic reviews, PRISMA was chosen to guide the processes of searching, selecting, and analyzing the most relevant studies in the domain [1].

Following its standard workflow—identification, screening, eligibility, and inclusion—relevant works were filtered based on research terms such as “Recommender Systems” and “Hotels”. The application of the PRISMA methodology ensured greater consistency and reliability in the review, providing a solid foundation for understanding the main approaches, emerging trends, and technological gaps in the development of hotel-focused recommender systems.

2.2 Selection Method

The initial step consisted of querying the B-on database using the following search expression:

SU (recommendation system* OR recommender system*) AND SU (hotel OR hotels)

This query was designed to capture studies that directly address recommender systems applied within the hospitality domain. Subsequently, filters were applied to refine the search by language (English), publication type (peer-reviewed journal articles and conference papers), and publication period (last 5 years), ensuring the selection of up-to-date and relevant contributions in this fast-evolving technological field.

After the identification phase, the screening stage involved a review of article titles and abstracts by three independent reviewers. Only studies aligned with the core research themes were included:

- Techniques used for hotel recommendation
- Data processing for hotel recommendation
- Challenges and trends in hotel recommender systems

2.2.1 **Elegibilidade**

As mentioned by [1], the process involved several sequential steps: identification, screening, eligibility assessment, and inclusion of articles for analysis.

So, articles were included if they met the following criteria:

- Articles published between 2020 and 2025.
- Peer-reviewed studies, preferably in scientific journals.
- Publications in English.
- Works that explicitly cover the application of Recommendation Systems for Hotels.
- Works that describe practical applications or methodological proposals related to the theme.

In the same direction, the exclusion criteria follow:

- Articles unrelated to Recommendation Systems for Hotels.
- Studies that do not address decision support in a clear or structured way.
- Too specific work.
- Too generic work.
- Publications without access to the full text.

Following the screening phase, the full-text reading of all eligible articles was conducted in order to assess their adherence to the predefined inclusion criteria and to extract relevant insights that could address the research questions. This step was crucial to ensure the academic and technical relevance of the selected publications and to support the construction of a consistent theoretical framework.

2.2.2 **Quantitative Overview of the Selection Process**

The selection process of the systematic literature review followed the four phases proposed by the PRISMA methodology: identification, screening, eligibility, and inclusion.

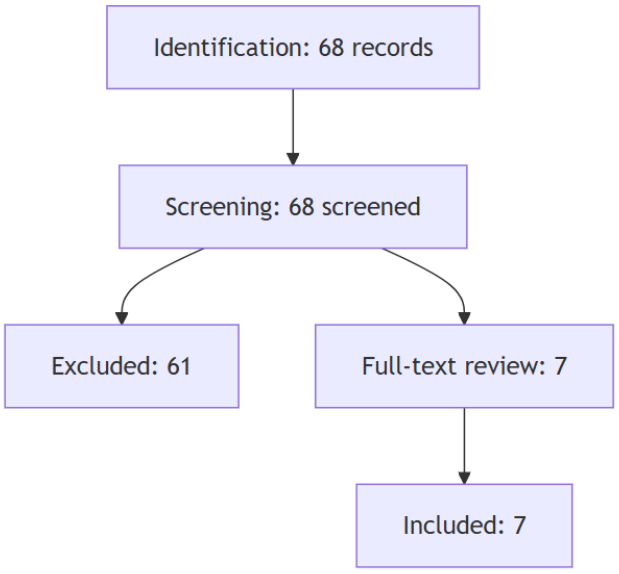

Figura 1 - PRISMA fluxogram

This rigorous process, supported by the PRISMA framework, ensured the consistency, transparency, and reliability of the literature review, thus enabling a comprehensive and up-to-date overview of the state of the art in hotel recommendation technologies.

2.3 Research Questions

To guide the systematic review and ensure that the selected studies effectively contribute to the objectives of this work, the following research questions (RQs) were defined:

- RQ1: Which recommendation techniques and algorithms are most commonly used in systems applied to the hospitality domain?
- RQ2: What data sources and attributes are considered relevant for personalizing hotel recommendations?
- RQ3: What challenges and limitations are identified in recommender systems for the hotel sector according to the literature?
- RQ4: What technological trends and emerging approaches are being explored to improve the effectiveness of these systems?

These questions aim to map the state of the art regarding methods, data, and innovations, while also identifying gaps and opportunities for future research in the context of recommender systems applied to hospitality.

2.4 Results

The analysis of the seven studies reveals a wide diversity of approaches for hotel recommender systems, addressing the four proposed research questions.

2.4.1 Most Common Techniques and Algorithms

Collaborative filtering (CF) remains the predominant technique. This is evident in models such as the Sentiment-enhanced Neural Collaborative Filtering proposed by Dursun and Ozcan (2023), which incorporates explicit user preferences and sentiment analysis [2], as well as in the PLTS-based model developed by Wang et al. (2023), which combines modified cosine similarity with semantic weighting of attributes [3]. Krishna et al. (2023) introduced a multi-criteria system that integrates item-item collaborative filtering with multiple regression and backward elimination [4]. Other methods include content-based filtering, as explored by Shah and Jacob (2022), which leverages NLP techniques such as TF-IDF and word2vec [5], and hybrid or session-based approaches, as in Adamczak et al. (2023), who focus on sequential modeling and temporal user interactions [6]. Deep learning-based approaches, particularly multitask learning (MTL) as exemplified by the FMTL model proposed by Li et al. (2023), are also emerging as key strategies for personalization and multi-objective optimization [7].

2.4.2 Relevant Data Sources and Attributes

The studies rely primarily on user-generated hotel reviews. The types of attributes extracted vary widely, from explicit hotel features (e.g., location, price, cleanliness) to implicit factors such as sentiment polarity and behavioral patterns [2] [4]. Krishna et al. (2023) utilize structured TripAdvisor data with six explicit criteria and contextual metadata such as trip type and hotel class [4]. Li et al. (2023) focus on behavioral interaction logs (e.g., clicks and conversions), addressing the challenges of sparse and long-tail data distributions typical in tourism domains [7]. Adamczak et al. (2023) highlight in-session data, such as clicks, filter usage, temporal and geolocation data, and price range—key for contextualized recommendations [6].

2.4.3 Challenges and Limitations

Commonly cited challenges include data sparsity and long-tail distributions, especially in high-cost, low-frequency booking domains, as noted by Li et al. (2023) [7] and Adamczak et al. (2023) [6]. Semantic ambiguity and the presence of biased or malicious reviews are highlighted by Wang et al. (2023) [3]. Computational complexity is a recurring issue, especially in multi-criteria and fuzzy logic-based systems, as reported

by Krishna et al. (2023) [4] and Solano-Barliza et al. (2024) [8]. Additional concerns include cold-start problems, the need for real-time adaptation, and difficulties in assigning accurate weights to decision criteria.

2.4.4 **Emerging Technological Trends**

Several studies underscore the increasing adoption of deep learning and context-aware systems. Li et al. (2023) propose the FMTL model, which employs multitask learning and temperature-based gating to jointly optimize CTR and CVR under sparse data conditions [7]. Adamczak et al. (2023) demonstrate advances in session-based recommenders capable of real-time sequence modeling [6]. The use of fuzzy logic and asymmetric aggregation, as introduced by Solano-Barliza et al. (2024), also represents a novel direction for handling nuanced preferences [8]. Sentiment analysis, probabilistic linguistic term sets (PLTS), and hybrid recommender architectures are gaining traction as effective tools for enhancing recommendation quality in uncertain or noisy textual environments [3].

2.5 Discussion and Conclusion

The studies align in recognizing the importance of personalization and the predominant use of collaborative filtering, often enhanced by modern techniques. Key differences lie in the technical focus and the types of data utilized.

Erro! A origem da referência não foi encontrada. and **Erro! A origem da referência não foi encontrada.** provide a structured comparison of the selected studies, highlighting their technical approaches, data sources, key challenges, and emerging trends. **Erro! A origem da referência não foi encontrada.** summarizes the main recommendation techniques adopted, ranging from collaborative filtering and content-based methods to hybrid and deep learning models. It also outlines the types of data utilized—such as user reviews, interaction logs, contextual attributes—and identifies the most prominent challenges faced by each approach, including data sparsity, computational complexity, and semantic ambiguity.

Erro! A origem da referência não foi encontrada. focuses on technological trends and innovative approaches observed across the studies. These include the application of multitask deep learning models, integration of sentiment analysis and fuzzy logic, session-based recommendation strategies, and the development of context-aware and real-time adaptive systems. Together, these tables reveal a clear trajectory in hotel

recommender systems research: moving from traditional recommendation paradigms toward more dynamic, context-sensitive, and user-centric architectures.

Table 1 provides a comparative overview of the methods, data sources, and challenges.

Table 1 -Comparative Summary of Reviewed Articles

Art.	Main Technique	Data Sources / Attributes	Reported Challenges
[2]	Collaborative Filtering + Sentiment Analysis	Reviews with explicit user preferences	Sentiment noise, subjectivity
[3]	CF with PLTS	User reviews, attributes like price and service	Semantic ambiguity, fake reviews
[4]	Item-item CF + Regression	TripAdvisor: 6 criteria + hotel class & trip type	Multicollinearity, model complexity
[5]	Content-Based Filtering + NLP	Customer reviews, TF-IDF, word2vec	Lack of contextual adaptation
[6]	Session-Based Recommender	Clicks, filters, geo-temporal context	Short session lifespans, real-time constraints
[7]	Multitask Learning (FMTL)	Behavioral logs (CTR, CVR)	Sparse data, multi-objective trade-offs
[8]	Graded Logic + Asymmetric Aggregation	Booking.com, Google Maps, location, safety	Structured data requirements, high computational cost

Table 2 - Emerging Trends in Hotel Recommendation Research

Trend	Applied Techniques	Supporting Studies
Deep Learning & Multitask Learning	FMTL with multi-level gating	[7]
Sentiment-Semantic CF	PLTS, sentiment scoring	[2] [3]
Context-aware & Real-Time Systems	Session modeling, hybrid CF-context	[6] [4]
Hybrid Recommenders	Combination of CF + content + context	[6] [8]
Fuzzy Logic & Criteria Aggregation	Asymmetric logic-based models	[8]

The study highlights the importance of combining explicit data, user ratings, and user reviews to improve recommendation accuracy and address data sparsity. This approach demonstrates the potential of integrating traditional collaborative filtering techniques with sentiment analysis methods to deliver more personalized and precise recommendations. It is particularly valuable in the tourism sector, where users often rely on reviews and ratings to make informed decisions about accommodations.

The articles reflect a continuous evolution in hotel recommender systems, increasingly focused on real-time personalized experiences and the intensive use of heterogeneous data. However, none of the studies delve deeply into user data privacy or explore robust strategies for group or federated recommendations—gaps that represent promising directions for future research and improvements in hotel recommendation systems.

3 Technical Specifications Description

3.1 Database

The hotel recommendation system uses a database composed of CSV, JSON files, and sparse matrices (.npz) to store and process information about users, hotels, and interactions. These matrices are fundamental for implementing Collaborative Filtering, Content-Based Filtering, and Hybrid Recommendation techniques.

3.1.1 **Database Structure**

The database contains the following files described on Table 3, Table 4 and Table 5 Table 6:

Table 3 - CSV Files

File	**Description**
REVIEWS_DF.csv	Contains user reviews of hotels, including ratings and comments.
USER_DF.csv	Stores user information, such as location, number of reviews, and helpful votes received.
hotel_df.csv	Contains hotel details, including name, location, category, and attributes like cleanliness and cost-benefit.

Table 4 - .npz Files (Sparse Matrices)

File	**Description**
hotel_features.npz	Matrix of hotel attributes, used for content-based filtering.
hotel_similarity_matrix.npz	Matrix of hotel similarity, calculated based on shared attributes.

File	Description
user_hotel_matrix.npz	Interaction matrix between users and hotels, essential for collaborative filtering.
user_similarity_collab.npz	Matrix of user similarity, used to find similar profiles.

Table 5 - JSON Files

File	Description
users.json	Contains structured user information, including identifiers and preferences.
hotel_id_to_idx.json	Maps hotel identifiers to indexes in the similarity matrix.
user_id_to_idx.json	Maps user identifiers to indexes in the interaction matrix.

3.1.2 Example Matrices

Table 6 stores user-hotel interactions, indicating ratings and visits. The values represent ratings given by users to hotels

Table 6 - User-Hotel Interaction Matrix (user_hotel_matrix.npz)

User\Hotel	Hotel 1	Hotel 2	Hotel 3	Hotel 4
User A	5.0	0.0	3.0	0.0
User B	0.0	4.0	0.0	2.0
User C	1.0	3.5	0.0	5.0

.

Table 7 defines attributes such as service quality, cleanliness, cost-benefit, and location. The values indicate average ratings of the attributes.

Table 7 - Hotel Features Matrix (hotel_features.npz)

Hotel	Service Quality	Cleanliness	Average Price	Category
Hotel A	4.5	4.8	120€	5★
Hotel B	3.0	3.5	80€	3★
Hotel C	4.7	4.6	150€	4★

Table 8 stores cosine similarity between user profiles. The vValues close to 1 indicate high similarity.

Table 8 - User Similarity Matrix (user_similarity_collab.npz)

User	User A	User B	User C
User A	1.0	0.85	0.65
User B	0.85	1.0	0.70
User C	0.65	0.70	1.0

Table 9 records the degree of similarity between hotels based on shared attributes. Higher values indicate greater similarity between hotels

Table 9 - Hotel Similarity Matrix (hotel_similarity_matrix.npz)

Hotel	**Hotel A**	**Hotel B**	**Hotel C**
Hotel A	1.0	0.75	0.80
Hotel B	0.75	1.0	0.60
Hotel C	0.80	0.60	1.0

3.2 Recommender System Architecture

The implemented recommendation system combines two main approaches: Collaborative Filtering and Content-Based Filtering, using the cosine similarity matrix as the primary metric to identify similarities between users and items. Additionally, strategies are adopted to penalize over-represented cities, avoid redundant recommendations, and address the Cold Start problem, both for new users and new hotels, ensuring the quality of recommendations from the very first interaction.

3.2.1 Collaborative Filtering

The Collaborative Filtering method is based on user similarity to recommend hotels. Similarity is calculated using the cosine similarity matrix applied between user feature vectors, which include collected interactive data (such as location, number of visited cities, reviews, and helpful votes). This approach enables the identification of similar users and provides personalized recommendations based on the collective behavior of the community. The choice of this technique is due to its proven effectiveness in capturing implicit preferences in large databases.

3.2.2 **Content-Based Filtering**

The Content-Based approach, in turn, focuses on specific characteristics of hotels, using explicit user preferences, such as service, cleanliness, location, number of rooms, hotel class, and preferred region. Categorical variables, such as region, are encoded using OneHotEncoder to better represent them in the vector space. The system performs a strict filtering to select hotels that meet the defined class and region criteria and then calculates the similarity between the user profile and the filtered hotels, ensuring that recommendations are aligned with the stated preferences.

3.2.3 **Cosine Similarity Matrix**

A common problem found in hotel recommendation systems is data sparsity, resulting in sparse matrices where most values are missing. This occurs due to the large number of items (hotels and users) relative to the users' capacity to evaluate them. In other words, each user rates only a few hotels rather than all of them, making personalized recommendation difficult due to the lack of information to infer preferences. This requires specific techniques to handle missing data and improve recommendation quality. In this study, cosine similarity was chosen as the metric, which measures the angle between vectors in multidimensional space. This metric is well-suited for sparse and scaled data, common in recommendation systems, as it evaluates the direction of vectors regardless of magnitude, efficiently capturing behavior patterns and preferences between users and items.

3.2.4 **Additional Strategies (Post-processing)**

To improve diversity and reduce popularity bias, additional adjustments are applied. Highly popular cities (such as New York) have their scores reduced by a penalization factor (e.g., multiplication by 0.6), avoiding excessive recommendations in those locations. Furthermore, hotels already interacted with (visited or rated) by the user are removed from the final list by setting their score to zero. These filters ensure that recommendations are new and non-redundant for the user.

3.2.5 Strategies for Cold Start

To solve the Cold Start problem, the system implements two complementary approaches:

- **Cold Start User-Based**: Interactively asks for basic data from the new user and creates a feature vector combining numerical variables and location. It uses cosine similarity to compare the new user with existing users, making weighted recommendations and dynamically updating the user-item matrices for inclusion.
- **Cold Start Hotel-Based**: Requests detailed information about specific hotel preferences and applies a strict filter to limit recommendations to hotels that match explicit preferences, such as class and region. Similarity is then calculated between the user profile and the filtered set of hotels.

However, like in the user-based approach, flexibility is built into the system: even if some preference fields are left blank, it adapts by relaxing filters and adjusting the similarity calculation accordingly. This ensures that users still receive relevant recommendations based on whatever subset of attributes they have specified, avoiding a complete lack of suggestions due to missing data.

3.3 Recommender System Implementation

3.3.1 General Recommendation Flow

The overall system flow is as follows:

- **Authentication**: User verification (login or token). If it fails, returns an access error.
- **Data Loading**: Initialization of sparse matrices, feature vectors, and ID mappings (users, hotels, similarities).
- **Cold-Start**: If the user is new (not registered), the cold-start module is used to collect initial preferences and generate profile-based recommendations.
- **Hybrid Calculation**: For existing users, collaborative filtering and content-based filtering are applied to compute scores (as described below).
- **Post-Processing**: City penalty is applied, and already viewed hotels are removed from the list.
- **Response**: Hotels are ordered by final scores, and the top-K recommended hotels are returned.

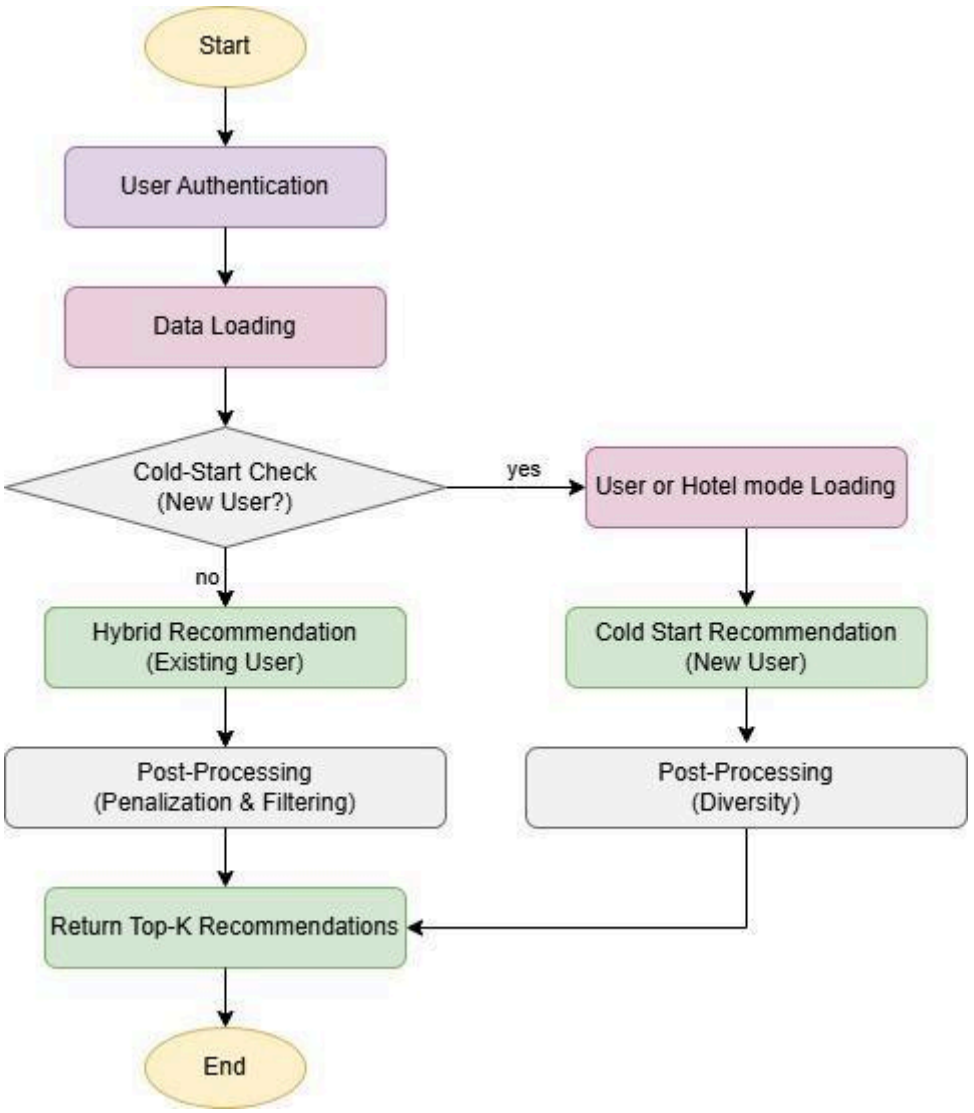

Figura 2 - Recommendation Flow

3.3.2 **Frontend: main.py – FastAPI API**

This module initializes the web application using the FastAPI framework, a modern and high-performance framework for building Python APIs. In main.py, REST endpoints are defined for user authentication (login in Figura 4), new user registration depending on the mode(Figura 5), and recommendation generation (Figura 7). Additionally, it loads persistent data required for the system to function (such as sparse matrices of user-hotel interactions, ID mapping dictionaries, and hotel metadata).

3.3.3 Strategies

- **API Definition:** Creation of endpoints to handle requests (@app.get(), @app.post()).
- **Integration with the Recommendation System:** Calls functions from recommender_v2.py and recommender_cold_start_def.py.
- **Server Execution:** Uses **uvicorn** to launch the FastAPI server.

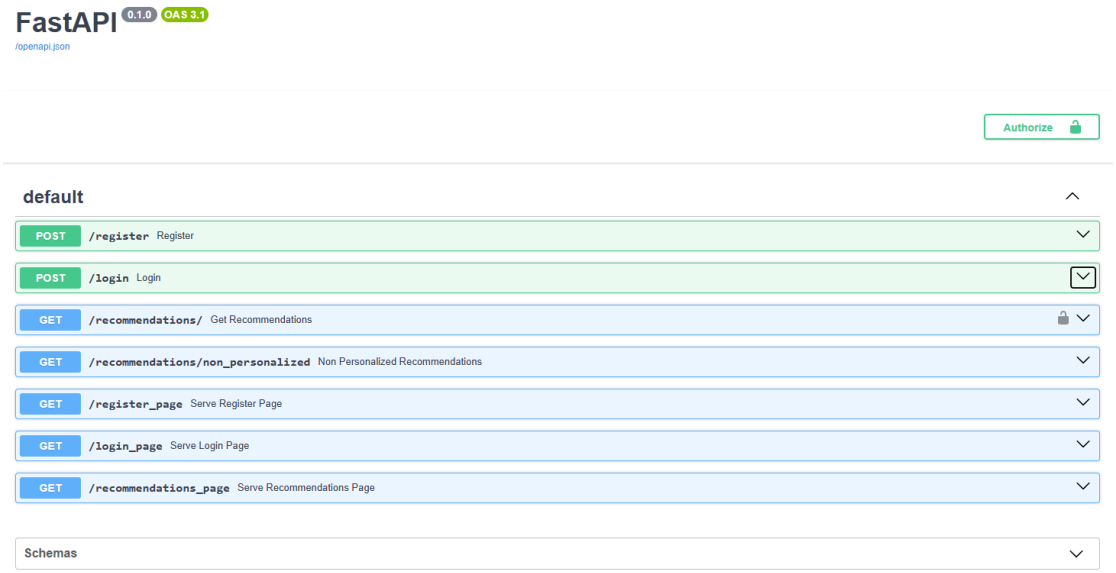

Figura 3 - REST endpoints with FastAPI

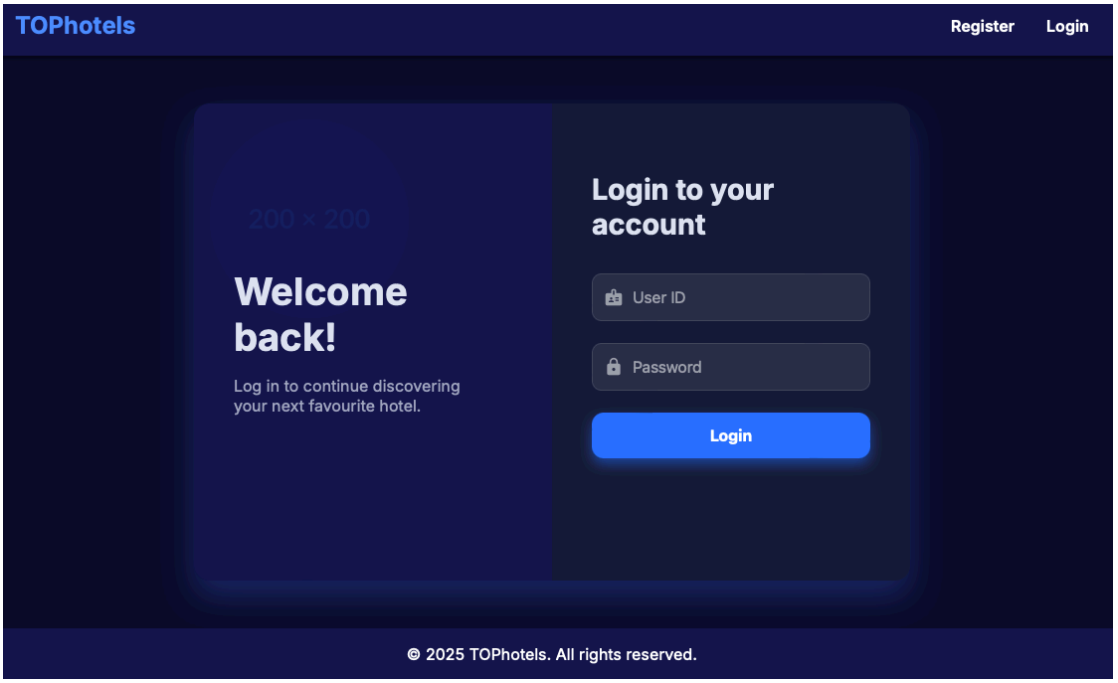

Figura 4 - Authentication

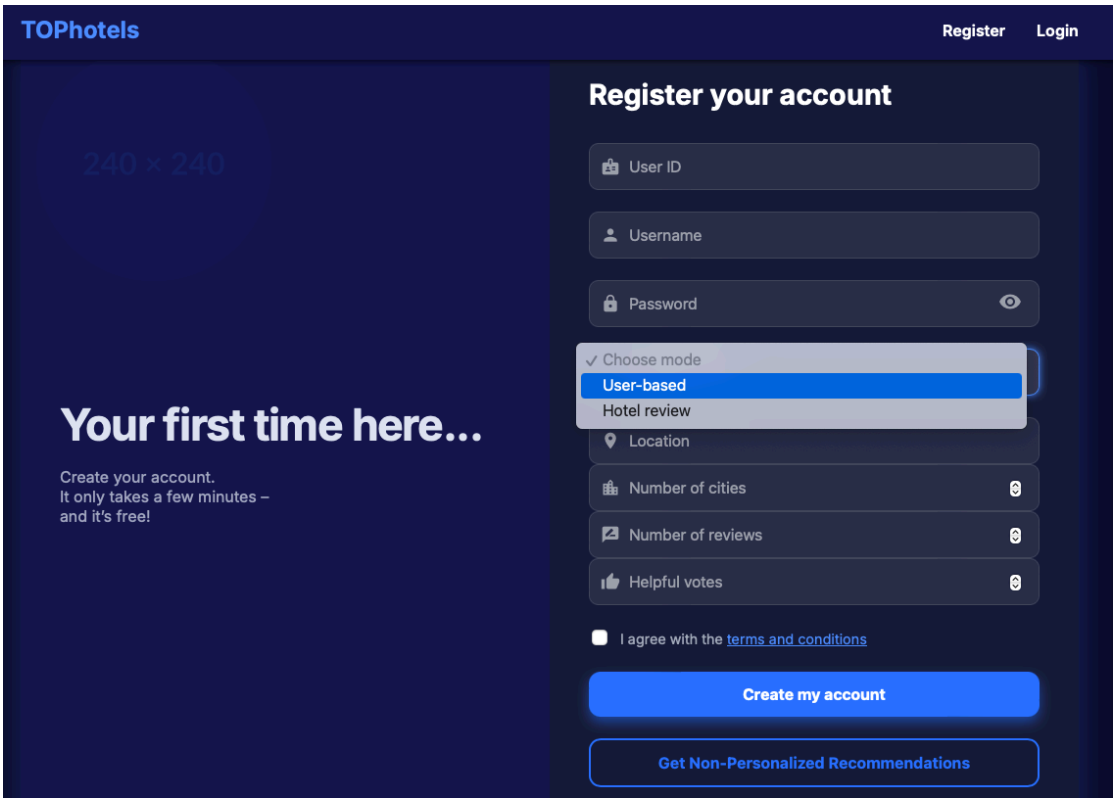

Figura 5 - New user - User or hotel mode

Figura 7 – Recommendations

In the *recommender_v3* system, the recommendation output not only lists suggested hotels but also displays key hotel attributes, the percentage match with user preferences, and the normalized scores used for ranking. This allows users to understand how each recommendation was generated based on their input.

3.3.4 **Core Recommendation Logic: recommender_v2.py**

This module implements the hybrid recommendation algorithm, combining Collaborative Filtering and Content-Based Filtering. The primary function, hybrid_recommend(), generates recommendations for an existing user.

Function hybrid_recommend()

- Combines Collaborative Filtering and Content-Based Filtering to generate hybrid recommendations.
- Computes user similarity and hotel similarity.

Uses a parameter alpha to weigh the two methods:

$$score\_hybrid = \alpha * score\_collab + (1 - \alpha) * score\_content$$

- Removes hotels the user has already interacted with.
- Applies city penalty and returns the best hotels.

Function apply_city_penalty()

It prevents overrepresented cities from dominating recommendations:

- Loads hotel metadata (hotel_df.csv).
- Applies a penalty factor for cities such as New York (0.6) and Houston (0.85).
- Adjusts hotel scores and reorders recommendations.

Function get_non_personalized_recommendations()

This function generates hotel recommendations without considering the user's interaction history, solely based on the average review scores of the hotels. Here's how it works step by step

- Generates non-personalized recommendations, based only on hotel quality.
- Loads the hotel feature matrix (hotel_features.npz).
- Computes a combined score based on average ratings and hotel category.
- If diversify=True, boosts the scores of hotels similar to the top-rated ones.

3.3.5 Recommendation for New Users: recommender_cold_start_def.py

This module handles new users with no interaction history. It includes functions such as cold_start_recommendation() and cold_start_recommendation_combined(), which collect initial user preferences (e.g., desired city, hotel type, personal profile).
Based on these responses or basic user attributes, an initial profile vector is created to recommend hotels with similar attributes (using cosine similarity of feature vectors).
A combined strategy also exists to integrate multiple criteria.

Function cold_start_recommendation()

- Generates recommendations for users with no previous interactions.
- Requests user information (desired location, number of reviews, and helpful votes) or hotel-related attributes (e.g., cleanliness, category, price).
- Creates a user feature vector.
- Computes similarity between the user vector and existing users.
- Generates recommendations based on the most similar profiles.

Function cold_start_recommendation_combined()

Allows recommendations based on user profile or hotel preferences.
If there is insufficient data to compute user similarity, hotel attribute-based **recommendations are prioritized**.

User Mode

- Requests user location, number of cities visited, number of reviews, and helpful votes received.
- Creates a feature vector based on this information.
- Computes cosine similarity between this vector and existing users.
- Generates hotel recommendations based on highly-rated hotels from similar users.

Hotel Mode

- Retrieves hotel attributes, including service quality, cleanliness, value, category, and location.
- Uses Content-Based Filtering to compute hotel similarity by comparing relevant attributes.
- Selects the most similar hotels to the top-rated ones and returns them as suggestions.

Function get_non_personalized_recommendations()

- Recommends hotels solely based on quality.
- Uses the hotel feature matrix (service quality, price, category).
- Computes scores based on average ratings and number of interactions.
- Optional diversification: Boosts hotels similar to top-rated ones.

3.3.6 Core Recommendation Logic: recommender_v3.py

The latest version of the hotel recommender system introduces a more flexible and robust hybrid recommendation engine. It adapts to both users with interaction history and those with limited or no data, using smart fallbacks, clustering, and better normalization.

Function hybrid_recommend() – Enhanced Logic

This function generates personalized hotel recommendations using a hybrid approach that combines collaborative filtering (CF) and content-based filtering (CBF). Key improvements include:

Cold-Start Handling Within the Function: If a user is not recognized, it automatically delegates to cold_start_recommendation_combined(), ensuring smooth integration without requiring external routing logic.

No-History Users: Cluster-Based Fallback: If the user exists but has not rated any hotel, the function uses a pre-trained KMeans clustering model(user_cluster_model.pkl) to assign the user to a cluster based on latent factors (from a matrix factorization model). The system then recommends popular hotels within that cluster, using a precomputed list in cluster_top_hotels.json.

This method allows the system to give meaningful recommendations to users without history, without defaulting to generic or non-personalized suggestions.

Clustering Model: Static KMeans Integration

To support this cluster-based fallback mechanism, a separate script was created to train and export a static KMeans model on user latent factor representations (U_factors.npy).

This model groups users into clusters based on their inferred preferences. Once trained, it allows new users to be immediately classified and receive tailored recommendations based on group behavior.

To determine the optimal number of clusters (k), an elbow method analysis was performed. The inertia score (within-cluster sum of squares) was plotted against different values of k. Below is the elbow graph used during this analysis:

As shown, the "elbow" in the graph appears around k = 5 or 6, which reflects a trade-off between granularity and overfitting. Based on this, an appropriate number of clusters was selected and fixed in the KMeans model used in production.

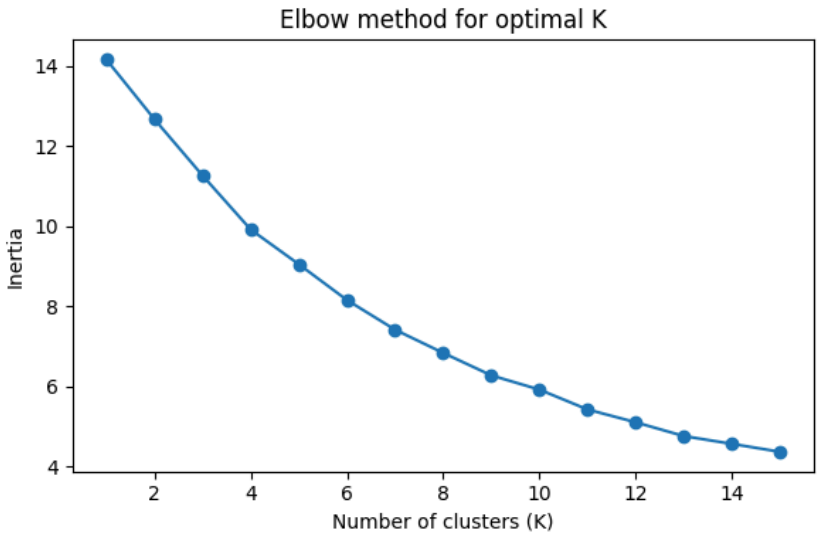

Figure 8: Graph of elbow method

4 Results

4.1 Hybrid Personalized Recommendation – Happy Path

Once a user registers and logs in, the system is able to retrieve their past interactions and generate a fully personalized set of hotel recommendations.

Using the hybrid_recommend() function, a list of 10 hotels was returned with normalized scores on a 1–10 scale, combining collaborative and content-based filtering. The results show a clear alignment with the user's preference for mid-range hotels with high cleanliness and service ratings, particularly in coastal cities.

For example a recurrent user, EmilyScotland, with 11 cities visited, and 17 reviews on the profile, has this recommendation:

Hotel Name	Location	Category	Normalized Score (1–10)
DoubleTree by Hilton Hotel Boston - Downtown	Boston	3.5★	**10.00**
The Kensington Park Hotel	San Francisco	3.0★	8.40
Chancellor Hotel on Union Square	San Francisco	3.0★	8.27
The Westin Book Cadillac Detroit	Detroit	4.0★	8.02
Hampton Inn & Suites Denver Downtown	Denver	2.5★	7.49
DoubleTree Club by Hilton Hotel Boston Bayside	Boston	3.0★	7.46
Courtyard by Marriott San Diego Downtown	San Diego	3.0★	7.37
Hampton Inn Philadelphia Convention Center	Philadelphia	3.0★	6.58
Hilton Philadelphia City Avenue	Philadelphia	3.5★	6.26
W New York - Times Square	New York City	4.5★	3.88

Table 10: Table with recommendations of a known user

4.1.1 **Registration**

When a new user signs up, the system requests basic preference information (e.g., city, preferred hotel class, value expectations). Even with incomplete data, the recommender allows **flexibility** by generating suggestions based on whichever fields were filled.

For example, if the user only selects a target city and price range, the system filters the hotel database accordingly and computes similarities to generate an initial set of recommendations.

The module cold_start_recommendation_combined supports multiple modes and input scenarios. The test file test_cold_start.py demonstrates these cases clearly:

Test #	Mode	Inputs Provided	Description	Sample Output Example
1	User	Location + Number of Helpful Votes + Number of Reviews	Typical user info including location and activity metrics	Hotels with IDs and scores based on user similarity
2	User	Location + Number of Cities Visited	Basic location and travel history	Hotels recommended based on clusters with location
3	User	Unknown Location + Number of Reviews	Handles cases where location is not in the database	General recommendations ignoring location specifics
4	Hotel	Hotel attributes (service, cleanliness, value, class, etc.)	Hotel-based mode where preferences about hotel features given	Hotels similar to the input hotel profile

Table 11: Table with different combinations of new users

Output given*:*

▸ ***Test 1: With location + helpful + reviews***

[(93520, 918.6730886501223), (93517, 803.6600142290475), (93562, 795.4653121507018), (2079052, 776.8565198640416), (266157, 738.3737161231597), (93437, 709.3154639723131), (113317, 692.7894203600897), (122005, 688.3140006924451), (109413, 639.3773786080461), (214197, 630.2605452084583)]

▸ ***Test 2: With location + cities***

[(93520, 1211.3443587222762), (2079052, 1080.3083297046894), (93517, 1046.4039734830435), (266157, 995.8051922039838), (93562, 931.4683532618556), (109413, 906.7581004968283), (83040, 891.4591021443115), (122005, 883.1575055712636), (214197, 872.5590710008518), (665258, 839.4384145870229)]

▸ ***Test 3: Location unknown***

[(93520, 1026.4956183329205), (2079052, 935.0977170253593), (93517, 847.6998418566674), (122005, 846.1663403674972), (93562, 813.0346437030429), (266157, 742.7249189705408), (214197, 708.1671598144668), (109413, 685.6608775036038), (93437, 685.2315404763438), (99288, 650.5214456234157)]

▸ ***Test 4: Mode hotel***

[(2627745, 0.8715462990498509), (2015227, 0.8714992255241759), (286542, 0.8708131547246588), (277997, 0.8706151889539766), (1147292, 0.8674462100441844), (112178, 0.8639541677822127)]

4.1.2 **Login**

On the login page, a set of synthetic users was created along with their corresponding passwords. This synthetic user database serves as a controlled environment for simulation and testing purposes. By having predefined users, the system can reliably simulate different user interactions, ensuring consistent and repeatable results during the development and evaluation of the recommendation system.

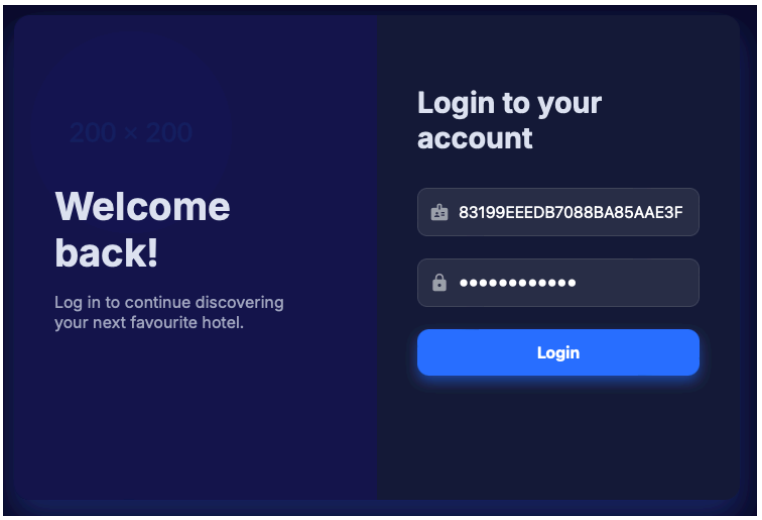

Figure 9.1 : Login page of one user

4.1.3 **Recommendations**

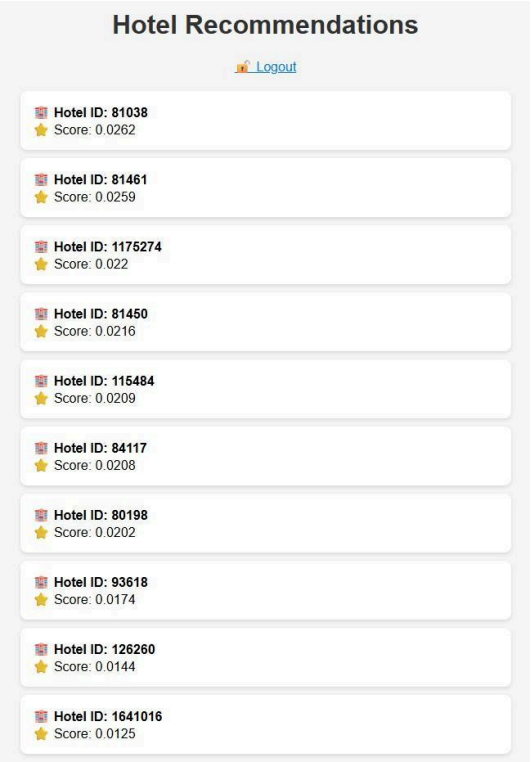

Figure 9.2: Recomendation of the login page user

4.2 Non Personalized Recommendation

In the absence of user data (i.e., before login), the system can fall back on a non-personalized recommendation module, which ranks hotels purely by their global quality (average rating, number of reviews, etc.).

This provides a solid entry point for anonymous users and maintains system usability even without registration.

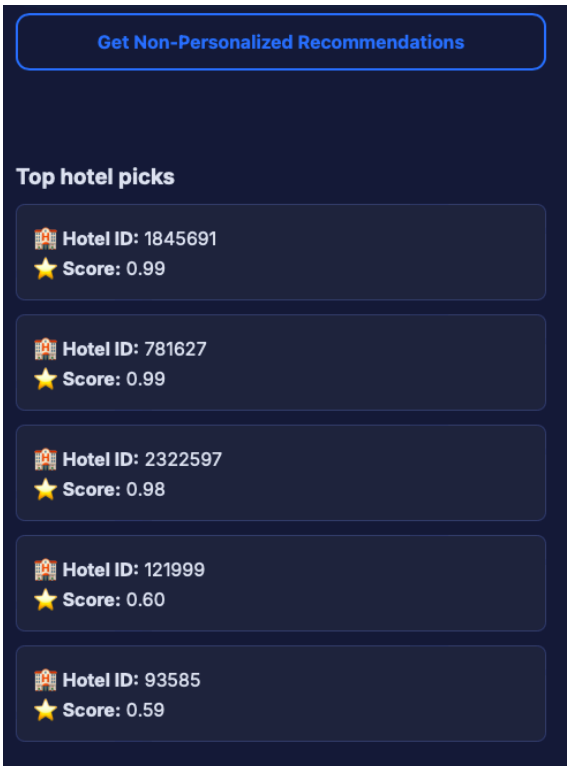

Figure 10: Output of a Non-Personalized Recommendations

4.3 Including Reviews

Since we do not have access to actual written user reviews in text form, our system relies on existing users' review scores and ratings to build a broader understanding of user profiles. To achieve this, we developed a clustering model that groups users with similar behaviors and preferences. This allows us to leverage information from users within the same cluster to personalize recommendations more effectively, even without explicit textual feedback from the new user.

This clustering approach gives us a wider vision of the variety of user profiles and helps the system suggest relevant hotels based on similar users' preferences.

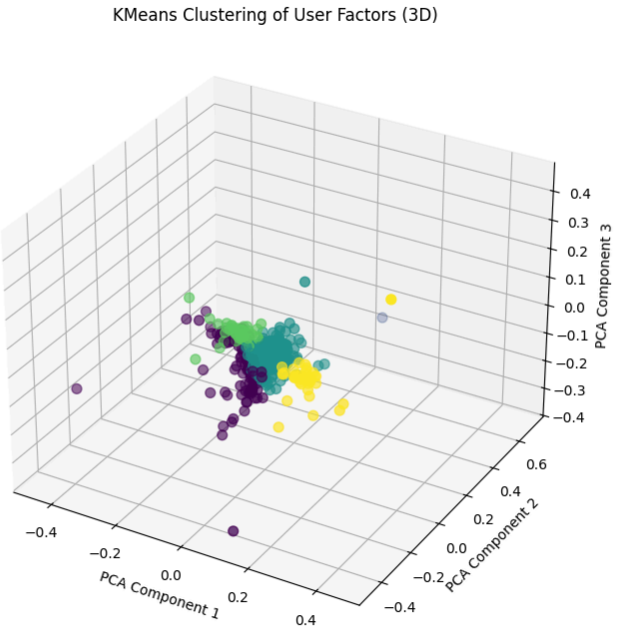

Figure 11: K-Means clustering

For future implementations, when textual reviews become available, we plan to integrate a Natural Language Processing (NLP)-based sentiment analysis system. This would analyze the content of reviews to extract more nuanced insights about hotels, such as specific strengths and weaknesses, enabling the recommender to combine quantitative ratings with qualitative user feedback for richer and more trustworthy recommendations.

5 Conclusions

In conclusion, this project successfully developed and implemented a hotel recommendation system as part of the Recommendation Systems course within the Computer Engineering master's program. Addressing the increasing complexity of matching diverse accommodation options with varied user profiles, the system employs a hybrid approach that integrates Collaborative Filtering and Content-Based Filtering to generate personalized hotel recommendations. Through cosine similarity and sparse matrix operations, it effectively models user preferences and behaviors, while cold-start strategies ensure the system remains functional even for new users or items with limited data.

The system design was preceded by a state-of-the-art review conducted using the PRISMA methodology, which guided architectural and algorithmic decisions with evidence-based support. Additional mechanisms, such as non-personalized ranking and post-processing penalization of overrepresented cities, contributed to enhancing recommendation diversity and avoiding redundancy.

Although the current system demonstrates robust and accurate recommendation capabilities, future enhancements could focus on incorporating more granular user preference data and real-time contextual information. Furthermore, integrating advanced AI tools such as deep learning models or more sophisticated machine learning techniques (e.g., neural collaborative filtering, attention mechanisms) could improve the adaptability and precision of recommendations. This project provided valuable practical and theoretical insights into the challenges and opportunities of applying recommender systems in the hospitality domain, offering a flexible, modular foundation for future academic and real-world developments.

References

[1] M. J. a. M. J. E. a. B. P. M. a. B. I. a. H. T. C. a. M. C. D. a. S. L. a. T. J. M. a. A. E. A. a. B. S. E. a. o. Page, "The PRISMA 2020 statement: an updated guideline for reporting systematic reviews," *bmj,* vol. 372, 2021.

[2] C. Dursun e A. Ozcan, "Sentiment-enhanced Neural Collaborative Filtering Models Using Explicit User Preferences," pp. 1-4, 2023.

[3] E. Wang, Y. Chen e Y. Li, "Research on a Hotel Collaborative Filtering Recommendation Algorithm Based on the Probabilistic Language Term Set," vol. 11, 2023.

[4] C. V. M. Krishna, G. A. Rao e S. Anuradha, "Analysing the impact of contextual segments on the overall rating in multi-criteria recommender systems," vol. 10, pp. 1-35, 2023.

[5] H. Shah e L. Jacob, "Hotel Recommendation System Based on Customer's Reviews Content Based Filtering Approach," pp. 222-226, 2022.

[6] J. Adamczak, Y. Deldjoo, F. B. Moghaddam, P. Knees, G.-P. Leyson e P. Monreal, "Session-based Hotel Recommendations Dataset: As part of the ACM Recommender System Challenge 2019," vol. 12, pp. 1-20.

[7] Y. Li, F. Zeng, N. Zhang, Z. Chen, L. Zhou, M. Huang, T. Zhu e J. Wang, "Multitask Learning Using Feature Extraction Network for Smart Tourism Applications," vol. 10, pp. 18790-18798, 2023.

[8] A. Solano-Barliza, A. Valls, A. Moreno, J. Dujmovic, M. Acosta-Coll, J. Escorcia-Gutierrez e E. De-La-Hoz-Franco, "Personalized Hotel Recommender System Based on Graded Logic with Asymmetric Criteria," vol. 246, pp. 2864-2873, 2024.