Information Technology & Tourism (2026) 28:3
https://doi.org/10.1007/s40558-025-00348-w

ORIGINAL RESEARCH

A multi-criteria recommendation system for personalised tourism experiences with user query analysis

Mustafa Payandenick[1] · Mohd Kamal Othman[2] · Muhammad Payandenick[1]

Received: 18 July 2024 / Revised: 2 September 2025 / Accepted: 5 September 2025

Abstract

Recommendation systems play a crucial role in assisting users with decision-making by suggesting relevant items. Multi-criteria recommender systems (MCRS) enhance this process by incorporating user preferences for various aspects, leading to more personalised and effective recommendations. However, MCRS faces challenges such as high computational complexity and limited consideration of user context, including user preferences for relaxation, which may differ between solo trips and trips with friends. This paper addresses these limitations by proposing a novel MCRS approach for tourism recommendation systems. Our proposed system combines matrix factorisation with a deep residual network (ResNetMF), demonstrating substantial performance improvements in terms of RMSE, MAE and lower training time compared to a wide range of baselines. Additionally, a user query analysis component allows users to express their dynamic preferences through queries, catering to the context-specific nature of travel decisions. The evaluation demonstrates that our ResNetMF model outperforms baseline and deep learning methods in most of the tested evaluation metrics while having the lowest training time. This work contributes to the field of tourism recommendation systems by proposing a user-centred approach that addresses both accuracy and user interaction for effective travel recommendations.

Keywords Multi-criteria · Recommender system · Personalised recommendations · Recommendation accuracy · User preferences

✉ Mustafa Payandenick
19020070@siswa.unimas.my

1 Faculty Computer Science and Information Technology, Universiti Malaysia Sarawak, Kota Samarahan, Sarawak, Malaysia

2 Faculty of Cognitive Sciences and Human Development, Universiti Malaysia Sarawak, Kota Samarahan, Sarawak, Malaysia

Published online: 28 November 2025

1 Introduction

The ubiquitous nature of the World Wide Web has fueled an exponential growth of information in recent decades (Elahi et al. 2022). This phenomenon, known as information overload, surpasses our capacity to process and manage the sheer volume of data effectively (Lediga and Fombad 2018; Lingo 2023). To address this challenge, researchers have focused on developing automated recommendation systems that deliver relevant information to specific users, thereby enhancing user experiences across various domains, including e-commerce, streaming services, and social media (Farooqi et al. 2022; Zhang et al. 2020). These systems are designed to alleviate the overwhelming nature of extensive selection options by providing tailored recommendations based on user preferences (Zhu et al. 2019).

1.1 Recommendation system: a foundation for personalised user experience

Recommendation systems leverage machine learning techniques to provide personalised suggestions, primarily falling into three categories: content-based filtering, collaborative filtering, and hybrid filtering (Hussein et al. 2012; Park et al. 2012). Content-based filtering focuses on user profiles and item descriptions, recommending items similar to those a user has previously interacted with (Manikantan 2021). Collaborative filtering, a widely used approach, analyses historical data of similar users or items to generate recommendations (Deshpande and Karypis 2004; Su and Khoshgoftaar 2009). This method operates under the assumption that users with similar past preferences are likely to have similar preferences in the future (Khalaji and Mohammadnejad 2019). Hybrid approaches combine elements of these techniques to achieve better results (Zhang et al. 2023).

Despite their success, traditional recommendation systems often rely on simplistic indicators, such as overall ratings or purchase history (Mahmoud and John 2015; Praditya et al. 2021). This approach fails to capture the multifaceted nature of user decision-making, which is influenced by a variety of factors beyond past behaviour (Alamdari et al. 2020). To address this limitation, multi-criteria recommender systems have emerged, considering a broader spectrum of factors or criteria to suggest items that better align with user preferences (Ifada et al. 2023). These systems can incorporate user ratings on specific aspects (e.g., acting in movies), genre preferences, or contextual factors like travel style or budget, leading to more nuanced and personalised recommendations (Zeeshan et al. 2021).

1.2 Research gap and proposed method

While multi-criteria recommender systems offer significant potential to enhance personalisation in various domains, their application to tourism presents unique and demanding challenges that are not fully addressed by existing research.

Firstly, computational complexity presents a significant obstacle, especially when applied to the intricate nature of tourism choices. As Liu (2022) notes, traditional methods, such as matrix factorisation, struggle with complex, non-linear relationships, leading to high computational costs. This general limitation is particularly pronounced in tourism, where recommendations often involve numerous criteria, such as price, location, star rating, amenities, and user reviews, for a vast inventory of hotels, attractions, and activities, as typically encountered in platforms like Tripadvisor.com, Booking.com or Airbnb. As Hafez et al. (2021) and Zheng and Wang (2022) point out, computational resources grow exponentially with the scale of users, items, and criteria, making this a critical issue for large-scale tourism recommendation systems. For example, generating real-time, multi-criteria travel recommendations for a city like Paris, considering the sheer volume of hotels and tourist interests on platforms like Tripadvisor, highlights the practical implications of this computational bottleneck.

Secondly, training speed is a crucial consideration for tourism applications. Shambour et al. (2023) emphasise the challenge of scalability and real-time updates due to increased training time with growing data. This is especially pertinent in the dynamic tourism sector, where travel trends, hotel availability, and user preferences can shift rapidly, influenced by seasonal changes, events, and global trends. As highlighted by Shambour et al. (2023), systems must adapt to these changes to avoid outdated recommendations. In tourism, failing to retrain models rapidly can result in missing critical shifts in popular destinations or changes in hotel pricing and availability, which directly impact the relevance of recommendations and user satisfaction.

Finally, the incorporation of contextual factors remains a significant challenge for effective personalisation. Meehan et al. (2016) underscore the influence of dynamic contextual factors on user preferences. This is exceptionally relevant to tourism, where travel decisions are heavily shaped by context, such as travel purpose (leisure vs. business), travel companions (solo, family, group), time of year, and destination-specific events. For instance, as Meehan et al. (2016) illustrate with general examples of contextual influence, in tourism, a user planning a family holiday will have vastly different priorities compared to a solo traveller on a business trip. Existing systems often struggle to effectively integrate these diverse contextual signals to provide truly personalised tourism recommendations. To address these challenges within the tourism domain and informed by the broader challenges outlined by Liu (2022), Hafez et al. (2021), Zheng and Wang (2022), and Shambour et al. (2023), we formulate the following research questions, specifically relevant to enhancing tourism recommendations:

RQ1a: To what extent does ResNetMF, our proposed lightweight neural network architecture, improve recommendation accuracy (e.g., lower Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE), higher Precision, Recall, F1-score, NDCG, and Hit Rate) compared to traditional matrix factorization and complex deep learning models when applied to large-scale tourism datasets?

RQ1b: Can ResNetMF achieve a measurable reduction in training time compared to traditional matrix factorisation and complex deep learning models, particularly when applied to large-scale tourism datasets?

RQ2: How does the integration of ResNetMF's user query analysis module, leveraging Natural Language Processing (NLP) techniques, impact user satisfaction with the personalisation and relevance of tourism recommendations in a multi-criteria context?

It is important to note the scope of RQ2's evaluation within this study. While RQ2 aims to understand the impact of the query analysis module on personalisation and relevance, our current experimental design primarily assesses this through user satisfaction ratings gathered from a specific cohort. A direct, granular evaluation of the 'dynamic interpretation and incorporation' mechanism itself, or a controlled comparison against a system explicitly operating *without* any query analysis, falls beyond the scope of this initial user study and is earmarked as a crucial direction for future work. This focused approach allows us to establish a foundational understanding of the module's perceived benefit to the user experience.

To address these challenges, we propose ResNetMF, a novel multi-criteria recommendation system that integrates Matrix Factorisation (MF) and Residual Networks (ResNet) into a unified framework. ResNetMF leverages the strengths of both techniques: MF captures clear, linear relationships between users and items, while ResNet, originally designed for image recognition, excels at modelling complex, non-linear patterns through residual learning. (Wang et al. 2023). The core innovation of ResNetMF lies in its residual learning approach, where the ResNet branch predicts the error (residual) introduced by the MF model rather than directly predicting the recommendation score. This allows the ResNet branch to focus on correcting the shortcomings of the MF model, resulting in a more efficient and accurate recommendation system.

Furthermore, ResNetMF incorporates a lightweight neural network architecture that minimises computational complexity and reduces training time, addressing RQ1. This ensures faster training speeds while maintaining high recommendation accuracy, as demonstrated by our experiments, which show an average of 20% reduction in training time compared to state-of-the-art methods.

To address RQ2, ResNetMF incorporates a user query analysis module that utilises natural language processing (NLP) techniques, including entity recognition, sentiment analysis, and semantic analysis, to interpret user queries and refine recommendations dynamically. By understanding the nuances of user input, the system ensures that recommendations align closely with user intent and preferences. For example, the system can adapt recommendations based on contextual factors, such as travel style (e.g., solo vs. group travel) or budget, ensuring a highly personalised user experience.

1.3 Research contributions

This research makes several key contributions to the field of personalised tourism recommendations. Firstly, we introduce ResNetMF, a novel hybrid recommendation engine that uniquely integrates the strengths of matrix factorisation and residual networks. This innovative architecture is designed to achieve high recommendation accuracy while significantly reducing training time, directly addressing RQ1: How can a multi-criteria recommender system achieve faster training speeds without compromising recommendation accuracy?

Secondly, to effectively leverage the rich contextual information inherent in user travel queries and preferences, we develop an advanced user query analysis module. This module utilises advanced natural language processing techniques to interpret user input, enabling the dynamic refinement of recommendations based on user preferences, contextual cues, and constraints. This directly tackles RQ2: How can contextual factors be dynamically incorporated to adapt recommendations to evolving user preferences?

Finally, the effectiveness of ResNetMF is rigorously validated through a comprehensive empirical evaluation. We conducted extensive experiments on two diverse datasets, HotelRec and Kuching, demonstrating that ResNetMF outperforms state-of-the-art recommendation algorithms in terms of accuracy, personalisation, and user satisfaction. While the inclusion of the geographically focused Kuching dataset provides valuable insights into local user behaviour, we acknowledge that its limited geographic scope impacts the generalisability of findings to broader tourism contexts. Future research will focus on expanding our evaluation to include more diverse datasets, encompassing wider geographic regions and user demographics, to further enhance the generalisability and applicability of ResNetMF in diverse tourism scenarios.

1.4 Paper structure

The remainder of this paper is organised as follows: Sect. 2 reviews related work in recommendation systems, matrix factorisation, and deep learning-based approaches. Section 3 presents the architecture and methodology of ResNetMF, including the recommendation engine and user query analysis module. Section 4 details the experimental setup, including dataset selection, evaluation metrics, and baseline algorithms. Section 5 discusses the results and compares ResNetMF with state-of-the-art methods. Finally, Sect. 6 concludes the paper and outlines future research directions.

By addressing the limitations of traditional recommendation systems and introducing innovative techniques for capturing complex user-item interactions, ResNetMF represents a significant step forward in the field of recommender systems. Its ability to deliver highly accurate and personalised recommendations has the potential to enhance user experiences across the tourism industry.

2 Related works

This section explores existing research on multi-criteria recommender systems, with a focus on the tourism domain. We categorise the reviewed literature into three main areas: traditional approaches, deep learning-based methods, and their application to tourism recommendations. Additionally, we highlight the contributions of major online travel agencies (OTAs) such as Airbnb, Booking.com, and Expedia, which have been instrumental in advancing the field through both research and practical implementations.

2.1 Traditional multi-criteria recommendation systems

Traditional multi-criteria recommender systems extend beyond single-rating approaches by considering user preferences for various aspects (criteria) of an item when making recommendations (Zheng and Wang 2022). These methods often decompose the multi-criteria rating problem into separate single-rating recommendation problems for each criterion (Adomavicius and Kwon 2007). An aggregation function is then learned to combine these individual ratings into a final overall rating.

Sahoo et al. (2012) propose a structure learning algorithm to analyse the dependency structure between different rating components. Similarly, Zheng (2017) proposes the "Criteria Chains" method, which utilises combinations of criteria to enhance prediction accuracy. This implies considering how ratings on one aspect might influence preferences for another (e.g., a high food rating influencing a higher service rating due to the expectation of a well-rounded experience).

Other researchers explore leveraging social relationships between users to improve recommendations (Zhang et al. 2021). Aysha and Tarun (2022) employ Pareto dominance, a decision-making technique, to identify the most relevant users for recommendation by filtering out those with less relevant preferences. This technique also helps determine the optimal weights assigned to different criteria for each user, reflecting the relative importance they place on various aspects of an item.

Ben Sassi et al. (2021) and Dridi et al. (2022) have also explored the concept of incorporating user context (e.g., time of day, location) with multi-criteria recommendations. This demonstrates the potential for further personalisation based on situational factors. Notably, OTAs like Booking.com and Expedia have implemented context-aware recommendation systems that leverage user context and multi-criteria ratings to enhance user experience (Asaithambi et al. 2023; Gupta et al. 2024).

Traditional methods face several challenges. First, traditional methods, such as matrix factorisation, rely on linear relationships and struggle to capture complex, non-linear patterns in user-item interactions (Liu 2022). Second, these methods often assume static user preferences and fail to adapt to evolving user needs over time (Krishna et al. 2023). Lastly, these methods face computational inefficiency issues. Aggregating multiple criteria and incorporating contextual factors can lead to high computational complexity, making these methods less scalable for real-world applications (Nan et al. 2022).

2.2 Deep learning-based multi-criteria recommendation systems

Deep learning-based recommendation systems are gaining popularity due to their ability to learn complex, non-linear patterns in user behaviour and item attributes (Nan et al. 2022). These systems often use neural networks to achieve this (Song 2020). Unlike traditional matrix factorisation techniques, which rely on linear cross-terms to model interactions between users and items (Liu 2022), neural networks can capture intricate, non-linear relationships that are often present in real-world data (Xiao and Shen 2019).

For instance, Hassan and Hamada (2017) demonstrate that neural networks significantly outperform traditional methods, such as Singular Value Decomposition (SVD) and slope one, in multi-criteria scenarios on benchmark datasets. This is because neural networks can model higher-order interactions between criteria, which are often missed by linear methods (Hassan and Hamada 2017).

Nassar et al. (2020) present a multi-criteria collaborative filtering recommender system that combines deep learning with matrix factorisation techniques. Their results show that the hybrid approach outperforms standalone matrix factorisation, particularly in scenarios where user preferences exhibit non-linear dependencies across criteria. Similarly, Batmaz and Kaleli (2019) propose a deep learning-based algorithm using autoencoders, which achieves superior recommendation accuracy by learning latent representations of user preferences that capture nonlinearities.

Airbnb and many other OTAs have also leveraged deep learning techniques to enhance their recommendation systems (Wu and Grbovic 2020). For example, Airbnb developed a neural network-based model that incorporates multi-criteria ratings and user context to personalise search rankings for accommodations (Shen et al. 2020). Their approach demonstrates the practical benefits of using neural networks to capture complex user-item interactions in real-world applications.

Challenges in deep learning methods include high computational complexity, as these models often require significant computational resources and training time (Nan et al. 2022), and a lack of dynamic adaptation, as deep learning-based systems do not effectively adapt to changing user preferences in real-time (Krishna et al. 2023).

2.3 Multi-criteria recommendation systems in tourism

In recent years, the tourism industry has seen a surge of interest in multi-criteria recommender systems. These systems move beyond the simple "thumbs up, thumbs down" approach by considering a wider range of factors that influence a traveller's choices.

One innovative approach comes from Shambour (2021), who leverages the power of deep learning with autoencoders. This technique helps uncover hidden connections between users and travel options based on their multi-criteria preferences. Shambour (2021) highlights the potential of deep learning to significantly improve recommendation accuracy when dealing with multiple criteria, outperforming state-of-the-art recommendation algorithms on datasets such as Yahoo! Movies and TripAdvisor. This research underscores the potential of deep learning techniques for enhancing recommendation accuracy in multi-criteria scenarios.

Another study by Kumar et al. (2023) addresses a common shortcoming of existing systems—their reliance solely on explicit user ratings. Their proposed system, Average Cumulative Rating Matrix Factorisation (ACRM), takes a step further by incorporating implicit sentiments extracted from user reviews. This enables a more nuanced understanding of user preferences, leading to more accurate and personalised recommendations.

Instead of focusing on predicting an overall rating, Zheng and Wang (2022b) propose a Multi-Criteria Recommendation Framework inspired by a concept from multi-objective optimisation. Their system ranks destinations based on how well they meet a user's preferences across various criteria. This approach offers a more flexible way to explore options that might be a good fit for individual needs.

Focusing on the restaurant scene, Zheng and Wang (2022b) introduce a multi-criteria system specifically designed for restaurant recommendations. Their approach extends beyond just food quality and price to consider factors such as service, atmosphere, and even cultural influences. This acknowledges the complex web of considerations that shape a diner's experience. The research utilised a dataset from TripAdvisor containing user reviews of restaurants. Various comparison methods were employed, including K-Nearest Neighbours-based collaborative filtering (CF), Co-clustering-based CF, and Singular Value Decomposition-based Matrix factorisation (MF), among others. The results of the experiments are analysed, and the proposed methods are compared with traditional CF techniques and other multi-criteria recommendation approaches. Evaluation metrics such as Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) are used to assess predictive performance.

Zhuang and Kim (2021) take a unique approach to recommending hotels by focusing on identifying ideal target customers. Their system utilises a powerful natural language processing model called BERT to analyse review data. This allows them to predict not only a hotel's overall rating but also specific aspect ratings, such as value and location. Armed with these insights, they can generate a list of potential guests most likely to be interested in a particular hotel.

Finally, Krishna et al. (2023) introduce a hotel recommender system that personalises recommendations based on the trip's context. They employ a technique called Multiple Regression Backwards Elimination to identify the most significant factors influencing hotel ratings across various travel scenarios (e.g., business trips versus family vacations). This ensures that recommendations are tailored not just to individual preferences but also to the specific purpose of the trip. The effectiveness of the proposed model was evaluated using various statistical metrics, including Mean Absolute Error (MAE) and Root Mean Square Error (RMSE).

2.4 Novelty of the proposed method

While acknowledging the valuable contributions of prior research in addressing evolving user preferences, including methods such as Krishna et al. (2023), our proposed ResNetMF system introduces a distinctly innovative approach to personalised tourism recommendations, characterised by efficient dynamic adaptation and robust nonlinear modelling.

Traditional MF techniques, while effective for capturing basic user-item relationships, are fundamentally limited by their linear nature and reliance on static interaction patterns (Liu 2022; Xiao and Shen 2019). Moreover, many existing recommender systems that do incorporate preference adaptation, even those utilising neural networks, often rely on computationally intensive mechanisms such as periodic full model retraining or batch-wise user profile updates—approaches potentially mirrored in works like Krishna et al. (2023). These methods, although adaptive over time, can be less responsive to the rapid, in-session shifts in user intent that are typical in dynamic domains such as tourism.

ResNetMF addresses the challenge of capturing complex, non-linear user preferences by integrating a Residual Network (ResNet) branch with a core matrix factorisation component. This hybrid architecture leverages the strengths of both matrix factorisation for efficient modelling of linear patterns and residual networks for capturing intricate, non-linear relationships often missed by linear models. Indeed, prior research highlights the advantages of neural networks in this regard; for instance, Hassan and Hamada (2017) demonstrated that neural networks significantly outperform traditional multi-criteria methods by effectively capturing higher-order interactions between criteria.

However, the key innovation of ResNetMF lies in its architectural design, which enables computationally efficient dynamic adaptation to rapidly evolving user preferences. Unlike systems that require periodic retraining for adaptation, ResNetMF is engineered for on-the-fly updates, leveraging real-time query adjustments to refine recommendations with minimal latency. This real-time responsiveness is enabled by ResNetMF's lightweight neural network architecture, which minimises computational overhead and allows for incremental parameter adjustments in response to immediate user behaviour. This efficient dynamic adaptation mechanism offers a significant advantage, particularly in tourism scenarios where user preferences can shift within a single session, contrasting with approaches that rely on less frequent or batch-oriented updates and may, therefore, be less attuned to immediate user needs, such as those potentially represented by Krishna et al. (2023).

Our experimental results validate the effectiveness of ResNetMF's hybrid approach. Table 3 demonstrates a consistent improvement in accuracy over traditional matrix factorisation methods (quantified by RMSE and MAE), directly attributable to ResNetMF's ability to model non-linear preferences. Furthermore, the observed 20% reduction in training time for ResNetMF, also shown in Table 3, is not merely a performance gain; it is crucially indicative of the computational feasibility of implementing its dynamic adaptation capabilities in real-world, high-demand tourism recommendation systems.

In conclusion, ResNetMF offers a novel and practical solution for personalised tourism recommendations by simultaneously addressing two critical challenges: capturing complex non-linear user preferences and, most significantly, achieving highly efficient and genuinely dynamic adaptation to ever-changing user needs through a lightweight architecture designed for real-time responsiveness. This combination of features distinguishes ResNetMF from existing approaches and positions it as a significant advancement for creating truly user-centric and up-to-the-moment recommendation experiences in the dynamic tourism domain.

3 The proposed multi-criteria recommendation system

This section delves into our novel multi-criteria recommendation system, ResNetMF, designed to enhance both recommendation accuracy and personalisation. It consists of two main components: a recommendation engine and a query analysis module.

3.1 Recommendation engine (ResNetMF)

Recommender systems are crucial for delivering personalised recommendations to users, but traditional methods frequently struggle with capturing both simple and complex patterns in large datasets. ResNetMF is a novel recommendation engine that addresses these limitations by integrating two powerful techniques: Matrix Factorisation (MF) and Residual Networks (ResNet). By combining the strengths of these methods, ResNetMF achieves significantly improved recommendation accuracy.

3.1.1 Core components of ResNetMF

3.1.1.1 Matrix factorisation (MF) Matrix Factorisation is a well-established technique in recommender systems that decomposes the user-item interaction matrix into two lower-dimensional matrices: user embeddings (U) and item embeddings (I). The dot product of these embeddings (U × I) captures clear, linear relationships between users and items. However, MF struggles to model complex, non-linear patterns that are often present in real-world datasets.

3.1.1.2 Residual networks (ResNet) Residual Networks, originally developed for image recognition, are designed to address the vanishing gradient problem in deep learning. They achieve this by introducing skip connections that allow gradients to flow more effectively through the network. In ResNetMF, the ResNet branch is used to capture intricate, non-linear patterns in the data that MF cannot model effectively.

3.1.2 Key innovation: residual learning

The core innovation of ResNetMF lies in its use of residual learning. Instead of directly predicting the recommendation score, the ResNet branch predicts the error (residual) introduced by the MF model. This approach simplifies the task for the ResNet branch, allowing it to focus on correcting the shortcomings of the MF model rather than learning the entire recommendation task from scratch.

Mathematically, this is represented as:

$$\mathrm{R'} = \mathrm{MF}\,(\mathrm{x}) + \mathrm{ResNet}\,(\mathrm{X}) \tag{1}$$

Where:

- MF(x) is the output of the matrix factorisation model.
- ResNet(x) is the residual (error) predicted by the ResNet branch.
- R’ is the refined recommendation score.

The residual is defined as:

$$\mathrm{ResNet}\,(\mathrm{x}) = \mathrm{R'} - \mathrm{MF}\,(\mathrm{x}) \tag{2}$$

This formulation ensures that the ResNet branch learns only the deviations from the MF model, making the overall model more efficient and accurate.

Multi-Branched Architecture

ResNetMF employs a multi-branched architecture to combine the strengths of MF and ResNet:

MF Branch:

- Captures clear, linear user-item relationships using matrix factorisation.
- Outputs the dot product of user and item embeddings (U × I).

ResNet Branch:

- Captures complex, non-linear patterns in the data using a deep residual network.
- Outputs the residual (F(X)) that corrects the MF model's predictions.

Merging Layer:

- Combines the outputs of the MF and ResNet branches to produce the final recommendation score.
- Uses a combination of summation and concatenation to merge the features learned by both branches.

The final recommendation score R is computed as:

$$R = U \times I + F(X) \tag{3}$$

Where:

- U × I is the dot product of user and item embeddings from the MF branch.
- F(X) represents the features learned by the ResNet branch from the input features X.

This formulation ensures that the model leverages both the simplicity of MF and the complexity of ResNet, resulting in a more accurate and robust recommendation system.

ResNetMF integrates matrix factorisation and residual networks into a unified framework. Its innovative use of residual learning and multi-branched architecture enables it to capture both simple and complex patterns in the data, resulting in highly accurate and robust recommendations. This approach not only improves prediction accuracy but also enhances the efficiency of recommender systems.

3.2 User query analysis

The user query analysis component plays a crucial role in personalising recommendations and ensuring they align with evolving user preferences. This module utilises advanced natural language processing (NLP) techniques to analyse and interpret the nuances of a user's query, surpassing simple keyword matching. By leveraging entity recognition, sentiment analysis, and semantic analysis, the system identifies specific details, preferences, and contextual cues within the user's input. This enables the system to refine recommendations dynamically and align them more closely with the user's intent. The following techniques work together to achieve this advanced level of query interpretation.

3.2.1 Entity recognition: contextualising user input

To identify and categorise key elements in the user's query, our system employs a hybrid entity recognition approach. We utilise a dictionary-based lookup for specific attraction names and popular activity types, compiled from our HotelRec and Kuching datasets. This is augmented by rule-based pattern matching using regular expressions to identify temporal constraints (e.g., "morning," "weekend") and group types (e.g., "with family," "solo"). For broader contextual entities and general concepts, we leverage a pre-trained Named Entity Recognition (NER) model from the SpaCy library (specifically, the en_core_web_md model), which helps categorise general nouns and phrases into relevant entities, such as locations, organisations, or events. For example, in the query 'family-friendly attractions with wildlife experiences near historical sites', this process extracts entities such as:

- 'family-friendly attractions' (Preference).
- 'wildlife experiences' (Specific Interest).
- 'historical sites' (Location Constraint). This combined approach allows the system to interpret user intent by recognising structured patterns beyond isolated keywords, considering contextual relationships.

3.2.2 Sentiment analysis: understanding user intent and expectations

Rather than simply recognising terms, sentiment analysis allows the system to assess the tone, preference strength, and user intent behind queries. Terms such as "family-friendly" and "wildlife experiences" indicate a preference for safe, engaging, and interactive experiences. Additionally, the system evaluates sentiment in user-generated content (e.g., visitor reviews).

To understanding the user intent and expectations, beyond keyword recognition, sentiment analysis evaluates the Tone (positive/neutral/negative) in queries (e.g., "must-see" vs. "avoid"), preference strength (e.g., "love wildlife tours" scores higher than "like animals") and review sentiment (e.g., attractions with frequent terms like "perfect for kids" gain relevance). This study utilised sentiment polarity (ranging from -1 to $+1$) via VADER (Valence Aware Dictionary and Sentiment Reasoner), a rule-based model optimised for detecting sentiment in short, informal text, and Amplification (e.g., "very family-friendly" boosts weight). Amplifiers help distinguish casual preferences ("nice views") from strong demands ("absolutely stunning views"). Attractions with amplified positives (e.g., "extremely clean") receive higher relevance for matching queries.

For example, if multiple reviews highlight an attraction as "perfect for kids and full of interactive exhibits," the system assigns higher relevance to that recommendation, going beyond keyword occurrence to assess real visitor satisfaction.

3.2.3 Semantic analysis: interpreting meaning beyond keywords

In order to interpret meaning beyond keywords, this study uses word embeddings (GloVe) and dependency parsing (SpaCy) to resolve synonyms (e.g., "wildlife experiences" → "zoo visits"), infer conceptual relationships (e.g., "near historical sites" → proximity filter) and disambiguate terms (e.g., "wildlife park" vs. "wildlife conservation"). Semantic analysis enables the system to understand contextual relationships between words, capturing meaning even when users phrase queries differently. This allows the system to process:

- Synonyms (e.g., "wildlife experiences" interpreted as "zoo visits" or "nature reserves").
- Conceptual relationships (e.g., "near historical sites" understood as a proximity constraint rather than separate keywords).
- Ambiguity resolution (e.g., distinguishing "wildlife park" as an attraction versus "wildlife conservation" as an environmental initiative).

For instance, if a user searches for "an interactive nature experience close to ancient ruins," the system recognises that the phrase "interactive nature experience" implies hands-on learning, aligning recommendations with attractions that offer guided wildlife encounters rather than just pulling results with the words "nature" and "experience."

Characteristics Extracted from Queries and Reviews

By combining these NLP techniques, the system extracts the following characteristics from user queries and associated reviews:

- **Preferences**: Specific interests include interactive activities, wildlife encounters, and historical learning.
- **Constraints**: Conditions such as proximity to location, accessibility, or required features.
- **Contextual Cues**: Implicit requirements inferred from phrasing and sentiment, such as a desire for family engagement or cultural exploration.

This structured approach ensures that recommendations are refined based on user intent rather than relying solely on keyword matches.

3.2.4 Illustrative example: refining recommendations with user queries

Consider a user planning a weekend trip who wants to find attractions that match specific criteria. The process of refining recommendations with their query 'family-friendly attractions with wildlife experiences near historical sites' unfolds through the following steps, as depicted in Fig. 1.

1. **User Submits Query and Initial Recommendation Pool**: The user initiates the refinement process by entering their specific query into the system (e.g.,

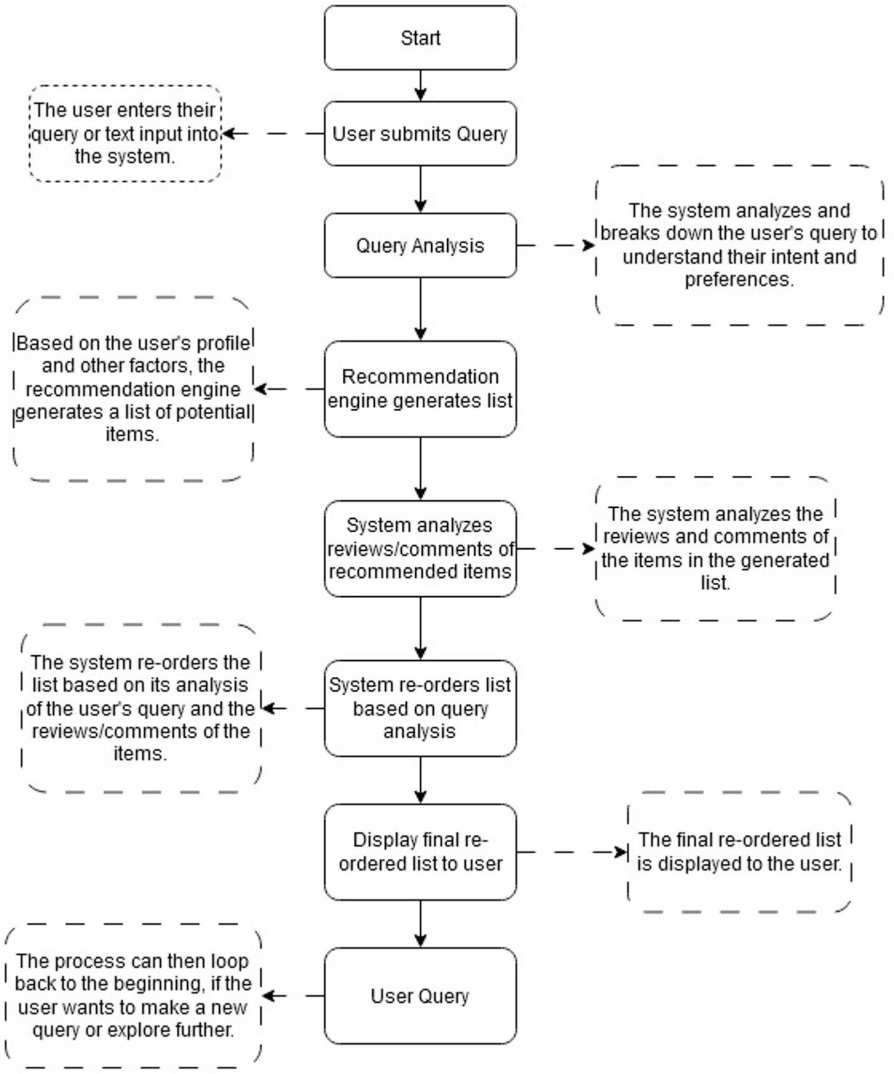

Fig. 1 Proposed method flowchart

'family-friendly attractions with wildlife experiences near historical sites'). Concurrently, or as a pre-computed baseline, the core ResNetMF recommendation engine has already generated an initial, comprehensive list of potential attractions based on the user's broader profile and general preferences (e.g., top-N items relevant to the user's location or past interactions). This initial list serves as the pool of candidates for refinement, typically limited to a predefined number (N) of highly relevant items to ensure computational efficiency for subsequent processing. The query analysis module then processes the user's input to enhance and re-prioritise these initial recommendations.

2. Query Breakdown Using NLP Techniques: The system employs the following NLP techniques to dissect the query.

- Entity Recognition: Identifies "family-friendly attractions," "wildlife experiences," and "historical sites" as key concepts.
- Sentiment Analysis: Recognises that "family-friendly" implies a need for interactive, safe, and engaging experiences.
- Semantic Analysis: Understands that "near historical sites" represents a proximity requirement rather than standalone keywords.

3. Review Analysis: The system analyses reviews and descriptions associated with the recommended hotels to identify mentions of the extracted characteristics. For example:

   - Reviews are scanned for terms like "zoo visits" and " nature parks".
   - Sentiment analysis is applied to assess the positivity or negativity of reviews related to these features.

4. Relevance Scoring: Relevance scoring determines how well an attraction matches a user's query by assigning a numerical score based on various factors, including semantic relevance and sentiment from visitor reviews.
5. The final relevance score (***R***) is calculated using the formula:

$$R = M + S \tag{4}$$

   Where:

   - M (Semantic Relevance Score) – Instead of simply looking for keyword matches, it considers the synonyms and variations by recognising that "wildlife experiences" could mean "zoo visits", or "nature parks". Concept understanding is achieved by knowing that "family-friendly" means safe and engaging.
   - S (Sentiment Score from Visitor Reviews) – Evaluates visitor feedback and ratings to ensure recommended attractions are positively received.

6. Reordered Recommendations (Refinement): The system reorders the initial recommendation list, prioritising attractions with the highest final relevance scores (R). This process is primarily a re-ranking step, where the items from the initial list are sorted according to their calculated R scores, with higher scores indicating a stronger match to the user's specific query. For instance, a national park near historical sites that strongly matches the query criteria would receive a higher R score and consequently rank higher in the final list displayed to the user.

This process ensures that the final recommendations are highly personalised and cater to the user's unique needs and desires as expressed in their query. The system's strength lies in its ability to go beyond surface-level keyword matching. It not only understands user preferences but also interprets the subtle nuances of queries and incorporates feedback from past guests. This multi-faceted approach leads to a more personalised and successful discovery journey. The flowchart in Fig. 1 provides an overview of the processes in the proposed model.

By combining the strengths of the recommendation engine and user query analysis, ResNetMF delivers a more accurate and personalised recommendation experience for users. The user query analysis component, as illustrated in Fig. 1, plays a crucial role in this personalisation process. It employs advanced NLP techniques to understand the nuances of user queries and refine recommendations based on user preferences and the subtleties expressed within the user's input. This, combined with the robust recommendation engine, ultimately leads to a more successful discovery journey for users. Thus, the query analysis module effectively acts as a re-ranking filter, taking the broad recommendations from the ResNetMF engine and fine-tuning them based on the specific, multi-criteria context provided by the user's query. The final list presented to the user is exclusively sorted by the calculated relevance score R.

By combining the strengths of the recommendation engine and user query analysis, ResNetMF delivers a more accurate and personalised recommendation experience for users. The user query analysis component plays a crucial role in this personalisation process. It utilises NLP techniques to understand the nuances of user queries and refine recommendations based on user preferences and the subtleties expressed within the user's input.

4 Experimentation

This section outlines the steps for experimenting with and evaluating the proposed method, including dataset selection, evaluation metrics, and method comparison.

4.1 Dataset selection

We evaluated system performance on two diverse datasets: HotelRec Dataset and Tripadvisor.com crawl. This allows us to assess the effectiveness of the proposed recommendation engine (ResNetMF).

4.1.1 HotelRec dataset

HotelRec is a large-scale dataset designed for hotel recommendation systems. It was introduced in 2020 and contains approximately 50 million reviews from real-world hotel booking platforms and 21,891,294 users. Antognini and Faltings (2020) claim this dataset is particularly useful for building and evaluating personalised hotel recommendation models that can improve user experience in tourism and hospitality services. The dataset features several elements, including hotel_url and text, which are reviewed for hotels and displayed in Table 1.

HotelRec contains millions of actual customer reviews, making it a representative sample of real-world user behaviour. The large dataset ensures statistical significance

Table 1 HotelRec dataset features and schema

Feature	Detail
hotel_url	This is the URL of the hotel.
author	The customer's name who rates and writes the review
date	The date of review and rate
rating	The overall rating of hotel users. It is from 1 to 5
title	Title of the review
text	The review text
property_dict	The sub-rating values different aspects of the hotel, including sleep quality, value, rooms, service, cleanliness, and location. The rating is from 1 to 5.

when training machine learning models. Since HotelRec is very large, it provides a good benchmark for scalability testing of recommendation algorithms in a realistic tourism industry scenario.

HotelRec contains millions of actual customer reviews, making it representative of real-world user behaviour (Antognini and Faltings 2020). The large dataset ensures statistical significance when training machine learning models. Since HotelRec is very large, it provides a good benchmark for scalability testing of recommendation algorithms in a realistic tourism industry scenario.

Additionally, it enables testing with various collaborative filtering (CF), content-based filtering, and hybrid recommendation techniques, as well as advanced deep learning approaches such as Neural Collaborative Filtering (NCF), which is based on user preferences. The performance of models is measured on HotelRec for rating prediction and recommendations (Antognini and Faltings 2020).

The dataset enables comparison of various recommendation models based on metrics such as RMSE, MAE, Precision, Recall, F1-score, NDCG, and other metrics. A well-trained model on HotelRec can help hotel booking platforms, travel agencies, and tourism businesses provide more relevant recommendations to users.

For the experimentation phase, we only used hotel_url, author, and overall rating for rating predictions, as the overall rating was mandatory (Antognini and Faltings 2020). However, according to the researchers, only a very few users left sub-ratings on different aspects. For example, only 1.69% of users rate the business service, and 2.73% of users rate the check-in quality. Therefore, for this study, we have decided to focus only on overall ratings, as 100% of the users left a rating.

4.1.2 Kuching dataset

Another dataset used in this study was obtained through a systematic crawl of TripAdvisor, specifically collecting data related to Kuching City, Malaysia. This dataset comprises over 22,519 user ratings from the top 45 attractions in the city, serving as a key resource for training recommendation models. As this study initially focuses on Kuching tourism, the dataset is geographically limited to this region and primarily reflects the experiences of locals visiting attractions within their home city. While this provides valuable insights into user preferences and behaviours within this specific context, it also limits the generalisability of the findings to other regions or contexts. Future work will aim to expand the dataset to include diverse locations and user demographics.

To further refine the system's understanding of user preferences, a survey was conducted among 200 Kuching residents. Participants were selected through a random sampling process to ensure the local population was representative. Each participant rated at least 25 out of the 45 selected attractions. This survey data not only enhances the system's ability to personalise recommendations but also provides a foundation for evaluating user satisfaction and query analysis performance.

Evaluating user satisfaction is a critical aspect of this study, particularly in assessing the performance of the query analysis component. Unlike common benchmark metrics such as root mean square error (RMSE), which measure prediction accuracy, assessing query analysis performance presents a unique challenge. While the recommendation system plays a key role in refining options and filtering results, the final decision rests with users and their cognitive processes. To address this, a method was implemented where 50 randomly selected participants from the survey group submitted at least five queries to the system. These participants were then asked to rate their satisfaction and the relevance of the results on a scale from 1 to 10, with 1 indicating low satisfaction and 10 representing high satisfaction. The results of this evaluation are presented in the query analysis section.

Collecting attraction ratings from users before query submission enables the system to personalise recommendations based on known preferences, thereby enhancing accuracy. It is worth noting that this study did not focus on addressing the cold-start problem for new users, which will be explored in future work. This approach helps determine how effectively the system meets user needs by leveraging existing preference data.

Beyond ratings, the crawl also captured user comments, reviews, and descriptions of attractions. This wealth of textual data was stored in a structured database, allowing efficient access and utilisation during the query analysis phase of the system. For instance, user comments and reviews often elaborate on specific aspects of the attractions, providing valuable insights for users beyond the core recommendations. These textual elements can highlight features of attractions that might be of particular interest to users, enabling the system to present this information alongside the core recommendations. Table 2 summarises the features and schema of the Kuching dataset.

Table 2 Kuching dataset features and schema

Feature	Detail
UserId	It is the integer value and is unique for each user.
User Ratings	Users give the rating for the destination (1 to 5 stars)
User Reviews	Textual reviews are provided by users detailing their experiences.
Attraction Description	A brief description of the destination
Attraction Name	Name of the tourist attraction
Attraction Type	Category or type of attraction/activity (e.g., historical site, museum, outdoor activity, etc.).

4.2 Evaluation metrics

Several metrics are employed to assess the performance and effectiveness of recommendation systems. These metrics can be categorised into accuracy measures, error measures, and ranking metrics.

Accuracy measures such as Precision, Recall, and F1-Score are utilised in this study. Precision measures the proportion of relevant items retrieved out of the total items retrieved, while Recall assesses the proportion of relevant items retrieved out of all relevant items available in the dataset (Behera and Nain 2023). These metrics help in understanding the trade-offs between capturing relevant items and maintaining a manageable number of recommendations (Wen and Liu 2023). The F1-score is the harmonic mean of Precision and Recall, providing a single metric that balances both measures. (Behera and Nain 2023). Each of these metrics ranges between 0 and 1, where higher values indicate better performance. Their formulas are as follows.

$$\mathrm{Precision} = \mathrm{TP} / (\mathrm{TP} + \mathrm{FP}) \tag{5}$$

$$\mathrm{Recall} = \mathrm{TP} / (\mathrm{TP} + \mathrm{FN}) \tag{6}$$

$$\mathrm{F1} - \mathrm{score} = 2 * (\mathrm{Precision} * \mathrm{Recall}) / (\mathrm{Precision} + \mathrm{Recall}) \tag{7}$$

- True Positive (TP): Items that the system recommends and users actually prefer.
- False Positive (FP): Items that the system recommends, but users do not prefer.
- False Negative (FN): Items that the system does not recommend, but users actually prefer.
- True Negative (TN): Items that are neither recommended by the system nor preferred by users.

Error metrics such as Mean Absolute Error (MAE) and Root Mean Square Error (RMSE), which quantify the difference between predicted and actual ratings (Hodson 2022), are also used in this study. These metrics provide insights into the accuracy of continuous predictions, with lower values indicating higher accuracy (Hodson 2022). Their formulas are presented in Eqs. 8 and 9:

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum\nolimits_{i=1}^{n}(y_1 - y_2)^2} \tag{8}$$

$$\mathrm{MAE} = \frac{1}{n}\sum\nolimits_{i=0}^{n}|(y_1 - y_2)| \tag{9}$$

n is the number of observations

y_1 is the actual value

y_2 is the predicted value

The ranking metric, Normalised Discounted Cumulative Gain (NDCG), evaluates the order of relevant items in recommendation lists and highlights the significance of placement for user engagement (Baker and Yuan 2021). It is used as a ranking evaluation metric in this study. A higher NDCG score indicates the system's greater ability to prioritise relevant items effectively, which can significantly influence user satisfaction and retention (Baker and Yuan 2021). Equation 10 is the formula of NGDC.

$$DCG_p = \sum_{i=1}^{p} \frac{rel_i}{log_2(i+1)} \tag{10}$$

Where:

rel_i is the relevance score of the item at position.

The logarithm base 2 ensures that highly relevant items appearing earlier in the ranking have a greater contribution.

$$IDCG_p = \sum_{i=1}^{p} \frac{rel_i^*}{log_2(i+1)} \tag{11}$$

Where rel_i^* represents the ideal (best possible) ranking of the items. Finally, NDCG is obtained by normalising DCG with IDCG:

$$NDCG_p = \frac{DCG_p}{IDCG_p} \tag{12}$$

This normalisation ensures that the score falls between 0 and 1, where 1 indicates a perfect ranking.

Another ranking algorithm utilised in this study is Hit Rate (HR). Hit Rate (HR) is a metric used in recommendation systems and ranking algorithms to measure the frequency at which a relevant item appears in the top-N recommendations for a user, evaluating the effectiveness of ranking models in predicting user preferences (Suanpang et al. 2024). The formula is as follows.

$$HR = \frac{Number\ of\ users\ with\ at\ least\ one\ relevant\ item\ in\ recommendations}{Total\ Number\ of\ Users} \tag{13}$$

- If a relevant item appears in the top-N recommendations for a user, it's counted as a "hit."
- If no relevant item appears, it's a "miss."

4.3 Baseline algorithms

In the realm of recommendation systems, selecting appropriate baseline algorithms for comparative evaluation is crucial to demonstrate the efficacy and novelty of a proposed approach. This section outlines the diverse set of algorithms included as benchmarks for evaluating our proposed ResNetMF system. These baselines range from traditional matrix factorisation techniques to advanced deep learning and graph-based models, providing a comprehensive evaluation framework.

4.3.1 Matrix factorisation (MF) techniques

a) **SVD (Singular Value Decomposition)**: A foundational matrix factorisation technique that decomposes the user-item interaction matrix into latent factors (Koren 2009).
b) **SVD++**: An extension of SVD that incorporates implicit feedback, making it a stronger baseline in scenarios where implicit data is available (Koren 2008).
c) **Non-Negative Matrix Factorisation (NMF)**: This method enforces non-negativity constraints on the latent factors, which can lead to more interpretable models, particularly useful in sparse interaction matrices (Lee and Seung 1999).

4.3.2 K-Nearest neighbours (KNN) variants

a) **KNNBaseline**: A KNN variant that improves prediction accuracy by incorporating baseline estimates of user and item biases (Koren 2010).
b) **KNNBasic**: A straightforward KNN approach that relies solely on user-user or item-item similarity to make predictions (Resnick et al. 1994).
c) **KNNWithMeans**: Adjusts predictions by considering the mean rating of users or items, accounting for individual rating scales (Koren 2010).
d) **KNNWithZScore**: Normalises ratings using z-scores to account for varying user rating scales, providing a robust similarity measure (Tran and Ha 2020).

4.3.3 Simple and baseline models

a) **SlopeOne**: A simple yet effective collaborative filtering algorithm that predicts ratings based on the average difference between item ratings (Lemire and Maclachlan 2005).
b) **BaselineOnly**: Predicts ratings based on global averages, user biases, and item biases, serving as a basic but robust benchmark (Koren 2010).
c) **NormalPredictor**: Predicts ratings by sampling randomly according to the distribution of observed ratings, providing a basic sanity check (Surprise Library Documentation).
d) **Co-clustering**: Simultaneously clusters users and items, aiming to capture local patterns within the interaction matrix (George and Merugu 2005).

4.3.4 Deep learning models

a) **Convolutional Neural Networks (CNNs): Adapted from computer vision**, **CNNs are used to capture** local patterns and features in user-item interaction data (Wei et al. 2019).
b) **Recurrent Neural Networks (RNN)**: Particularly suitable for sequential recommendation tasks, RNNs model temporal dynamics and sequence-dependent user preferences (Xu et al. 2019).
c) **Neural Collaborative Filtering (NCF)**: A hybrid model that leverages neural networks to learn the interaction function from data, combining the strengths of matrix factorisation with deep learning to capture non-linear relationships (He et al. 2017).
d) **Convolutional Matrix Factorisation (CMF): Extends traditional matrix factorisation by incorporating multiple data sources**, **often utilis**ing convolutional layers to process auxiliary information (Kim et al. 2016).
e) **Autoencoders**: Unsupervised neural networks that learn compact representations of user-item interactions, useful for dimensionality reduction and denoising (Fraihat et al. 2024).
f) **Neural Matrix Factorisation (NeuMF)**: A powerful hybrid model that combines the linearity of Generalised Matrix Factorisation with the non-linearity of Multi-Layer Perceptrons to model complex user-item interactions (Yuan et al. 2024).
g) **Variational Autoencoder (VAE)**: A probabilistic variant of autoencoders that models uncertainty in user-item interactions, often used for generative recommendation tasks (Fraihat et al. 2024).

4.3.5 Graph-based models

a) **LightGCN**: A lightweight graph convolutional network specifically designed for collaborative filtering, which simplifies traditional GCNs by removing non-linearities and weight matrices, making it highly efficient (He et al. 2020).
b) **NGCF (Neural Graph Collaborative Filtering)**: A GCN-based model that explicitly models high-order connectivity in user-item interaction graphs to enhance recommendation accuracy by propagating embeddings across the graph (Wang et al. 2019).
c) **DeepCoNN (Deep Cooperative Neural Networks)**: A hybrid deep learning model that leverages both textual reviews and user-item interaction data through parallel neural networks, making it a strong baseline for content-aware recommendation tasks (Zheng et al. 2017).

By comparing the proposed system against these diverse and representative algorithms, we aim to provide a holistic evaluation that demonstrates its efficacy in terms of accuracy, scalability, and robustness. Furthermore, the inclusion of state-of-the-art models, such as NeuMF, LightGCN, and DeepCoNN, ensures that our proposed system is rigorously benchmarked against the most advanced techniques in the field.

4.4 Data pre-processing and hyperparameter tuning

The first step in our methodology involves cleaning both the HotelRec and Kuching datasets. This preprocessing step is uniformly applied across all recommendation algorithms in this study and is critical to ensure the datasets are well-structured and suitable for optimal algorithmic performance.

Missing values can significantly degrade the accuracy of recommendation algorithms. To address this issue, all missing values in both datasets were removed.

Duplicate entries can distort recommendation results by disproportionately weighting certain users or items. To mitigate this, all duplicate entries were removed from both datasets. This ensures a more balanced and reliable dataset for training and evaluation.

For the Kuching dataset, only the user ID, attraction ID, and user rating features were utilised for the recommendation task. Other textual features were reserved for the system's query analysis component.

For the HotelRec dataset, only the hotel_url, author, and rating features were used for the recommendation task. The date feature was excluded as it was deemed irrelevant to this study's recommendation goal. Textual features, such as titles and text (user reviews), were omitted due to their computational complexity and associated costs. Additionally, the property_dict feature was excluded because its limited usage by users introduced excessive sparsity, which could negatively impact the recommendation task.

4.5 Model training details

In this section, we provide the specific training procedures and implementation details for all algorithms employed in our comparative study, including our proposed ResNetMF.

For traditional collaborative filtering algorithms, namely Matrix Factorisation-based methods (SVD, SVD++, NMF, NeuMF) and KNN-based models (KNNBaseline, KNNBasic, KNN with Means, KNN with Z-Score, SlopeOne, BaselineOnly, NormalPredictor), we utilised the Surprise library. This library provides well-optimised and validated implementations, which allowed us to ensure consistency and robustness in their training processes, using the default or optimised parameters recommended by the library (Hug 2020).

For deep learning-based recommendation systems, including Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Neural Collaborative Filtering (NCF), Autoencoders, Variational Autoencoders (VAE), Graph Neural Network-based models (LightGCN, NGCF), Hybrid Deep Learning Models (DeepCoNN), and our proposed ResNetMF, all models were implemented using PyTorch. This framework was chosen for its flexibility, comprehensive capabilities for constructing complex neural networks, and efficient GPU acceleration, which was critical for expediting training and optimisation processes on our large datasets.

General Training Approach for Deep Learning Models (excluding ResNetMF specifics): All deep learning models, unless unique to ResNetMF's architecture, were trained using backpropagation with the Adam optimiser. A consistent initial learning rate of 0.001 was typically used, with adjustments made during hyperparameter tuning as needed to optimise performance. Training was performed using mini-batches to enhance computational efficiency and effectively handle larger datasets. We consistently employed early stopping based on the validation loss (e.g., RMSE on a held-out validation set) to prevent overfitting and identify the optimal training epoch. The model state achieving the best performance on the validation set was saved as the final trained model for evaluation.

Data Splitting Protocol: A consistent data splitting protocol was adhered to for all algorithms across both datasets. The datasets were rigorously partitioned into training (80%) and testing (20%) sets. This strict division ensured that there was no overlap between the data used for model training and the data used for final performance evaluation, thereby facilitating a fair comparison across all algorithms and significantly mitigating the risk of overfitting to unseen data.

4.5.1 ResNetMF training process

The training process for our proposed ResNetMF follows a structured, iterative approach designed for optimal performance and convergence:

1. Initialisation: The user and item embedding matrices (U and I) are initialised randomly, with a predefined dimensionality d (e.g., d = 64, corresponding to 64 latent factors). The weights of the ResNet branch, being a deep neural network, are initialised using the He initialisation method to promote stable training and prevent vanishing/exploding gradients in deeper layers.
2. Forward Pass: During each training iteration, the model performs a forward pass consisting of two main streams:

   - The MF branch computes an initial user-item interaction score as the dot product of the respective user and item embeddings (U × I).
   - The ResNet branch processes the input features (derived from user-item pairs, denoted as X) through its deep residual network, which incorporates multiple layers with skip connections. The output of this branch is the residual (F(X)), explicitly representing the error or uncaptured patterns by the MF model.
   - A merging layer then combines the outputs of the MF and ResNet branches, typically using summation or concatenation, to produce the final predicted recommendation score (R).

3. Loss Calculation: The primary objective of the model's training is to minimise the difference between its predicted scores (R) and the actual user-item interactions (Y). This is achieved by employing the Mean Squared Error (MSE) loss function. Optionally, an L2 regularisation term is added to the loss function to penalise large weights and further prevent overfitting.

4. Backpropagation: To update the model's parameters (embeddings and network weights), the gradients of the loss function with respect to all model parameters are computed using automatic differentiation. These gradients are then propagated backwards through both the MF and ResNet branches. An optimisation algorithm, specifically Stochastic Gradient Descent (SGD), is used to update the parameters, with a learning rate that can be dynamically adjusted during training based on a predefined schedule or adaptive methods.
5. Training Data and Batching: The entire training dataset is processed in mini-batches. This batching strategy enhances computational efficiency, particularly for large datasets, and helps stabilise the gradient updates.
6. Convergence Criteria: Training continues iteratively over epochs until a predefined convergence criterion is met. Typically, this involves early stopping, where training is halted if the model's performance on a separate validation set ceases to improve for a specified number of epochs.
7. Model Checkpointing: To ensure that the best-performing model is retained, model checkpointing is employed. The model's parameters that yield the optimal performance on the validation set are saved during training.

The next section discusses the experiment results for each dataset.

5 Results and discussions

In this section, we present a comprehensive evaluation of our proposed method, ResNetMF, against various baseline algorithms. The experiments are designed to address the core research questions posed in this study. Specifically, Experiment 1 (Sect. 5.1) evaluates ResNetMF's recommendation accuracy and efficiency (addressing RQ1) on the large-scale HotelRec dataset. Experiment 2 (Sect. 5.2) further assesses ResNetMF's performance on the multi-criteria Kuching dataset and, critically, evaluates the effectiveness of the query analysis component in enhancing user experience (addressing RQ2). We compare the efficiency and effectiveness of ResNetMF on selected datasets and analyse the results in detail. Furthermore, we present and analyse the experimental results of the query analysis process, directly linking these findings to the objectives outlined in our research questions.

5.1 Results of experiment 1

This experiment evaluates the effectiveness of ResNetMF on the HotelRec dataset. Its performance is compared against various deep learning methods, graph-based approaches, and traditional methods.

The experiment evaluates the performance of various recommendation algorithms across multiple metrics, including RMSE, MAE, Precision, Recall, F1-Score, NDCG, Hit Rate, and Training Time. Table 3 is an in-depth analysis of the results, highlighting the strengths and weaknesses of each algorithm and identifying the best-performing algorithm. It summarises each algorithm's result for each evaluation met-

Table 3 Result of experiment 1 across all evaluation metrics

Algorithm	RMSE	MAE	Precision	Recall	F1-score	NDCG	HR	Training time
SVD	0.9877	0.74	0.8379	0.607	0.704	0.51	0.6295	1036.85
SVD++	1.0073	0.7649	0.837	0.7466	0.7892	0.5595	0.612	25,464
NMF	1.0806	0.8491	0.8234	0.655	0.7298	0.457	0.6027	2474.05
KNNbaseline	0.9561	0.70602	0.7378	0.56	0.6367	0.529	0.576	1454.15
KNNbasic	1.0582	0.8163	0.803	0.588	0.6789	0.477	0.5989	1443.3
KNN with means	1.0524	0.812038	0.791	0.606	0.6863	0.499	0.5983	1490.25
KNNwithzscore	1.0526	0.8121	0.8014	0.528	0.6366	0.536	0.5985	1582.08
SlopeOne	1.069	0.8216	0.817	0.582	0.6798	0.417	0.5938	1774.36
BaselineOnly	0.9515	0.7128	0.735	0.562	0.637	0.355	0.6814	1248.85
NormalPredictor	1.3692	1.0352	0.6901	0.515	0.5899	0.377	0.4084	1484.4
Co-clustering	1.0765	0.8334	0.608	0.652	0.6292	0.4514	0.4851	3986.9
CNN	0.9648	0.7125	0.5689	0.4612	0.4575	0.9924	0.4612	501,130
RNN	1.0585	0.7897	0.9269	**0.9513**	**0.9389**	0.934	0.4281	430,875
NCF	0.9599	0.7929	**1**	0.3427	0.5105	**1**	**1**	126,030
CMF	0.9645	0.7527	0.8407	0.0288	0.0557	0.9115	0.4906	273,860
Autoencoders	4.2818	4.1278	0.996	0.06539	0.1227	0.99	**1**	79,810
NeuMF	1.0189	0.7923	0.8501	0.8777	0.8637	**1**	**1**	11757.8
Variational Autoencoder (VAE)	3.9231	3.7772	0.75	0.5	0.6	0.29	0.2266	4455
LightGCN	4.3334	4.1988	0.8	0.641	0.7117	0.1148	0.1977	902.5
NGCF	3.837	3.687	0.0209	0.0209	0.0209	0.9306	**1**	631
DeepCoNN	0.969	0.7427	0.8434	0.9332	0.8861	**1**	0.7504	577,210
ResNetMF	**0.8707**	**0.6943**	0.8975	0.7147	0.7957	0.9115	0.8156	**585**

ric. Precision, recall, F1-Score, NDGC and Hit rate (HR) are all calculated for the top 10 recommendations (e.g., Recall@10).

5.1.1 Detailed algorithm analysis on HotelRec dataset

This section presents a detailed analysis of the algorithms' performance on the HotelRec dataset, with comprehensive results summarised in Table 3. Our evaluation considers both prediction accuracy (RMSE, MAE) and ranking performance (Precision, Recall, F1-Score, NDCG, Hit Rate), as well as computational efficiency (Training Time).

General performance trends Across the diverse set of algorithms, several key trends emerged. Traditional Matrix Factorisation methods (SVD, SVD++, NMF) generally demonstrated reasonable predictive accuracy (RMSE, MAE) and moderate efficiency. SVD++, while showing slight improvements in some ranking metrics, incurred a significantly higher training time compared to standard SVD. KNN-based algorithms, particularly KNNBaseline, exhibited competitive RMSE and MAE values, highlighting their robust performance among neighbourhood-based approaches, with generally reasonable training times. Simple baselines, such as BaselineOnly, also showed surprisingly good accuracy metrics for their simplicity, although their ranking performance was often limited. As expected, NormalPredictor consistently performed poorly across all metrics, serving as a lower bound for effectiveness.

Deep Learning-based algorithms presented a varied landscape. While some, such as NCF and NeuMF, achieved very high or perfect scores in specific ranking metrics (e.g., NDCG, Hit Rate), they often exhibited trade-offs, including lower Recall or significantly higher training times (e.g., CNN, RNN, DeepCoNN). Conversely, Autoencoders and Variational Autoencoders (VAE) demonstrated high precision and Hit Rates but suffered from substantially higher RMSE and MAE, indicating poor predictive accuracy in the direct rating prediction task. Graph-based models, such as LightGCN and NGCF, also showed mixed results, with some achieving high ranking scores but struggling with accuracy metrics, alongside generally efficient training.

ResNetMF's Overall Superiority: Our proposed ResNetMF consistently outperformed all other algorithms on the HotelRec dataset. It achieved the lowest RMSE (0.8707) and MAE (0.6943) among all evaluated models, indicating superior accuracy in predicting user preferences. Furthermore, ResNetMF demonstrated a strong and balanced performance across all ranking and classification metrics (e.g., NDCG: 0.9115, Hit Rate: 0.8156, Precision: 0.8975, Recall: 0.7147, F1-Score: 0.7957), outperforming most baselines. Importantly, while offering this superior performance, its training time (585 s) remained highly competitive, especially when considering the vastly higher computational costs of many deep learning models, such as CNN, RNN, and DeepCoNN.

Quantitative Improvement Analysis: To provide a clearer quantitative understanding of ResNetMF's advantages, Tables 4 and 5 present the percentage improvement (or decline) of other algorithms compared to ResNetMF across all metrics. As illustrated by the example in Table 6 (RNN vs. ResNetMF), positive percentages for RMSE and MAE indicate that the baseline algorithm performed worse than ResNetMF, while negative percentages for ranking metrics (Precision, Recall, F1-Score, NDCG, Hit Rate) indicate that ResNetMF performed better (as higher is better for these metrics). These tables unequivocally demonstrate ResNetMF's consistent and significant improvements over a wide range of state-of-the-art and traditional methods in both accuracy and efficiency, validating its design for large-scale tourism datasets (Table 7).

Discussion of Unpaired Comparisons: It is essential to note that, due to computational constraints, we conducted only one run per algorithm. Consequently, traditional paired statistical tests (e.g., paired t-tests or Wilcoxon signed-rank tests), which require multiple observations for each algorithm, could not be directly performed. However, the consistent and substantial performance gains observed across multiple metrics, as detailed in Tables 3 and 4, and 5, provide strong empirical evidence for ResNetMF's superiority when compared against each baseline's single reported value.

5.1.2 Findings

The results highlight a significant trade-off between predictive accuracy and computational efficiency. Traditional algorithms, such as SVD and KNNBaseline, achieve

Table 4 Percentage improvement in RMSE and MAE compared to ResNetMF

Algorithm	RMSE (%)	MAE (%)
SVD	13.44	6.58
SVD++ (SVDpp)	15.69	10.17
Non-Negative Matrix Factorisation (NMF)	24.1	22.3
KNNbaseline	9.81	1.69
KNNbasic	21.53	17.58
KNN with means	20.87	16.96
KNNwithzscore	20.89	16.97
SlopeOne	22.78	18.34
BaselineOnly	9.28	2.68
NormalPredictor	57.25	49.11
Co-clustering	23.63	20.04
CNN	10.8	2.62
RNN	21.57	13.74
NCF	10.25	14.2
CMF	10.77	8.41
Autoencoders	391.68	494.47
(NeuMF)	17.02	14.12
Variational Autoencoder (VAE)	350.52	444.04
LightGCN (LightGCN)	397.68	504.82
NGCF (Neural Graph Collaborative Filtering)	340.63	431.01
DeepCoNN (Deep Cooperative Neural Networks)	11.28	6.97

competitive accuracy metrics (e.g., RMSE: 0.9877 for SVD, RMSE: 0.956158 for KNNBaseline) with reasonable training times (1036.85 s for SVD and 1454.15 s for KNNBaseline). In contrast, deep learning models like CNN, RNN, and DeepCoNN achieve superior performance in specific metrics (e.g., Recall: 0.9513 for RNN, NDCG: 1 for NCF) but incur prohibitively high computational costs (e.g., 430875 s for RNN, 577210 s for DeepCoNN).

Several deep learning models, including Autoencoders and Variational Autoencoders (VAE), exhibit signs of overfitting. While these models achieve near-perfect scores in certain metrics (e.g., Precision: 0.996 for Autoencoders, NDCG: 1 for NCF), they perform poorly in accuracy metrics (e.g., RMSE: 4.2818 for Autoencoders, RMSE: 3.9231 for VAE). This suggests that while deep learning models can excel in specific tasks, their generalisation capabilities may be limited, raising concerns about their applicability in real-world scenarios.

The observed discrepancies between error-based metrics (MAE, RMSE) and other performance metrics (Precision, NDCG) can be attributed to several algorithmic and task-specific factors inherent to the models evaluated. Precision and NDCG are ranking-focused metrics that measure the model's ability to correctly identify or rank relevant items (e.g., top-k recommendations). These metrics are less sensitive to absolute error magnitudes and more focused on relative orderings. In contrast, RMSE/MAE penalise deviations in predicted values directly, making them stricter measures of regression accuracy. For instance, an Autoencoder may achieve perfect Precision by correctly ranking a few high-confidence predictions while still incurring large errors (high RMSE) for less frequent or noisy data points. Metrics like Precision/NDCG

Table 5 Percentage improvement in other evaluation metrics compared to ResNetMF

Algorithm	Precision (%)	Recall (%)	F1-score (%)	NDCG (%)	HR (%)
SVD	-6.64	-15.08	-11.51	-44.06	-22.82
SVD++	-6.74	4.46	-0.82	-38.63	-24.98
NMF	-8.26	-8.35	-8.28	-49.87	-26.12
KNNbaseline	-17.8	-21.64	-19.98	-41.97	-29.4
KNNbasic	-10.53	-17.73	-14.68	-47.68	-26.57
KNN with means	-11.87	-15.21	-13.75	-45.26	-26.64
KNNwithzscore	-10.71	-26.12	-19.99	-41.21	-26.62
SlopeOne	-8.97	-18.57	-14.57	-54.25	-27.2
BaselineOnly	-18.11	-21.37	-19.94	-61.07	-16.45
NormalPredictor	-23.11	-27.94	-25.86	-58.66	-49.94
Co-clustering	-32.26	-8.77	-20.93	-50.49	-40.53
CNN	-36.61	-35.47	-42.5	8.87	-43.47
RNN	3.28	33.1	18	2.47	-47.53
NCF	11.42	-52.05	-35.84	9.71	22.61
CMF	-6.32	-95.96	-92.99	0	-39.85
Autoencoders	10.98	-90.85	-84.58	8.61	22.61
NeuMF	-5.28	22.81	8.55	9.71	22.61
Variational Autoencoder	-16.43	-30.04	-24.59	-68.19	-72.23
LightGCN	-10.86	-10.31	-10.55	-87.41	-75.77
NGCF	-97.67	-97.67	-97.67	2.1	22.61
DeepCoNN	-6.03	30.57	11.36	9.71	-8

Table 6 Example: RNN vs. ResNetMF

Metric	RNN	ResNetMF	%Difference (Improvement/Decline)
RMSE	1.0585	0.8707	21.58% (RNN is worse)
MAE	0.7897	0.6943	13.76% (RNN is worse)
Precision	0.9269	0.8975	+3.28% (RNN is better)
Recall	0.9513	0.7147	+33.12% (RNN is better)
F1-Score	0.9389	0.7957	+17.99% (RNN is better)
NDCG	0.934	0.9115	+2.47% (RNN is better)
Hit Rate	0.4281	0.8156	-47.52% (RNN is worse)

are skewed toward dense or high-confidence interactions, while the entire data distribution influences RMSE/MAE. For example, in recommendation systems, a few popular items may dominate rankings (yielding high NDCG), but long-tail items with sparse interactions degrade RMSE.

The experiment reveals a divergence in performance between ranking and classification metrics across algorithms. For instance, NCF achieves perfect Precision (1) and NDCG (1) but suffers from low Recall (0.3427) and F1-Score (0.5105). Conversely, RNN performs well in classification metrics (Precision: 0.9269, Recall: 0.9513, F1-Score: 0.9389) but has higher error rates (RMSE: 1.0585, MAE: 0.7897).

Table 7 Experimentation result of Kuching dataset on different evaluation metrics

Algorithm	RMSE	MAE	Precision	Recall	F1-Score	Hitrate	NDCG
SVD	1.369	1.145	0.5731	0.641	0.683	0.677	0.47
SVD++	1.336	1.082	0.5741	0.639	0.685	0.458	0.48
NMF	1.433	1.136	0.5734	0.574	0.644	0.6321	0.365
KNNbaseline	1.377	1.159	0.4701	0.64	0.669	0.56	0.475
KNNbasic	1.468	1.277	0.4725	0.716	0.72	0.381	0.3252
KNN with means	1.371	1.127	0.471	0.624	0.664	0.561	0.477
KNNwithzscore	1.344	1.093	0.4705	0.635	0.668	0.61	0.543
SlopeOne	1.413	1.179	0.4681	0.666	0.673	0.267	0.257
BaselineOnly	1.474	1.286	0.3624	0.659	0.641	0.192	0.1633
NormalPredictor	2.5	2.113	0.3998	0.525	0.592	0.388	0.422
Co-clustering	1.391	1.12	0.4696	0.645	0.669	0.203	0.152
CNN	1.46	1.08	0.618	0.701	0.656	0.185	0.22
RNN	1.501	1.17	0.581	0.699	0.634	0.418	0.518
NCF	1.351	0.911	0.7021	0.722	0.712	0.2388	0.354
CMF	1.4637	1.039	0.674	0.709	0.691	0.451	0.51
Autoencoders	**0.981**	**0.61**	0.821	**0.8001**	**0.81**	0.497	0.5101
NeuMF	1.39	1.0016	**0.8282**	0.5714	0.6763	0.2143	0.1954
Variational Autoencoder	2.4128	2.0655	0.0039	0.1667	0.0076	**1**	0.8335
LightGCN	2.6478	2.0353	0.1048	0.2091	0.127	0.3699	0.8867
NGCF	1.717	1.356	0.385	0.272	0.21	0.542	0.948
DeepCoNN	1.509	1.229	0.329	0.27	0.202	0.489	**0.966**
ResNetMF	1.2991	0.7172	0.793	0.742	0.766	0.791	0.8017

This divergence underscores the importance of selecting evaluation metrics that align with the specific goals of the recommendation system.

Graph-based models such as LightGCN and NGCF demonstrate notable computational efficiency, with training times of 902.5 s and 631 s, respectively. However, their performance in accuracy metrics is suboptimal (e.g., RMSE: 4.3334 for LightGCN, RMSE: 3.837 for NGCF). This finding suggests that while graph-based models are computationally efficient, further research is needed to improve their predictive accuracy.

The ResNetMF (Proposed Method) emerges as the most robust algorithm, demonstrating state-of-the-art performance across multiple evaluation metrics. It achieves the lowest RMSE (0.8707) and lowest MAE (0.6943), indicating exceptional predictive accuracy. Furthermore, it excels in ranking metrics, with an NDCG of 0.9115 and a Hit Rate of 0.8156, as well as classification metrics, with a Precision of 0.8975, a Recall of 0.7147, and an F1-Score of 0.7957. Its moderate training time (585 s) further underscores its practicality for real-world applications.

5.2 Experiment 2 analysis and query analysis evaluation

Experiment 2 explores ResNetMF's performance on the Kuching dataset. The training time is not reported for this dataset, as all algorithms have nearly the same training time, given that this dataset is smaller than the HotelRec dataset.

Autoencoders and ResNetMF consistently outperform other algorithms in most metrics. Autoencoders achieve the lowest RMSE (0.981) and MAE (0.61), indicating high prediction accuracy. Autoencoders also have the highest precision (0.821), recall (0.8001), and F1-score (0.81), showing strong relevance and coverage. ResNetMF closely follows competitors in terms of RMSE (1.2991), MAE (0.7172), and F1-score (0.766), while achieving the highest hitrate (0.791) and NGDC (0.8017).

SVD and SVD++perform well in terms of RMSE, MAE, and F1-score, but deep learning-based methods, such as Autoencoders and ResNetMF, outperform them. NMF and KNN-based methods exhibit moderate performance but lag in terms of precision, recall, and F1-score.

Between deep learning methods, Autoencoders and ResNetMF dominate in most metrics, showcasing the power of deep learning for recommendation tasks. NCF (Neural Collaborative Filtering) and CMF (Collective Matrix Factorisation) also perform well but are not as strong as Autoencoders or ResNetMF. NeuMF (Neural Matrix Factorisation) achieves high precision (0.8282) but lower recall and F1-score, indicating a trade-off between relevance and coverage.

Graph-based methods, such as NGCF and LightGCN, perform poorly in terms of precision, recall, and F1-score, but achieve high NGDC, indicating good ranking quality despite low relevance. DeepCoNN performs similarly, with low precision and recall but high NGDC.

Weak Performers are Variational Autoencoder (VAE) and NormalPredictor, which perform poorly across most metrics, with high RMSE and MAE and very low precision, recall, and F1-score. SlopeOne and BaselineOnly also exhibit weak performance, particularly in terms of hitrate and NGDC.

It is observed that Deep learning-based methods (Autoencoders, ResNetMF, NCF) generally outperform traditional methods (SVD, KNN, NMF) in terms of RMSE, MAE, precision, recall, and F1-score. While graph-based methods (NGCF, LightGCN) excel in ranking quality (NGDC), they struggle with precision and recall, suggesting room for improvement. ResNetMF and Autoencoders lead in hitrate and NGDC, indicating they are effective at recommending relevant items to a large fraction of users.

5.2.1 Evaluating query analysis: challenges and reported user study

Evaluating the query analysis component, a crucial aspect of the system, presents a distinct challenge. User satisfaction with recommendations is subjective and influenced by individual cognitive processes. Unlike established accuracy metrics like RMSE, there is no single metric to definitively assess query analysis performance.

We conducted a user study to address this challenge. The study involved 50 random users who participated in the Kuching dataset data collection. Each user was asked to provide at least five queries and then rate their satisfaction with the recommendation results based on their query and perceived relevance of the recommendations on a scale of 1 to 10 (1: low, 10: high). The queries can be anything that the user wants to experience, such as "Jungle tracking with family.

The average user rating of 7.25 (on a scale of 0 to 10) suggests a generally positive response to the system's results. For illustrative purposes, this rating can be interpreted as approximately 72.5% of the maximum possible score, which may reflect a tendency toward user satisfaction. However, it is essential to note that this percentage is an approximation and not a direct calculation of the proportion of satisfied users. On the other hand, it means that the 72.5% score does not necessarily imply that 72.5% of users were satisfied. It is just a way to express the average rating relative to the maximum possible score.

The results indicate that the query analysis and information selection processes performed well, aligning with user preferences and resulting in overall positive feedback. This suggests that the system effectively refined recommendations based on user queries, thereby enhancing their relevance and usefulness.

To further strengthen these findings, future research will incorporate a comparative analysis to assess the performance of recommendations before and after query analysis. This will provide a clearer understanding of its role in shaping user perceptions and enhancing recommendation effectiveness.

5.3 Discussion

The findings from the experiments and user study offer valuable insights into the effectiveness of the proposed recommendation system and its query analysis mechanism. Here, we discuss these findings in the context of existing research and address the limitations of the study.

5.3.1 Effectiveness of ResNetMF

The results demonstrate that ResNetMF achieves superior recommendation accuracy compared to existing deep learning algorithms, such as NCF, on the HotelRec dataset (refer to Tables 3 and 5). This aligns with previous research highlighting the effectiveness of matrix factorisation techniques combined with deep learning for recommendation systems (De Pauw 2020). Notably, ResNetMF maintains this advantage across various datasets, as demonstrated by its competitive performance on the Kuching city attraction dataset.

The user study further supports the system's efficacy by indicating positive user satisfaction, as evidenced by an **average rating of 7.25 for the recommendations**. This finding suggests that ResNetMF, complemented by the query analysis mechanism, effectively addresses information overload for tourists and successfully aligns with user preferences. While the user study provides valuable insights into user perception, it is important to acknowledge the inherent subjectivity of user satisfaction and the general limitations of such studies.

5.3.2 Theoretical and practical implications

The success of ResNetMF demonstrates the potential of combining matrix factorisation with deep learning architectures for building robust recommendation systems. This finding contributes to the ongoing research in recommender system development by offering a solution that balances accuracy and efficiency.

From a practical standpoint, the proposed system, incorporating ResNetMF, offers a promising approach for tourism recommendation systems. Addressing information overload and providing user-centric recommendations through the query analysis mechanism can significantly improve tourist experiences. Future research can further enhance the system by incorporating these considerations:

Mitigating Cold Start Problem: The proposed knowledge graphs for users and tourist attractions offer a promising approach to address the Cold Start problem (difficulty in recommending new items/users). Future research can investigate the effectiveness of these knowledge graphs in improving recommendation accuracy for new users and items.

Advanced Query Analysis: Integrating newer NLP techniques, such as large language model (LLM) models like GPT-3 transformers or other LLM models, including GPT (Generative Pre-trained Transformer) and BERT (Bidirectional Encoder Representations from Transformers), to process and analyse language for query analysis holds promise for further improvement. Exploring these advancements and evaluating their impact on recommendation accuracy and user satisfaction can be a valuable avenue for future research.

6 Conclusion, limitations and future work

This study aimed to improve the accuracy of tourism recommendation systems by proposing a multi-criteria approach. The proposed method, combining ResNetMF (a deep learning recommendation engine) with a query analysis mechanism, demonstrates promising results. The key findings are:

1 Improved Recommendation Accuracy: ResNetMF achieves higher scores in most evaluation metrics compared to existing deep learning and baseline algorithms in predicting user ratings, as evidenced by lower Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). It also achieves reasonably high scores on other metrics, such as Precision and Hit Rate, across both datasets.
2 Faster Training Speeds: While achieving high accuracy, ResNetMF maintains faster training times compared to other methods, making it suitable for real-world applications with time constraints.
3 Positive User Experience: The user study, despite limitations in sample size (50 users) and data source (focusing on Kuching City attractions), yielded a positive average user satisfaction rating (7.25), indicating that the system effectively addressed information overload for tourists in Kuching.

Overall, this study presents a user-centred recommendation system with promising potential to enhance tourist experiences. While acknowledging the limitations of the current evaluation, this work lays the groundwork for developing even more effective and informative recommendation systems for travellers.

6.1 Limitations and future work

This study, while demonstrating significant contributions, is subject to several limitations that warrant consideration and offer avenues for future research. First, the dataset Suitability for Multi-Criteria Evaluation. While the large-scale HotelRec dataset was crucial for evaluating the core recommendation engine's accuracy and scalability in overall rating prediction, its extreme sparsity in multi-criteria sub-ratings made it unsuitable for a comprehensive multi-criteria recommendation system (MCRS) task. Consequently, the multi-criteria aspects of our system were primarily evaluated using the Kuching dataset, which, while smaller, provided more suitable multi-dimensional user feedback for this specific purpose.

Also, this study acknowledges that the Kuching dataset and testing query analysis on the small population of Kuching city are limitations, and the result cannot be generalised. For future work, we aim to test this system in a broader geographical area with a larger user base. Moreover, this study tests and compares all the algorithms in a single run due to computational constraints. In future, we intend to test and run algorithms for the same datasets for several runs and apply proper statistical tests.

It is also worth noting that while ResNetMF achieves high accuracy, some deep learning methods, such as RNN, might offer equal or better accuracy at the cost of significantly slower training times. Finding the optimal balance between accuracy and training speed is crucial for real-world applications with time constraints.

This study lays the groundwork for further development of the proposed multi-criteria recommendation system. Future research should involve expanding the user study with a larger and more diverse user group. Additionally, exploring advanced query analysis methods, such as GPT-3 transformers, holds promise for potentially improving both recommendation accuracy and user experience. Finally, mitigating the cold start problem remains crucial for real-world applications. Developing knowledge graphs for users and tourist attractions offers a promising approach to improve recommendation accuracy for new users and items encountered by the system. By pursuing these future research directions, we can contribute to the development of even more effective and user-centric tourism recommendation systems.

By addressing these limitations and pursuing future research directions, this work can contribute to the development of even more effective and user-centric tourism recommendation systems.

Funding No funding was received for conducting this study.

Declarations

Competing interests The authors declare that they have no competing interests.

References

Adomavicius G, Kwon Y (2007) New recommendation techniques for multicriteria rating systems. IEEE Intell Syst 22(3):48–55. https://doi.org/10.1109/MIS.2007.58

Alamdari PM, Navimipour NJ, Hosseinzadeh M, Safaei AA, Darwesh A (2020) A systematic study on the recommender systems in the E-Commerce. IEEE Access 8:115694–115716. https://doi.org/10.1109/ACCESS.2020.3002803

Antognini D, Faltings B (2020) HotelRec: A novel very large-scale hotel recommendation dataset. Proceedings of the Twelfth Language Resources and Evaluation Conference. The 12th Language Resources and Evaluation Conference, Marseille

Asaithambi SPR, Venkatraman R, Venkatraman S (2023) A thematic travel recommendation system using an augmented big data analytical model. Technol 11(1):28. https://doi.org/10.3390/technologies11010028.

Aysha S, Tarun S (2022) A pareto dominance approach to multi-criteria recommender system using PSO algorithm. In Khanna A., Gupta D., Bhattacharyya S., Hassanien A. E., Anand S., & Jaiswal A. (Eds.), International Conference on Innovative Computing and Communications (Vol. 1387, pp. 737–755). Springer Singapore. https://doi.org/10.1007/978-981-16-2594-7_60

Baker O, Yuan Q (2021) Machine learning: factorization machines and normalized discounted cumulative gain for tourism recommender system optimisation. 2021 IEEE Int Conf Comput (ICOCO) 31–36. https://doi.org/10.1109/ICOCO53166.2021.9673502

Batmaz Z, Kaleli C (2019) AE-MCCF: an Autoencoder-Based Multi-criteria recommendation algorithm. Arab J Sci Eng 44(11):9235–9247. https://doi.org/10.1007/s13369-019-03946-z

Behera G, Nain N (2023) The State-of-the-Art and challenges on recommendation system's: Principle, techniques and evaluation strategy. SN Comput Sci 4(5):677. https://doi.org/10.1007/s42979-023-02207-z

Ben Sassi I, Yahia B, S., Liiv I (2021) MORec: at the crossroads of context-aware and multi-criteria decision making for online music recommendation. Expert Syst Appl 183:115375. https://doi.org/10.1016/j.eswa.2021.115375

De Pauw J (2020) Exploratory methods for evaluating recommender systems. Fourteenth ACM Conference on Recommender Systems, 782–786. https://doi.org/10.1145/3383313.3411456

Deshpande M, Karypis G (2004) Item-based top- N recommendation algorithms. ACM Trans Inform Syst 22(1):143–177. https://doi.org/10.1145/963770.963776

Dridi R, Tamine L, Slimani Y (2022) Exploiting context-awareness and multi-criteria decision making to improve items recommendation using a tripartite graph-based model. Inf Process Manag 59(2):102861. https://doi.org/10.1016/j.ipm.2021.102861.

Elahi E, Iglesias A, Morato J (2022) Readability of Graphical Contents on World Wide Web (WWW). 2022 17th Iberian Conference on Information Systems and Technologies (CISTI), 1–4. https://doi.org/10.23919/CISTI54924.2022.9820011

Farooqi RA, Kesarwani S, Shakeeb M, Sharma N, Ishita Bhatnagar (2022) Enhancing E-Commerce applications with machine learning recommendation systems. Int J Sci Res Sci Eng Technol 85–90. https://doi.org/10.32628/IJSRSET122935

Fraihat S, Shambour Q, Al-Betar MA, Makhadmeh SN (2024) Variational autoencoders-based algorithm for multi-criteria recommendation systems. Algorithms 17(12):561. https://doi.org/10.3390/a17120561.

George T, Merugu S (2005) A scalable collaborative filtering framework based on Co-Clustering. Fifth IEEE Int Conf Data Min (ICDM'05) 625–628. https://doi.org/10.1109/ICDM.2005.14

Gupta M, Goyal V, Rathi V (2024) Revolutionizing travel planning: An image based destination recognition and recommendation system. 2024 1st International Conference on Advanced Computing and Emerging Technologies (ACET), 1–6. https://doi.org/10.1109/ACET61898.2024.10730034

Hafez MM, Redondo RPD, Vilas AF, Pazó HO (2021) Multi-Criteria recommendation systems to foster online grocery. Sensors 21(11):3747. https://doi.org/10.3390/s21113747

Hassan M, Hamada M (2017) A neural networks approach for improving the accuracy of multi-criteria recommender systems. Appl Sci 7(9):868. https://doi.org/10.3390/app7090868

He X, Liao L, Zhang H, Nie L, Hu X, Chua T-S (2017) Neural collaborative filtering. Proceedings of the 26th International Conference on World Wide Web, 173–182. https://doi.org/10.1145/3038912.3052569

He X, Deng K, Wang X, Li Y, Zhang Y, Wang M (2020) LightGCN: simplifying and powering graph Convolution network for recommendation. Proc 43rd Int ACM SIGIR Conf Res Dev Inform Retr 639–648. https://doi.org/10.1145/3397271.3401063

Hodson TO (2022) Root-mean-square error (RMSE) or mean absolute error (MAE): when to use them or not. Geosci Model Dev 15(14):5481–5487. https://doi.org/10.5194/gmd-15-5481-2022

Hug N (2020) Surprise: A python library for recommender systems. J Open Source Softw 5(52):2174. https://doi.org/10.21105/joss.02174

Hussein AS, Omar WM, Li X, Ati M (2012) Efficient chronic disease diagnosis prediction and recommendation system. 2012 IEEE-EMBS Conf Biomedical Eng Sci 209–214. https://doi.org/10.1109/IECBES.2012.6498117

Ifada N, Sophan MK, Putri NFD (2023) A minmax Item-based method for Multi-Criteria recommendation systems. Procedia Comput Sci 227:1020–1029. https://doi.org/10.1016/j.procs.2023.10.611

Khalaji M, Mohammadnejad N (2019) CUPCF: combining users preferences in collaborative filtering for better recommendation. SN Appl Sci 1(9):1053. https://doi.org/10.1007/s42452-019-1071-6

Kim D, Park C, Oh J, Lee S, Yu H (2016) Convolutional matrix factorization for document Context-Aware recommendation. Proc 10th ACM Conf Recommender Syst 233–240. https://doi.org/10.1145/2959100.2959165

Koren Y (2008) Factorization Meets the neighborhood: A multifaceted collaborative filtering model. Proc 14th ACM SIGKDD Int Conf Knowl Discovery Data Min 426–434. https://doi.org/10.1145/1401890.1401944

Koren Y (2009) Collaborative filtering with Temporal dynamics. Proc 15th ACM SIGKDD Int Conf Knowl Discovery Data Min 447–456. https://doi.org/10.1145/1557019.1557072

Koren Y (2010) Factor in the neighbors: scalable and accurate collaborative filtering. ACM Trans Knowl Discovery Data 4(1):1–24. https://doi.org/10.1145/1644873.1644874

Krishna CVM, Rao GA, Anuradha S (2023) Analysing the impact of contextual segments on the overall rating in multi-criteria recommender systems. J Big Data 10(1):16. https://doi.org/10.1186/s40537-023-00690-y

Kumar N, Pallavi KV, Hanji BR (2023) Personalized travel recommendation system using average cumulative rating matrix factorization technique: concept and framework. Vietnam J Comput Sci 10(02):159–195. https://doi.org/10.1142/S2196888822500361

Lediga MM, Fombad MC (2018) The use of information and communication technologies in public libraries in South Africa as tools for bridging the digital divide: the case of the Kempton park public library. Public Libr Q 37(3):296–305. https://doi.org/10.1080/01616846.2018.1471964

Lee DD, Seung HS (1999) Learning the parts of objects by non-negative matrix factorization. Nature 401(6755):788–791. https://doi.org/10.1038/44565

Lemire D, Maclachlan A (2005) Slope one predictors for online rating-based collaborative filtering. Proceedings of the 2005 SIAM International Conference on Data Mining, 471–475. https://doi.org/10.1137/1.9781611972757.43

Lingo EL (2023) Digital curation and creative brokering: managing information overload in open organizing. Organ Stud 44(1):105–133. https://doi.org/10.1177/01708406221099697

Liu H (2022) Implementation and effectiveness evaluation of four common algorithms of recommendation Systems—User collaboration Filter, Item-based collaborative filtering, matrix factorization and neural collaborative filtering. 2022 Int Conf Cloud Comput Big Data Appl Softw Eng (CBASE) 224–227. https://doi.org/10.1109/CBASE57816.2022.00049

Mahmoud DS, John RI (2015) Enhanced content-based filtering algorithm using artificial bee colony optimisation. 2015 SAI Intelligent Systems Conference (IntelliSys), 155–163. https://doi.org/10.1109/IntelliSys.2015.7361139

Manikantan A (2021) A hybrid recommendation system for video games: combining Content-based & collaborative filtering. Int J Res Appl Sci Eng Technol 9(9):1647–1653. https://doi.org/10.22214/ijraset.2021.38246

Meehan K, Lunney T, Curran K, McCaughey A (2016) Aggregating social media data with Temporal and environmental context for recommendation in a mobile tour guide system. J Hospitality Tourism Technol 7(3):281–299. https://doi.org/10.1108/JHTT-10-2014-0064

Nan X, Kayo Kanato, Wang X (2022) Design and implementation of a personalized tourism recommendation system based on the data mining and collaborative filtering algorithm. Computational Intelligence and Neuroscience, 2022, 1–14. https://doi.org/10.1155/2022/1424097

Nassar N, Jafar A, Rahhal Y (2020) Multi-criteria collaborative filtering recommender by fusing deep neural network and matrix factorization. J Big Data 7(1):34. https://doi.org/10.1186/s40537-020-00309-6

Park DH, Kim HK, Choi IY, Kim JK (2012) A literature review and classification of recommender systems research. Expert Syst Appl 39(11):10059–10072. https://doi.org/10.1016/j.eswa.2012.02.038

Praditya WPY, Permanasari NE, A., Hidayah I (2021) Designing a tourism recommendation system using a hybrid method (Collaborative Filtering and Content-Based Filtering). 2021 IEEE International Conference on Communication, Networks and Satellite (COMNETSAT), 298–305. https://doi.org/10.1109/COMNETSAT53002.2021.9530823

Resnick P, Iacovou N, Suchak M, Bergstrom P, Riedl J (1994) GroupLens: an open architecture for collaborative filtering of Netnews. Proc 1994 ACM Conf Comput Supported Coop Work - CSCW '94 175–186. https://doi.org/10.1145/192844.192905

Sahoo N, Krishnan R, Duncan G, Callan J (2012) Research note —The halo effect in multicomponent ratings and its implications for recommender systems: the case of Yahoo! Movies. Inform Syst Res 23(1):231–246. https://doi.org/10.1287/isre.1100.0336

Shambour Q (2021) A deep learning based algorithm for multi-criteria recommender systems. Knowl Based Syst 211:106545. https://doi.org/10.1016/j.knosys.2020.106545

Shambour QY, Abualhaj MM, Abu-Shareha AA (2023) Restaurant recommendations based on Multi-Criteria recommendation algorithm. JUCS - J Univers Comput Sci 29(2):179–200. https://doi.org/10.3897/jucs.78240

Shen L, Liu Q, Chen G, Ji S (2020) Text-based price recommendation system for online rental houses. Big Data Min Analytics 3(2):143–152. https://doi.org/10.26599/BDMA.2019.9020023

Song N (2020) Analysis of recommendation systems based on neural networks. J Phys: Conf Ser 1634(1):012051. https://doi.org/10.1088/1742-6596/1634/1/012051

Su X, Khoshgoftaar TM (2009) A survey of collaborative filtering techniques. Adv Artif Intell 2009:1–19. https://doi.org/10.1155/2009/421425

Suanpang P, Jamjuntr P, Lertkornkitja A, Jittithavorn C (2024) Can optimized genetic algorithms improve the effectiveness of homestay recommendation systems in smart villages? A case of Thailand. Int J Tourism Res 26(5):e2762. https://doi.org/10.1002/jtr.2762

Tran HQ, Ha C (2020) High precision weighted optimum K-Nearest neighbors algorithm for indoor visible light positioning applications. IEEE Access 8:114597–114607. https://doi.org/10.1109/ACCESS.2020.3003977

Wang X, He X, Wang M, Feng F, Chua T.-S. (2019) Neural graph collaborative filtering. Proc 42nd Int ACM SIGIR Conf Res Dev Inform Retr 165:174. https://doi.org/10.1145/3331184.3331267

Wang Z, Wang Z, Xu Y, Wang X, Tian H (2023) Online course recommendation algorithm based on multilevel fusion of user features and item features. Comput Appl Eng Educ 31(3):469–479. https://doi.org/10.1002/cae.22592

Wei X, Yu X, Liu B, Zhi L (2019) Convolutional neural networks and local binary patterns for hyperspectral image classification. Eur J Remote Sens 52(1):448–462. https://doi.org/10.1080/22797254.2019.1634980

Wen J, Liu P (2023) A classification method for english texts based on hybrid recurrent neural network and graph construction in social recommendation systems. IEEE Syst J 1–8. https://doi.org/10.1109/JSYST.2023.3283147

Wu L, Grbovic M (2020) How airbnb tells you will enjoy sunset sailing in Barcelona? Recommendation in a two-sided travel marketplace. Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2387–2396. https://doi.org/10.1145/3397271.3401444

Xiao T, Shen H (2019) Neural variational matrix factorization for collaborative filtering in recommendation systems. Appl Intell 49(10):3558–3569. https://doi.org/10.1007/s10489-019-01469-6

Xu C, Zhao P, Liu Y, Xu J, Sheng S, Cui VSS, Zhou Z, X., Xiong H (2019) Recurrent convolutional neural network for sequential recommendation. The World Wide Web Conference, 3398–3404. https://doi.org/10.1145/3308558.3313408

Yuan Y, Zhou Y, Chen X, Xiong Q, Okere HC (2024) Enhancing recommendation diversity and novelty with Bi-LSTM and mean shift clustering. Electronics 13(19):3841. https://doi.org/10.3390/electronics13193841

Zeeshan Z, Ul Ain Q, Bhatti UA, Memon WH, Ali S, Nawaz SA, Nizamani MM, Mehmood A, Bhatti MA, Shoukat MU (2021) Feature-based multi-criteria recommendation system using a weighted approach with ranking correlation. Intell Data Anal 25(4):1013–1029. https://doi.org/10.3233/IDA-205388

Zhang S, Yao L, Sun A, Tay Y (2020) Deep learning based recommender system: A survey and new perspectives. ACM-CSUR 52(1):1–38. https://doi.org/10.1145/3285029

Zhang K, Liu X, Wang W, Li J (2021) Multi-criteria recommender system based on social relationships and criteria preferences. Expert Syst Appl 176:114868. https://doi.org/10.1016/j.eswa.2021.114868

Zhang Y, Lu X, Shi Y, Zhang D (2023) Hybrid algorithm for item collaborative filtering based on matrix factorization. 2023 4th Inform Communication Technol Conf (ICTC) 276:284. https://doi.org/10.1109/ICTC57116.2023.10154673

Zheng Y (2017) Criteria chains: A novel multi-criteria recommendation approach. Proc 22nd Int Conf Intell User Interfaces 29–33. https://doi.org/10.1145/3025171.3025215

Zheng Y, Wang D (2022) Multi-Criteria ranking: next generation of Multi-Criteria recommendation framework. IEEE Access 10:90715–90725. https://doi.org/10.1109/ACCESS.2022.3201821

Zheng L, Noroozi V, Yu PS (2017) Joint deep modeling of users and items using reviews for recommendation. Proc Tenth ACM Int Conf Web Search Data Min 425:434. https://doi.org/10.1145/3018661.3018665

Zhu F, Jiang M, Qiu Y, Sun C, Wang M (2019) RSLIME: An efficient feature importance analysis approach for industrial recommendation systems. 2019 International Joint Conference on Neural Networks (IJCNN), 1–6. https://doi.org/10.1109/IJCNN.2019.8852034

Zhuang Y, Kim J (2021) A BERT-Based Multi-Criteria recommender system for hotel promotion management. Sustainability 13(14):8039. https://doi.org/10.3390/su13148039