



Faculty of Computer Science and Information Technology

**A Multi-Criteria Recommendation Technique for Personalized Tourism
Experiences**

Mustafa Payandenick

**Master of Science
2024**

A Multi-Criteria Recommendation Technique for Personalized Tourism Experiences

Mustafa Payandenick

A thesis submitted

In fulfillment of the requirements for the degree of Master of Science

(Software Engineering)

Faculty of Computer Science and Information Technology

UNIVERSITI MALAYSIA SARAWAK

2024

DECLARATION

I declare that the work in this thesis was carried out in accordance with the regulations of Universiti Malaysia Sarawak. Except where due acknowledgements have been made, the work is that of the author alone. The thesis has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

MustafaPNK
.....

Signature

Name: Mustafa Payandenick

Matric No.: 19020070

Faculty of Computer Science and Information Technology

Universiti Malaysia Sarawak

Date: 17 June 2024

ACKNOWLEDGEMENT

Firstly, I am grateful to the Almighty Allah for his abundant blessings and grace guiding me throughout this journey. I faced numerous challenges, and each taught me valuable lessons under Allah's divine guidance. I am immensely grateful for the strength and knowledge I gained during this journey.

This project required a lot of perseverance, and I'm proud of myself for not giving up on the challenges I faced. Looking back, I'm grateful for the resilience I developed while working on it. It definitely helped me hone my research and problem-solving skills, which will be valuable assets moving forward.

I express my heartfelt gratitude to my father for his unwavering love, care, and support throughout my studies. Additionally, I sincerely thank my brother Muhammad, who stood by me during this journey.

I am deeply grateful to my supervisor, Professor Wang Yin Chai, for his constant guidance and support throughout my research journey. His patience in explaining complex concepts and his willingness to help me troubleshoot challenges were invaluable. I also thank the Centre for Graduate Studies for their advice and support during my studies.

Finally, I thank the management of Universiti Malaysia Sarawak, especially Professor Ahmad Hata, for providing me with another opportunity to continue my studies.

ABSTRACT

This research aims to design a multi-Criteria recommendation technique for the tourism domain. In the past few decades, the growth of the World Wide Web has led to an unprecedented amount of information available to us. This phenomenon has resulted in what we now call information overload, where the sheer volume of data surpasses our capacity to manage it effectively. In order to address this issue, it is crucial to ensure that accurate information is communicated to the appropriate audience, as proper information dissemination is key. Like other industries, the tourism industry faces the challenge of information overload. The abundance of information can be overwhelming for both tourists and industry stakeholders. Some tourists like guided tours, while others prefer exploring independently, and that is where e-tour guides can be helpful. E-tour guides are digital tools like apps or websites that inform tourists about their destination. In this context, recommendation systems and information dissemination are closely related. Recommendation systems use algorithms to analyze user data and provide personalized recommendations based on their past behaviour and preferences. Both recommendation systems and information dissemination aim to provide users with relevant and useful information. This study aims to develop a multi-criteria recommendation system that effectively addresses the issue of information overload in the tourism industry by delivering pertinent information to the right users. The proposed method combines several techniques, including deep learning and traditional techniques, such as matrix factorization, to address common challenges like scalability and data sparseness. The approach is designed to provide personalized recommendations based on user location, site, and other relevant criteria. By emphasizing the filtering and provision of the most relevant information based on user location and site, the user experience and engagement can be significantly improved. This

study utilizes the Convolutional Matrix Factorization (CMF) algorithm due to its compatibility with tourism. The proposed algorithm, CMF with ResNet, combines the power of CMF with the superior performance of ResNet to overcome the limitations of CMF and achieve even better results for recommendation tasks. Using ResNet, the algorithm can learn more complex and nuanced patterns in the data, leading to more accurate recommendations. In the end, compared to all the tested algorithms, the proposed method outperformed and achieved higher scores on error measurement metrics. Additionally, it can handle complex problems like sparse data or very small amounts of training samples. The proposed method also provided more relevant recommendations to users compared to all other tested algorithms. Overall, the proposed method offers a novel and effective solution to the information overload challenge in the tourism industry.

Keywords: Intelligence information dissemination, information overload, multi-criteria recommendation system, convolutional matrix factorization (CMF), residual networks(ResNet)

Teknik Kepintaran Penyebaran Maklumat untuk Aplikasi E-Tour Guide

ABSTRAK

Penyelidikan ini bertujuan untuk mereka bentuk kepintaran penyebaran maklumat untuk domain pelancongan. Dalam beberapa dekad yang lalu, pertumbuhan World Wide Web telah membawa kepada jumlah maklumat yang belum pernah berlaku sebelum ini tersedia kepada kami. Fenomena ini telah mengakibatkan apa yang kini kita panggil sebagai maklumat berlebihan, di mana jumlah data yang banyak melebihi kapasiti kami untuk mengurusnya dengan berkesan. Untuk menangani isu ini, adalah penting untuk memastikan maklumat yang tepat disampaikan kepada pengguna yang sesuai, kerana penyebaran maklumat yang betul adalah penting. Seperti industri lain, industri pelancongan menghadapi cabaran lebih maklumat. Dengan peningkatan populariti melancong, jumlah maklumat yang tersedia boleh menjadi sangat menggalakkan untuk pelancong dan pihak berkepentingan industri. Walaupun sesetengah pelancong lebih suka lawatan berpandu, yang lain lebih suka meneroka destinasi mereka secara bebas, iaitu e-pemandu pelancong. E-pemandu pelancong ialah aplikasi, laman sesawang atau aplikasi yang menyediakan pelancong dengan maklumat tentang destinasi mereka. Dalam konteks ini, sistem pengesyoran dan penyebaran maklumat adalah berkait rapat. Sistem pengesyoran menggunakan algoritma untuk menganalisis data pengguna dan memberikan pengesyoran yang pribadi berdasarkan faktor seperti tingkah laku dan pilihan pada masa lalu mereka. Kedua-dua system pengesyoran dan penyebaran maklumat bertujuan untuk menyediakan pengguna dengan maklumat yang relevan dan berguna. Kajian ini bertujuan untuk membangunkan sistem pengesyoran pelbagai kriteria yang berkesan menangani isu lebih maklumat dalam industri pelancongan dengan menyampaikan maklumat yang berkaitan kepada pengguna yang tepat.

Kaedah yang dicadangkan menggabungkan beberapa teknik, termasuk pembelajaran mendalam dan teknik tradisional, seperti pemfaktoran matriks, untuk menangani cabaran biasa seperti kebolehskalaan dan ketersediaan data. Pendekatan ini direka bentuk untuk memberikan pengesyoran yang diperibadikan berdasarkan tempat lokasi pengguna, dan kriteria lain yang berkaitan. Dengan menekankan penapisan dan penyediaan maklumat yang paling relevan berdasarkan lokasi dan tempat, pengalaman dan penglibatan pengguna boleh dipertingkatkan dengan ketara. Kajian ini menggunakan algoritma Convolutional Matrix Factorization (CMF) kerana keserasiannya dengan domain pelancongan. Algoritma yang dicadangkan, CMF dengan ResNet, menggabungkan kuasa CMF dengan prestasi unggul ResNet untuk mengatasi batasan CMF dan mencapai hasil yang lebih baik untuk tugas pengesyoran. Menggunakan ResNet, algoritma boleh mempelajari corak yang lebih kompleks dan bernuansa dalam data, yang membawa kepada pengesyoran yang lebih tepat. Akhirnya, berbanding dengan semua algoritma yang telah diuji, kaedah yang dicadangkan mengatasi prestasi dan mencapai skor yang lebih tinggi pada metrik pengukuran ralat. Selain itu, ia boleh menangani masalah kompleks seperti data yang jarang atau jumlah sampel latihan yang sangat kecil. Kaedah yang dicadangkan juga memberikan pengesyoran yang lebih relevan kepada pengguna berbanding semua algoritma lain yang diuji. Secara keseluruhannya, kaedah yang dicadangkan menawarkan penyelesaian yang baru dan berkesan kepada cabaran lebihan maklumat dalam industri pelancongan.

Kata kunci: *Kepintaran penyebaran maklumat, sarat maklumat, sistem pengesyoran berbilang kriteria, pemfaktoran matriks konvolusi (CMF)*

TABLE OF CONTENTS

	Page
DECLARATION	i
ACKNOWLEDGEMENT	ii
ABSTRACT	iii
ABSTRAK	v
TABLE OF CONTENTS	vii
LIST OF TABLES	xi
LIST OF FIGURES	xiii
LIST OF ABBREVIATIONS	xiv
CHAPTER 1 INTRODUCTION	1
1.1 Research Background	1
1.2 Problem Statement	2
1.3 Objectives	4
1.4 Scope of Study	5
1.5 Thesis Structure	6
CHAPTER 2 LITERATURE REVIEW	8
2.1 Introduction	8
2.2 Information Dissemination	8
2.3 Recommendation Systems	13

2.3.1	Content-Based Filtering	14
2.3.2	Collaborative Filtering	15
2.3.3	Hybrid Filtering	19
2.4	Multi-Criteria Recommendation Systems	21
2.5	Tourism Multi-Criteria Recommendation System	26
2.6	Key Findings	31
2.7	Key Takeaways	32
2.8	Chapter Summary	36
	CHAPTER 3 METHODOLOGY AND DESIGN OF PROPOSED METHOD	37
3.1	Introduction	37
3.2	Methodology	37
3.3	System Architecture	42
3.4	Recommendation Engine (ResNetMF)	46
3.4.1	Notations	48
3.4.2	The Core Object of Study	49
3.4.3	Matrix Factorization	51
3.4.4	Residual Neural Network	55
3.4.5	Overall Mechanism of Proposed Recommendation Method (ResNetMF)	60
3.5	User Query Analysis	63
3.6	Conclusion	68

CHAPTER 4 EXPERIMENTAL DESIGN	69
4.1 Introduction	69
4.2 Experimental Design and Benchmarking	69
4.3 Datasets	72
4.3.1 Kuching Dataset	73
4.3.2 MovieLens Dataset	75
4.4 Evaluating ResNetMF with Diverse Recommendation Algorithms	77
4.4.1 Matrix Factorization (MF) Techniques	77
4.4.2 Collaborative Filtering (CF) Techniques	78
4.4.3 Content-Based (CBF) Techniques	78
4.4.4 Deep Learning Techniques	78
4.5 Evaluation Metrics	80
4.6 Pre-Experimental Phase	83
4.6.1 Data Preprocessing and Cleaning	83
4.6.2 Hyperparameter Tuning	84
4.7 Conclusion	86
CHAPTER 5 RESULTS AND ANALYSIS	87
5.1 Introduction	87
5.2 Results and Discussion	87
5.2.1 Experiment Number 1	87

5.2.2	Experiment Number 2	94
5.2.3	Key Findings from Recommendation Experiments	98
5.2.4	Dissemination Experiment (Query Analysis)	100
5.3	Contribution	102
5.4	Conclusion	104
CHAPTER 6 CONCLUSION AND FUTURE WORKS		105
6.1	Introduction	105
6.2	Achievements	106
6.3	Limitations	108
6.4	Research Contribution	109
6.5	Future Works	110
6.6	Summary	112
REFERENCES		113

LIST OF TABLES

	Page
Table 2.1 Basic Collaborative Sample	15
Table 2.2 Solved By Collaborative Filtering	15
Table 2.3 Summary of Different Types of Hybrid Recommendation Systems	20
Table 2.4 Different Criteria for Overall Rating	23
Table 3.1 A Sample of User-Item Matrix	50
Table 3.2 Sample User and Item Matrix	51
Table 3.3 Case Scenario Example	64
Table 3.4 First Scenario Example	65
Table 3.5 First Scenario new Ordering	65
Table 3.6 Second Scenario Example	66
Table 3.7 Second Scenario new Ordering	66
Table 3.8 Third Scenario Example	68
Table 4.1 Kuching Dataset Features	75
Table 4.2 MovieLens Dataset Features	76
Table 4.3 Confusion Matrix	82
Table 5.1 RMSE and MAE of Deep Learning Algorithms	88
Table 5.2 Training Speed of Deep Learning Methods	89
Table 5.3 Precision, Recall, F1-Score of Deep Learning Methods	90
Table 5.4 RMSE and MAE of Baseline Algorithms	91
Table 5.5 ResNetMF Improvement in RMSE & MAE	92
Table 5.6 Precision, Recall and F1-Score@10 Results of tested algorithms	93
Table 5.7 RMSE and MAE of Deep Learning Methods	95
Table 5.8 RMSE and MAE of Baseline Algorithms	96

LIST OF FIGURES

	Page
Figure 2.1: Proposed Tourism ID Architecture (Gavalas et al., 2014)	12
Figure 2.2: Recommendation System Types (Edited from (Roy & Dutta, 2022))	13
Figure 2.3: User-based Filtering Method	17
Figure 2.4: Item-based Filtering Method	18
Figure 2.5: Explicit vs Implicit Ratings (Falk, 2019)	24
Figure 3.1: Proposed Method Flowchart	40
Figure 3.2: System Architecture	43
Figure 3.3: Matrix Factorization Logic	52
Figure 3.4: Deep Residual Neural Network Concept	57
Figure 3.5: ResNet on Proposed System	58

LIST OF ABBREVIATIONS

MF	Matrix Factorization
CF	Collaborative Filtering
CBF	Content-based Filtering
NCF	Neural Collaborative Filtering
SVD	Singular Value Decomposition
NMF	Non-Negative Matrix Factorization
CNN	Convolutional Neural Network
CMF	Convolutional Matrix Factorization
RNN	Recurrent Neural Network
KNN	K-Nearest Neighbour
RMSE	Root Mean Squared Error
MAE	Mean Absolute Error
ResNet	Residual Network
ResNetMF	Residual Network Matrix Factorization
SVDpp	Singular Value Decomposition plus plus(SVD++)

CHAPTER 1

INTRODUCTION

1.1 Research Background

Over the past couple of decades, there has been a substantial surge in the volume of information attributed to the rapid and widespread proliferation of the World Wide Web. This trend has brought about a contemporary phenomenon known as information overload, whereby the magnitude of available information has exceeded the human capacity to process and manage it effectively (Caserio & Trucco, 2016). Therefore, conveying correct information to the right community is the key to proper information dissemination (Z. Zhang et al., 2021). Users face the challenge of the information overload era, and information producers suffer from this issue since they want to provide the right information to their users (G. Li et al., 2019).

The tourism industry is one of the most popular industries today. According to the Statista website, travel and tourism are among the most prominent industries worldwide, with a worldwide economic contribution surpassing 5.81 trillion U.S. dollars in 2022. However, this industry is resistant to the information overload predicament. With the development of new lifestyles, many people can travel.

Information overload in the tourism industry is a key challenge for stakeholders and tourists (Malhan & Dewani, 2023). Most tourists turn to the to obtain information about their destination, and other study shows that about three-quarters of tourists look to another tourism review about their destination before planning their trip (G. Li et al., 2019).

Some tourists prefer to use tour guides and join tours, while others prefer to discover their destination on their own due to the expenses of actual tour guides and the limitations of joining tours. E-tour guides play important roles in tourism information, education, and decision-making about their destination. E-tour guides are applications, websites, or apps that provide information about destinations to users. E-tour guide websites like TripAdvisor work based on user ratings and comments and provide information about different destinations. However, E-tour Guides also need help with the challenge of information overload (Meehan et al., 2016).

Therefore, scientists and researchers always seek the best-automated techniques to convey the right information to the right community and targeted users. Recommendation systems can address this issue. Recommendation systems use Machine Learning techniques to provide the right information according to user preferences (S. C. Nair, 2022). For this reason, using an intelligent recommendation system that disseminates information is the most desired way.

The recommendation system has merged into our daily life from e-commerce websites, Netflix, YouTube, and others. According to statistics, about 60 percent of videos on YouTube come from page recommendations (S. Zhang et al., 2020).

1.2 Problem Statement

The tourism industry faces the challenge of information overload that overwhelms tourists seeking destinations that align with their specific needs and preferences (Meehan et al., 2016). Traditional information dissemination methods often provide a "one-size-fits-all" approach, failing to address this issue.

Recommendation systems offer a personalized solution to the information overload problem in tourism. Unlike information dissemination, which provides generic information, recommendation systems tailor suggestions to user preferences through algorithms analyzing past behavior and user data (Fayyaz et al., 2020). These systems are prevalent in e-commerce, social media, and entertainment.

There are three main recommendation system types name as collaborative filtering (CF), content-based filtering (CBF), and hybrid filtering. CF leverages other users' evaluations, while CBF uses item descriptions and user profiles. Both have limitations such as cold start, sparsity, scalability for CF and insufficient data and unsuitable suggestions for CBF (S. C. Nair, 2022). Hybrid approaches combine these techniques for better performance and scalability.

However, these recommender systems often recommend items based on a simple overall rating or purchase history. This can be limiting because it doesn't account for the various factors that influence a user's decision .(Alamdari et al., 2020) Thus, multi-criteria recommendation systems, take into account various factors, or criteria, to suggest items that best match a user's preferences (Ifada et al., 2023), comes to place.

Let's have a scenario as follows. Imagine a system that recommends tourist destinations simply based on popularity. It might suggest the Eiffel Tower in Paris, even if the user is not interested in crowded landmarks. However, a multi-criteria recommender system would be more nuanced and could consider factors like travel style, budget, interests, and travel time. By considering these criteria and users' preferences on each one, the system could recommend destinations like a Hiking trail in Patagonia and white-water rafting in the Grand Canyon for adventure lovers, or a yoga retreat in Bali for relaxation seekers. This

way, the recommendations are more likely to align with your specific interests and travel style, leading to a more enjoyable and fulfilling vacation.

While multi-criteria recommendation systems offer a more personalized approach, they come with some challenges. First of all, these systems deal with more data points representing various criteria. This complexity can make it computationally expensive and difficult to identify patterns that lead to accurate recommendations (Zheng & Wang, 2022). The second challenge is related to Scalability and efficiency. As the number of users, items, and criteria grows, recommending efficiently becomes a challenge. The system needs to be scalable to handle large datasets without compromising recommendation quality and accuracy (Q. Y. Shambour et al., 2023). The third challenge is considering contextual factors. Ideally, the system should consider contextual factors that can influence user preferences (Meehan et al., 2016). For example, someone might prioritize relaxation on a solo trip but seek adventure when traveling with friends.

Therefore, this study aims to propose a recommendation system method that uses a multi-criteria technique that considers implicit data to increase recommendation accuracy, have less computationally expensive recommendation model, and be able to answer to user dynamic preferences in the tourism domain.

1.3 Objectives

This study proposes a multi-criteria approach for developing a recommendation system specifically designed for the tourism industry. Also, this study has the following objectives.

- i. To propose and validate a multi-criteria recommendation method that achieves an improvement of at least 8% in recommendation accuracy, measured by RMSE, compared to a baseline system like collaborative filtering on a publicly available dataset like MovieLens.
- ii. Evaluate and compare the training speed of the proposed method with existing methods like convolutional neural network (CNN) and Neural Collaborative Filtering (NCF). Aim to achieve at least a 10% reduction in training time compared to the baselines (less computationally expensive).
- iii. Develop and integrate a user query analysis module into the multi-criteria recommendation system to enable dynamic user interaction and preference expression with a minimum 50% user satisfaction rating as measured by average user rating after user query on Kuching city attractions.

1.4 Scope of Study

This research aims to develop a multi-criteria recommender system for personalized tourist destination recommendations. The system will leverage a combination of traditional matrix factorization and deep residual networks to analyze user data and recommend destinations that align with user criteria. These criteria may include factors like interests, and location. The system's effectiveness will be evaluated using metrics like precision and recall, and compared to the performance of existing recommendation system baselines. Additionally, the research will explore the potential for incorporating user query analysis features to address the user's dynamic preferences within the system's recommendations.

1.5 Thesis Structure

This thesis consists of six chapters. The outline of each chapter is summarized as follows:

Chapter 1 presents the general introduction of the topic, which includes the background study, problem statement, objectives, and research scope.

Chapter 2 presents a thorough literature review focusing on multi-criteria recommender systems in the tourism industry. It discusses various types of recommendation systems and delves deeper into existing research on multi-criteria approaches specifically applied to tourist destination recommendations. By reviewing the related literature, this chapter provides insights into the challenges and opportunities of developing an effective multi-criteria recommender system for the tourism industry.

Chapter 3 details the proposed system's methodology, outlining each component's logical steps and operations. It comprehensively explains how the system works and can be implemented effectively.

Chapter 4 provides about the experimental design for evaluating the proposed method. This chapter delves into different aspects of experiment design such as dataset selection, comparing baseline algorithm explanation, accuracy evaluation metrics and hyper parameter tuning.

Chapter 5 provides an in-depth discussion of the results of the experiments. Additionally, this chapter presents a detailed analysis of the results, including a comparison of the proposed method's performance with other benchmark algorithms.

Chapter 6 provides a conclusion to the research study and highlights the contributions of the proposed method. Additionally, this chapter presents potential future work that could be studied further. It also addresses the limitations and milestones of the research.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter explains information dissemination, recommendation system, and their relationships. Besides that, this chapter surveys different types of recommendation systems and the existing state of the art of recommendation systems. Besides, there will be a review of some researches related to tourism multi-criteria recommendation systems.

The chapter delves into the strengths and weaknesses of some of the most commonly used recommendation algorithms. It reviews relevant studies related to these algorithms, which helps readers identify which algorithm suits a particular problem.

Furthermore, this chapter summarizes the key takeaways from the review of the information dissemination framework, and recommendation systems. Readers will understand the significance of information dissemination, different recommendation systems, and their strengths and limitations. With this foundation, readers will be better equipped to understand the proposed solution for this study, which will be discussed in the following chapter.

2.2 Information Dissemination

Information dissemination and recommendation systems are closely related as both are concerned with providing users with relevant information. Information dissemination refers to sharing information with a large audience through various channels, such as news outlets, social media platforms, or websites (Huang et al., 2023). On the other hand,

recommendation systems are designed to offer personalized suggestions to individual users by considering their preferences, past behavior, and other relevant data (A. Nair et al., 2023).

Sometimes, recommendation systems can be seen as information dissemination, often used to promote new products, services, or content to users. For example, a recommendation system on a streaming platform might suggest a new movie or TV show to users based on their viewing history. However, recommendation systems can also be used to counteract the information overload problem that can result from information dissemination. By providing personalized recommendations, users can handle the sheer volume of information available and instead receive only the most relevant content. While information dissemination and recommendation systems have different goals and methods, they both play important roles in providing users with the information they need and want (Soleymani et al., 2023).

Information dissemination means sending information to an audience or a group of audiences. Sending information can occur through different mediums and channels, such as text messages, emails, blogs, TV news, and other forms. According to Klaib & Nuser (2019), information dissemination is a sub-discipline of information systems and science that involves the utilization of information for decision-making, establishing a formalized practice of information utilization, and responding to information needs.

We are oversaturated with information on the web due to its development. Thus, this causes users' poor decision-making and overwhelming. Thus, there is a huge need for techniques and systems to disseminate the right information to the right user(s). Information filtering techniques can help significantly address this issue.

Information filtering pursues a simple goal: to provide users with the most relevant or desirable content from the web. Contrasted with information retrieval, which seeks to present all results matching a query (Ceri et al., 2013), information filtering strives instead to identify the very best results and present only these (Khozooii et al., 2013).

Thus, having the most relevant and accurate information for decision-making is very important. The quality of decision-making can be highly affected by the quality of disseminated information (Yien et al., 2021). Information quality has several metrics, such as accuracy (closeness of a measurement to the true value), believability, relevancy, and completeness of the information (R. Y. Wang & Strong, 1996).

As the web and mobile applications have advanced, users expect a more personalized experience. Directing users to the appropriate information at the appropriate time and place is important. In order to achieve this, recommendation systems have been developed in many aspects of life, including marketing, entertainment, and education. Recommendation systems primarily focus on the user's interests, needs, or context in making decisions (e.g., recommending movies or books that match a user's preferences). However, these recommendation systems still need to address some problems, such as accuracy, reliability, and complexity.

For this reason, many researchers and stakeholders use recommendation systems, which are sub-class of information filtering, to better filter according to user's preferences and requirements. The recommender system (RS) filters out important information from a large pool of dynamically generated information to set some important decisions in terms of recommendations according to the user's past behavior, preferences, and interests (Kuanr &

Mohapatra, 2021). Moreover, a recommendation system is a method that helps the user to find useful information from a pool of available data (Sunilkumar, 2020).

Gavalas et al. (2014) outlined that recommendation systems are among the most important parts of tourism information dissemination. Since recommender systems are helping in managing the overwhelming amount of user-generated data on the internet by filtering it (Sinha & Dhanalakshmi, 2022).

Gavalas et al. (2014) proposed an information dissemination (ID) architecture for tourism. They claimed that the tourism industry is diverse in architectural and technological aspects. Therefore, they propose a multi-faced scheme classification of systems for tourism. The proposed architecture is shown in Figure 2.1.

According to Gavalas et al. (2014), recommendation systems are one of the important components of disseminating information in tourism. The proposed architecture consists of the main components of any information system, which are input, processes, and output. In this architecture, recommendation systems are the heart of information processing. This architecture has two inputs which are user query and user profile. Also, it obtains information from various sources, such as the tourism content database and external sources.

Furthermore, the job of a recommendation engine is to analyze all data from different sources, filter the best-selected and most relevant information from various sources, and pass the result to the communication channel. Communication channels integrate the result in the best form and then disseminate the result. In this case, the channel is a user's mobile device.

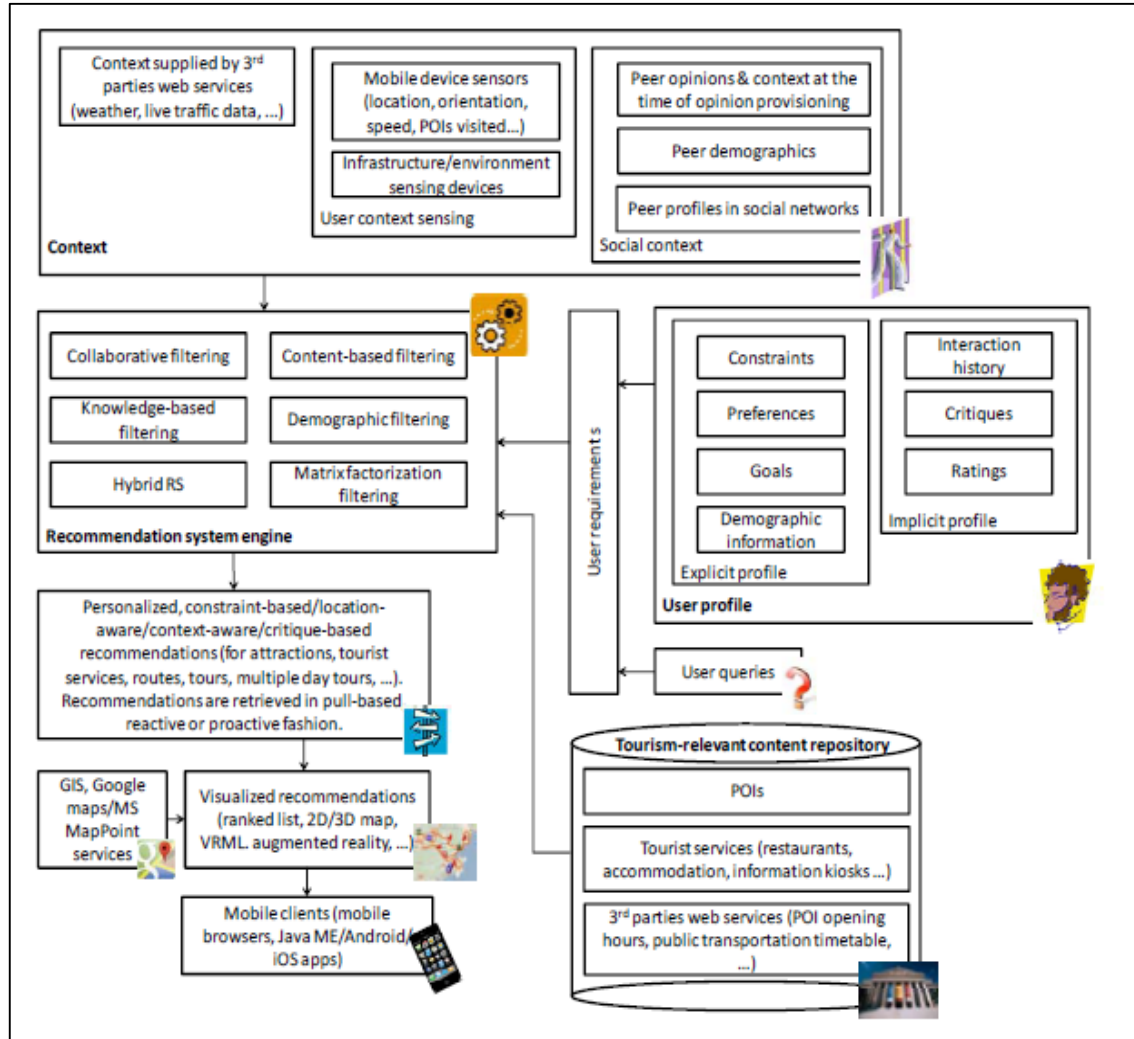


Figure 2.1: Proposed Tourism ID Architecture (Gavalas et al., 2014)

Therefore, according to this architecture, information filtering is important in information dissemination. Recommendation systems can hugely help to increase dissemination accuracy due to they are assisting in creating a platform that uncovers what users are interested in by looking at their records and behavior (Sinha & Dhanalakshmi, 2022).

In this study, this architecture is the benchmark for information dissemination purposes. Thus, this study only focuses on improving one of the main components, recommendation systems, for more accurate information dissemination. As displayed in

architecture, recommendation systems have various types that will be explained further in the next section.

2.3 Recommendation Systems

Recommendation systems are information filtering systems designed to suggest smaller choices to users from a huge pool of selections. The result must be personalized based on user information.

For example, the recommender system needs to display the top 20 choices of items the user may like rather than display all sets of selections, sometimes thousands of choices. There are three primary classifications of recommendation systems: content-based filtering, collaborative filtering, and hybrid filtering, as shown in the figure.

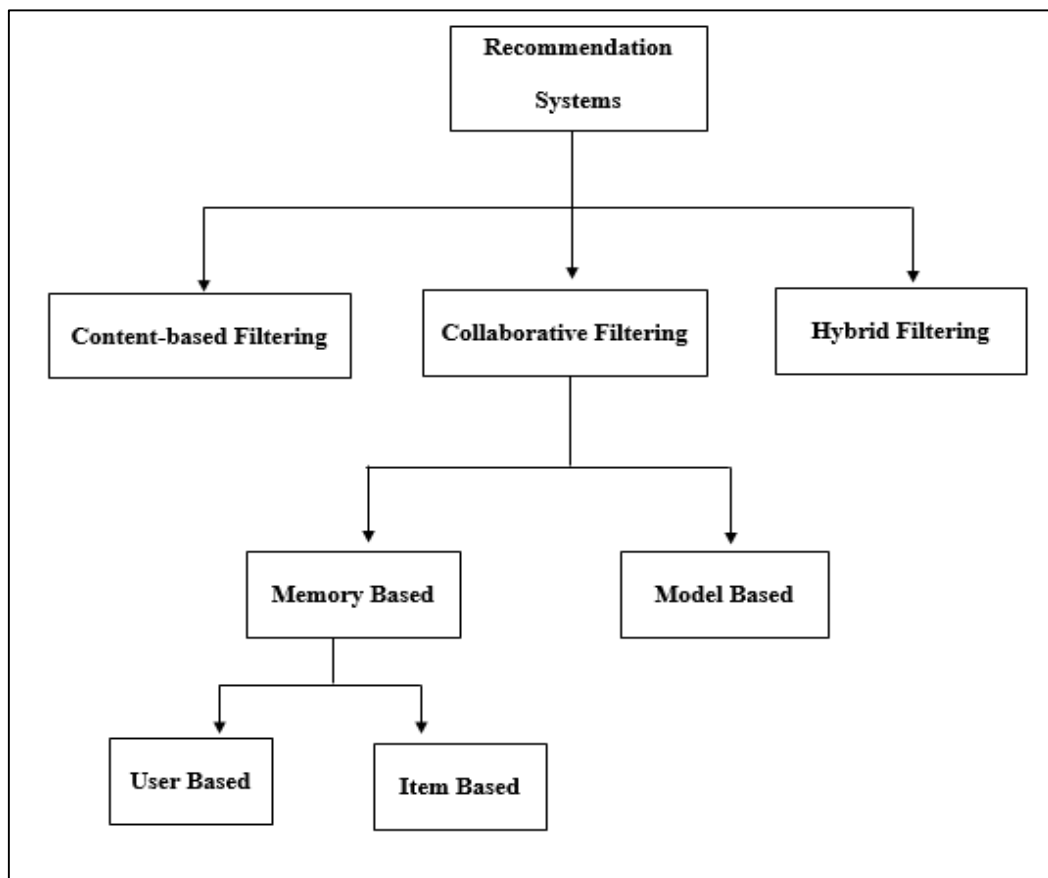


Figure 2.2: Recommendation System Types (Edited from (Roy & Dutta, 2022))

2.3.1 Content-Based Filtering

Content-based filtering methods work best on datasets that have a description of items and user profiles (Adilaksa & Musdholifah, 2021). Content-based filtering considers a user-centric approach that learns from the user's preferences and recommends according to user preferences. This approach needs description or data about users to recommend the best choice for users.

A recommender system that uses content-based filtering creates a user profile based on their data. These data can be explicit, like ratings or like user click rates. Once the user profile is created based on these data, the recommender system can suggest me according to the profile (Thannimalai & Zhang, 2021).

For example, a certain football club fan only watches football-related stuff. The user profile is created based on the information gathered as a football fan. Thus, mostly football-related items are recommended to this user. Facebook uses this approach for target ads, and Facebook watches video suggestions. Suppose there are two products, the first one is a Real Madrid jersey, and the second one is a badminton bat; then, according to this user profile and preferences, the Real Madrid jersey will be recommended.

Content-based filtering has advantages such as possible comparison between items and community information not required since it is based on user profile (X. Wu, 2022). This approach has disadvantages, such as a need for content description or domain knowledge of items, a cold start for new users, and no surprise in recommendation results since it is based on only the user's current interest. Also, some degree of knowledge engineering efforts is required (Parasuraman & Elumalai, 2021).

2.3.2 Collaborative Filtering

Collaborative filtering builds a model based on the history of items or users (Gai et al., 2019). This approach operates under the assumption that users who have shown interest in similar items in the past are likely to be interested in similar items in the future (Gai et al., 2019). This approach only relies on rating information about users or items. This method searches for users with similar rating patterns as the current user, utilizes the ratings these like-minded users give, and computes a forecast for the current user. Suppose the table below displays the likeness of different users with items.

Table 2.1: Basic Collaborative Sample

	Item 1	Item 2	Item 3	Item 4
User 1	Y	N	Y	Y
User 2		Y	N	N
User 3	Y	Y	N	
User 4	N		Y	
User 5	Y	Y	?	N

Suppose the system wants to predict item 3 likeness to user 5. First, the system looks for users with similar rating patterns to user 5. In this case, user 2 and user 3 have the same rating patterns as user 5, as highlighted in the table below. Since user 2 and user 3 have an “N” for item 3, thus the system predicts that it is an “N” for user 5.

Table 2.2: Solved By Collaborative Filtering

	Item 1	Item 2	Item 3	Item 4
User 1	Y	N	Y	Y
User 2		Y	N	N
User 3	Y	Y	N	
User 4	N		Y	
User 5	Y	Y	N	N

While collaborative filtering is a popular approach for building recommendation systems, it's not without its challenges (G. Li et al., 2019). One hurdle is data sparsity. This occurs when users haven't rated many items, or there are missing values in the data. This makes it difficult for the system to accurately predict user preferences (Natarajan et al., 2020). Another challenge is the cold start problem. This arises when the system encounters a new user or item with little to no data. Without enough information, the system struggles to generate effective recommendations (Hwang & Chae, 2022). Finally, collaborative filtering can face scalability limitations. As the amount of data fed to the system grows rapidly, the system can become overwhelmed. This makes it challenging to maintain efficiency and accuracy with large datasets (Natarajan et al., 2020).

There are two main sub-methods for this approach, namely model-based collaborative filtering and memory-based collaborative filtering. The latter is also referred to as neighborhood-based methods.

2.3.2.1 Memory-Based Collaborative Filtering

Memory-based collaborative filtering is a basic method that uses no models; the recommendations are made only on user past data (Nam, 2022). This method can use a simple distance-measurement approach, such as the nearest neighbour. This method only remembers how the user interacts with an item based on the rating user gives to an item (Nam, 2022). This approach has two sub-methods: user-based filtering and item-based filtering.

2.3.2.2 User-based Filtering Method

Suppose the system discovers that user A preferences and characteristics are similar to user D. Thus, the system recommends items that user D likes to user A. The users who like the same products as you will like the products you like. This method recommends based on similarities between users.

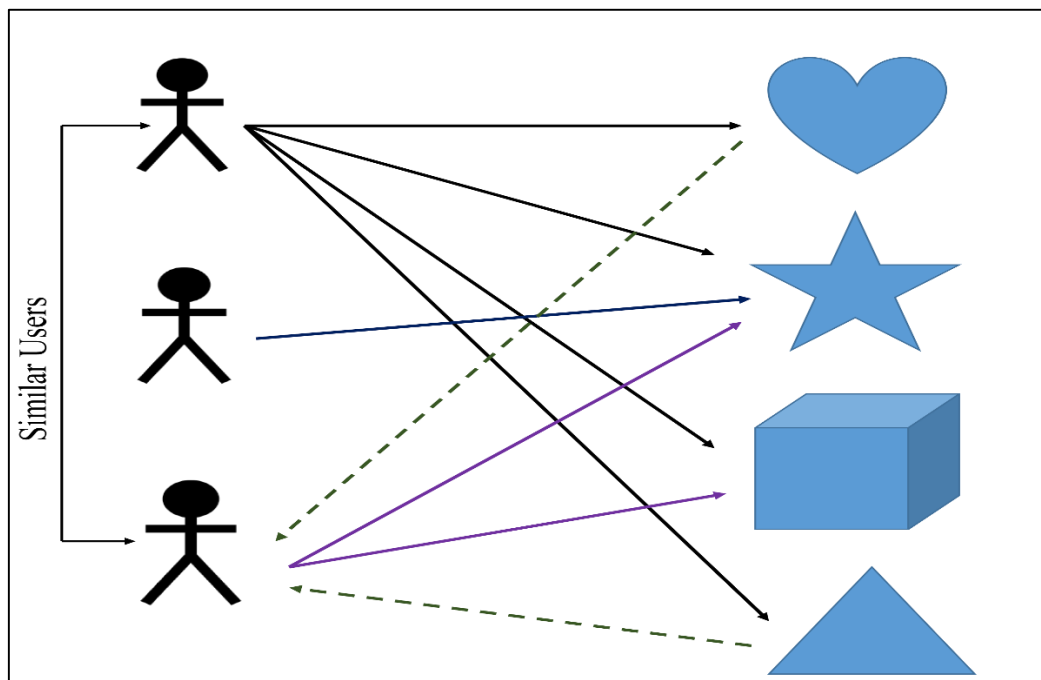


Figure 2.3: User-based Filtering Method

2.3.2.3 Item-based Filtering Method

Suppose user A likes item 1, a recommendation system might suggest items 2 and 3 because they are similar to item 1.

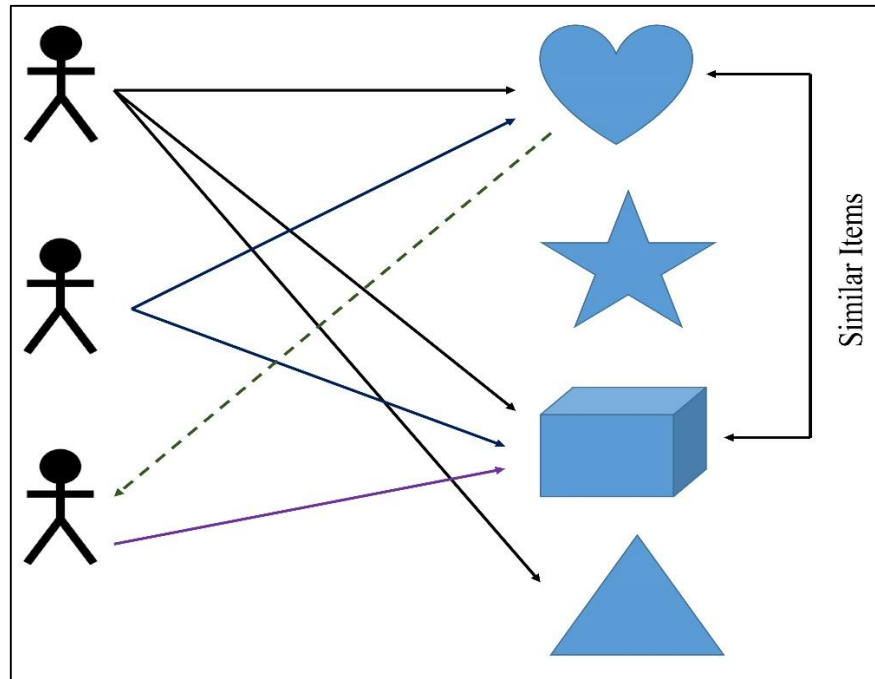


Figure 2.4: Item-based Filtering Method

Either item-based or user-based filtering, the similarity matrix needs to be created. An item-similarity matrix is generated through item-based collaborative filtering, which evaluates the similarity between two comparable items (Musa & Zhihong, 2020). Creating a user-similarity matrix that calculates the similarity between two users with similar preferences is necessary for user-based collaborative filtering.

2.3.2.4 Model-based Collaborative Filtering

Model-based collaborative filtering employs data mining and machine learning algorithms to make predictions for unassessed data. This method is based on learned models and only uses the learned model to make predictions. This method updates models frequently, and many techniques can be used, but it may make it computationally expensive (Pujahari & Sisodia, 2020).

The advantages of collaborative filtering approaches are that they do not need the description of items and users, there is no need for knowledge engineering efforts, they can make surprising results, and they can learn market segments (Y. Zhang et al., 2023). However, the collaborative filtering method requires rating feedback, and it faces the challenge of a cold start for new items and users.

2.3.3 Hybrid Filtering

The hybrid filtering technique uses a combination of two or more methods together. Hybrid methods refer to recommendation systems that are a combination of two or more different models (Y. Zhang et al., 2023). For instance, a hybrid model refers to integrating two or more recommendation algorithms. One type of hybrid model involves combining model-based collaborative filtering with item-based collaborative filtering, while another type involves blending content-based and collaborative filtering algorithms. It could also entail merging model-based collaborative filtering with memory-based collaborative filtering (Parasuraman & Elumalai, 2021). A hybrid method is employed when a single recommendation method, such as memory-based collaborative filtering, cannot address the issue. This approach can be expensive. However, it increases accuracy and reduces information loss during the recommendation. Hybrid recommender systems are classified into different categories: weighted, switching, cascade, feature augmentation, feature combination, meta-level, and mixed (Burke, 2002). The table below briefly describes these approaches.

Table 2.3: Summary of Different Types of Hybrid Recommendation Systems

Hybrid Methods	Descriptions
Weighted	The decided recommendation result obtained from different recommendation system scores (Pazzani, 1999). Different recommender systems results are combined into a final recommendation result based on their weights
Switching	Select a certain recommendation system based on the requirements
Cascade	A final recommendation system enhance the results of other recommender systems
Feature Augmentation	The output from one recommendation algorithm serves as input for another recommendation system. Features of one knowledge source is computed to make it compatible to work as an input for any other recommender algorithm (H. Zhang et al., 2018). The first recommendation result is the input of another one.
Feature Combination	The features of first recommender systems added to features of other recommendation systems
Meta-level	A previously trained model is utilized as an input for another recommendation system.
Mixed	The results of different recommendation systems are mixing as a final result

In recent years, to overcome other recommendation systems challenges, hybrid approaches have been the most used technique (Yu et al., 2020). As far as it is observed, most current approaches use hybrid methods.

The parameters to select a hybrid recommendation system type may depend on several factors, such as domain, data size, user behavior, performance, scalability, and the goal of the recommendation system.

The nature of the domain and the type of data available can influence the choice of a hybrid recommendation system. For example, content-based systems are more suitable for domains where the item features are easily available, while collaborative filtering is better for domains where user feedback is abundant. The dataset size can impact the choice of a

hybrid recommendation system. Matrix factorization-based models may be more appropriate for large datasets, while content-based or rule-based models may work better for smaller datasets.

Understanding user behavior is critical in selecting the right recommendation system. For example, if users rate items similarly, a content-based system is more appropriate than collaborative filtering. The objective of the recommendation system is also important. For example, if the goal is to recommend new items, a hybrid system combining collaborative and content-based filtering may be more suitable.

The performance of the system is another key factor. The recommendation system's accuracy, coverage, and diversity are important factors when selecting a hybrid recommendation system. The scalability of the system is another important factor to consider. Some algorithms may need to be more scalable to large datasets or require significant computational resources. Selecting the best hybrid recommendation system type involves balancing these parameters and considering the application's specific requirements.

2.4 Multi-Criteria Recommendation Systems

A multi-criteria recommendation system goes beyond simply suggesting items based on past user behavior or item popularity (Zeeshan et al., 2021). It takes into account multiple factors that contribute to a user's preference for an item, aiming to provide more personalized and relevant recommendations. As an example, a movie recommendation system that goes beyond just looking at what movies user liked in the past. It might also consider other factors such as user ratings on specific factors like acting, directing, or cinematography or whether the user prefer comedies, dramas, or action movies.

Single criteria recommendation systems recommend items based on a single factor, such as popularity, ratings, or similarity to items previously liked by the user (García-Crespo et al., 2011). On the other hand, multi-criteria recommendation systems consider multiple factors, such as user preferences, item attributes, and contextual factors, to make recommendations (Ko et al., 2022).

Single-criteria recommendation systems are generally simpler and easier to implement. However, they may not capture the full complexity of user preferences and may not provide personalized recommendations for diverse user groups (Arif et al., 2022). In contrast, multi-criteria recommendation systems can provide more personalized and diverse recommendations by considering multiple factors, but they are more complex and may require more data and computational resources (Q. Y. Shambour et al., 2022).

Users can be interested in the items based on different criteria. For example, in a hotel recommendation system, a user may be interested in the cleanliness of the room. Another user may be interested in the hotel's location, and both users give a 4-star rating to the hotel. In this case, the overall rating of the users has a poor reflection of the user's overall preferences for recommendation tasks. Thus, the recommendation system must depend on more than overall ratings. For example, consider Table 2.4; three users rated a hotel and expressed their criteria for a specific hotel based on cleanliness, location and price, and overall ratings. The range of ratings is from 1 to 5. As the table below, Emmy and Chai have the same overall ratings, but their patterns of ratings for cleanliness and location are far different. However, Nick and Chai have different overall ratings for the hotel, but their overall ideas about cleanliness and location are similar. Therefore, Chai and Nick are more

similar rather than Chai and Emmy. If the recommendation task is only based on the overall rating, it may cause fewer related predictions.

Table 2.4: Different Criteria for Overall Rating

	Criteria			
Users	Cleanliness	Location	Price	Overall
Emmy	1	4	4	4
Chai	4	1	4	4
Nick	4	1	4	3

Traditional collaborative filtering and recommendation systems mostly concentrated on user preference modelling for an item only based on overall ratings, which are addressed as single preference ratings. However, it is possible to capture deeper user preferences with several preference dimensions. For example, in Table 2.4, the recommendation system considers not only overall ratings and preferences but also deeper preferences and aspects of the users, such as cleanliness and location, which can improve recommendation quality. The term quality refers to the relatedness of recommended items to the users.

There are two different types of ratings of the users which are explicit and implicit ratings. Users directly give explicit ratings, and users manually rate an item. In contrast, implicit ratings can be extracted from the user's behavior (Chala et al., 2020). An example of implicit rating can be if a user asks for more information about an item, it can be considered positive feedback from the user. The figure below is from the Practical Recommendation Systems book, which properly explains the explicit and implicit ratings.

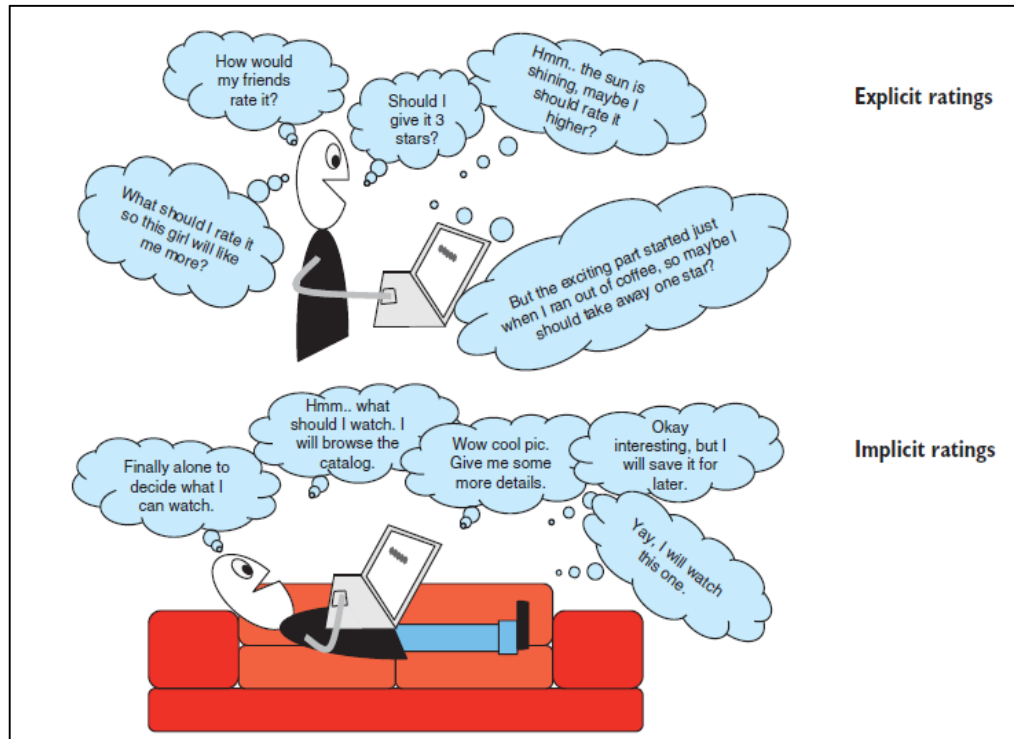


Figure 2.5: Explicit vs Implicit Ratings (Falk, 2019)

In the tourism industry context, a single criteria recommendation system may recommend popular tourist destinations based on user ratings or geographic proximity. However, it may not consider other factors such as user preferences, budget, and travel style (Arif et al., 2022). On the other hand, a multi-criteria recommendation system can consider these factors and provide more personalized recommendations that match the user's preferences and constraints.

A multi-criteria recommendation system is useful in the tourism industry. It has several advantages, such as improving recommendation accuracy, and diverse preferences, helping to increase customer satisfaction, enhancing decision-making, and enhancing competitiveness (Q. Y. Shambour et al., 2022). Let us discuss them in more detail.

Multi-criteria recommendation systems can improve recommendation's accuracy by considering multiple factors that influence user decision-making and can result in more relevant recommendations and increased user trust in the system.

Tourists have diverse preferences and interests, and a multi-criteria recommendation system can consider various criteria such as location, activities, budget, and accommodation preferences to provide personalized recommendations. By providing personalized recommendations, a multi-criteria recommendation system can increase customer satisfaction and improve their overall travel experience (Wan, 2022). Hence, single-criteria recommendation systems tend to recommend the same popular destinations or accommodations to all users, leading to a need for more diversity in recommendations. Multi-criteria recommendation systems can consider a wider range of options and suggest less popular but still suitable destinations and accommodations, providing users with more options.

A Multi-criteria recommendation system can cause efficient decision-making since, with the vast amount of information available online, it can be overwhelming for tourists to make decisions about where to go, what to do, and where to stay (Forouzandeh et al., 2022). A multi-criteria recommendation system can streamline decision-making and provide recommendations matching tourist preferences (Forouzandeh et al., 2022). A multi-criteria recommendation system in a highly competitive tourism industry can give businesses an edge by providing personalized recommendations that meet tourists' needs and preferences.

Using multi-criteria recommendation systems can lead to increased user satisfaction, improved accuracy, increased diversity of recommendations, and greater flexibility to adapt to changing user needs and preferences. A multi-criteria recommendation system can help

tourism businesses provide a better customer experience, improve efficiency, and gain a competitive advantage in the industry.

However, multi-criteria recommendation systems can have several challenges. Modelling user preferences is one of the challenges. it means accurately capturing and combining various factors in a way that reflects real-world preferences can be challenging (Z. Li, 2018). Another challenge is users evolving preferences. user preferences and the importance they attach to different criteria can change over time (Gupta & Kant, 2022). This could be due to various factors like personal experiences, trends, or evolving needs. Scalability and computational complexity are also challenges of multi-criteria recommendation systems. As the number of criteria and users increase, the system can become computationally expensive and struggle to handle the growing data volume (Gupta & Kant, 2022). This necessitates efficient algorithms to maintain scalability and deliver timely recommendations.

2.5 Tourism Multi-Criteria Recommendation System

Several studies used multi-criteria recommendations for tourism. As an example, (Q. Shambour, 2021) proposes a deep learning approach using autoencoders to capture hidden relationships between users and items based on multi-criteria preferences. It outperforms state-of-the-art recommendation algorithms on datasets like Yahoo! Movies and TripAdvisor. This research highlights the potential of deep learning techniques for improving recommendation accuracy in multi-criteria scenarios. This research proposes a deep learning-based algorithm for multi-criteria recommender systems, specifically utilizing autoencoders to address the challenge of capturing complex, non-linear relationships between users and items based on multi-criteria preferences.

This research uses Autoencoder-based Multi-Criteria (AEMC). The core component, AEMC is a deep learning architecture composed of multiple autoencoder layers. Each layer learns a compressed representation of the previous layer, capturing increasingly abstract and non-linear features. The AEMC learns low-dimensional vector representations (embeddings) for both users and items, encoding their multi-criteria preferences and characteristics. The learned user and item embeddings are combined to predict user ratings for unseen items, considering their multi-criteria preferences. The model is trained using backpropagation to minimize the prediction error between actual and predicted ratings.

The research demonstrates that the AEMC-based algorithm outperforms traditional collaborative filtering and content-based filtering methods on benchmark datasets like Yahoo! Movies and TripAdvisor. Also, the AEMC architecture is capturing complex, non-linear relationships between users and items based on multi-criteria preferences, potentially leading to more accurate recommendations. Another advantage highlighted by the researcher is that the AEMC can be adapted to different types of multi-criteria data and recommendation scenarios.

However, this approach has several disadvantages. Similar to many recommender systems, the model might struggle with data sparsity, especially for users with limited rating history or new items. The research doesn't explicitly address how it handles the cold-start problem for new items or users, potentially impacting recommendation accuracy in such cases. While the research mentions exploiting hidden relationships, the AEMC model can be challenging to interpret, making it difficult for users to understand the reasoning behind recommendations. Lastly, the training large-scale AEMC models can be computationally expensive, potentially limiting scalability for real-world applications with massive datasets.

Another research by Kumar et al. (2023) proposes a personalized travel recommendation system called Average Cumulative Rating Matrix Factorization (ACRM). It aims to address the limitations of existing systems by incorporating both explicit ratings and implicit sentiments from user reviews for more accurate recommendations. The system combines explicit user ratings with extracted sentiment scores from reviews to create a more comprehensive "Average Cumulative Rating" for each destination. The ACR scores and user preferences are used in a Matrix Factorization (MF) model to capture latent features of users and destinations. Based on the learned user and destination features, the system recommends personalized travel destinations that align with individual preferences.

This research has several strengths such as combining explicit ratings with implicit sentiments for more comprehensive recommendations. Leverages Matrix Factorization for efficient feature representation and recommendation generation. Offers a flexible framework adaptable to different data sources and recommendation scenarios.

However, this study has several limitations. It relies heavily on the accuracy of sentiment analysis techniques. Also, it may not capture all aspects of user preferences beyond the extracted features. Lastly, it is a limited evaluation of real-world datasets compared to some other approaches.

This research presents an interesting approach to personalized travel recommendations by incorporating sentiment analysis and Matrix Factorization. While it has some limitations, it represents a valuable contribution to exploring new methods for enhancing recommendation accuracy and user satisfaction in the travel domain.

Zheng & Wang (2022) proposed a Multi-Criteria Recommendation Framework. The framework is inspired by the concept of Pareto ranking in multi-objective optimization

(MOO) to rank items based on multiple criteria. It utilizes Fonseca and Fleming's ranking method to assign a score to each item based on its dominance over other items in terms of the criteria.

This model directly uses predicted criteria ratings to rank items without predicting an overall rating. It calculates a ranking score for each item based on how many other items it dominates in terms of all criteria. Two differential strategies of averaging and crowding distance are explored to address cases where items have the same ranking score. This model combines the Multi-Criteria Rank (MCRank) model with existing multi-criteria recommendation models (MCRS) that predict overall ratings. The ranking score is calculated as a weighted combination of the MCRank score and the predicted overall rating, with the weight-optimized through a grid search.

MCRank generally outperforms the baseline models, demonstrating the effectiveness of using Pareto ranking for item recommendation. The hybrid model often further improves performance, but the optimal weight and effectiveness depend on the specific baseline model and dataset. However, the study focuses on two differential strategies for handling items with the same ranking score. It is possible that other strategies could be more effective, and future work could explore this possibility. Also, the study does not investigate the computational complexity of the framework. This could be an important factor for large-scale recommendation systems, and future work could explore ways to make the framework more efficient.

Another study by Hong & Jung (2021) proposed a multi-criteria recommendation system for the tourism industry, specifically focusing on restaurant recommendations. The research introduces a multi-criteria tensor model to incorporate various factors such as food

quality, service, price, and overall experience, along with cultural factors, into restaurant recommendations. It discusses how user experiences in restaurants are influenced not only by tangible factors like food quality but also by intangible factors such as service quality and dining atmosphere. Additionally, it mentions external factors like demographics and weather influencing user judgments.

This study explains tensor factorization as a method to predict missing user preferences for restaurants. It describes the optimization objective and the use of Stochastic Gradient Descent (SGD) for minimizing the loss function. The research utilized a dataset from TripAdvisor containing user reviews of restaurants. Various comparison methods were employed, including K-Nearest Neighbors-based CF, Co-clustering-based CF, and Singular Value Decomposition-based MF, among others. The results of the experiments are analyzed, comparing the proposed methods 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.

The study explores the impact of these factors on the recommendation performance. The study finds that by considering multiple criteria improves recommendation performance. Also, considering cultural background further improves recommendation performance and user-based models outperform country-based models. Moreover, it finds that a combination of user preferences and cultural background provides the most significant improvement.

While the study provides a comprehensive analysis of multi-criteria recommendation systems incorporating cultural factors, some research gaps can be identified. First of all, there is a limited cultural group identification. The study primarily utilizes geopolitical

cultural groupings, potentially overlooking more nuanced cultural differences that could impact restaurant preferences. second, the study could not handle partial or missing data. Although sentiment analysis is mentioned as a potential solution, the study does not delve into specific techniques or strategies for handling partial or missing multi-criteria ratings effectively. While the importance of temporal factors is acknowledged, the study does not explore how temporal dynamics interact with cultural factors in shaping user preferences over time.

2.6 Key Findings

Based on the existing literature, it is evident that multi-criteria recommendation systems encounter several significant challenges that warrant further investigation. Among these challenges, three key areas stand out: computational complexity, the comprehensive capturing of user preferences to improve recommendation accuracy, and the accommodation of dynamic user preferences over time.

First of all, computational complexity poses a significant obstacle to the practical deployment of multi-criteria recommendation systems. For instance, the model proposed by Shambour (2021) exhibits issues related to computational complexity. The process of training large-scale Autoencoder-based Multi-Criteria (AEMC) models can be prohibitively computationally expensive, potentially impeding their scalability and applicability in real-world scenarios with extensive datasets. Addressing this challenge is essential to ensure the feasibility and efficiency of deploying such systems on a large scale.

Secondly, while efforts have been made to capture user preferences comprehensively, there remain gaps in the ability of existing approaches to capture all relevant aspects of user preferences beyond the extracted features. For instance, the approach

presented by Kumar et al. (2023) may not fully encapsulate the diverse range of factors that contribute to user preferences, potentially resulting in incomplete or inaccurate recommendations. Bridging this gap requires developing more nuanced and inclusive models that account for a broader spectrum of user preferences and behaviors, thereby enhancing the accuracy and relevance of recommendations.

Thirdly, the dynamic nature of user preferences presents a formidable challenge for multi-criteria recommendation systems. Users' preferences and interests evolve over time, necessitating adaptive recommendation algorithms capable of accommodating these changes. However, existing studies, such as those conducted by Zheng & Wang (2022) and Hong & Jung (2021), often overlook the temporal dimension of user preferences and do not investigate how their recommendation frameworks adapt to evolving user dynamics. Addressing this gap is crucial for ensuring the continued relevance and effectiveness of recommendation systems in dynamic user environments.

While multi-criteria recommendation systems hold significant promise for enhancing recommendation accuracy and user satisfaction, they face notable challenges related to computational complexity, comprehensive user preference capturing, and dynamic preference accommodation. Addressing these research gaps is essential for advancing the state-of-the-art recommendation system design and facilitating their widespread adoption in practical applications.

2.7 Key Takeaways

The nature of tourism data can be complex and diverse, and effective recommendation systems for tourism need to consider multiple types of data to provide personalized and accurate recommendations.

The tourism industry is a data-intensive industry, and many stakeholders are using recommendation systems to address information overload in the tourism industry. Therefore, this study must consider factors other than ratings to solve these issues. Most recommendation systems only use user ratings rather than other additional information to build their models (G. Li et al., 2019).

Given the advantages of a multi-criteria recommendation system and the availability of diverse data types in the tourism industry, a hybrid recommendation system that considers multiple criteria is proposed. This study aims to enhance recommendation accuracy and provide more relevant suggestions to users meanwhile offers less computationally expensive solution by provide faster system which can be useful for real-world application. In addition to improving accuracy, a multi-criteria hybrid recommendation system can provide users with a more diverse set of options, streamline the decision-making process, and enhance customer satisfaction. Hence, developing a multi-criteria hybrid recommendation system can be a valuable asset for businesses in the tourism industry, helping them stay competitive and provide exceptional customer experiences. There are several reasons why a multi-criteria recommendation system is useful, such as improved accuracy, increased coverage, and more personalized recommendations.

Improved accuracy and relevance: A multi-criteria recommendation system can improve accuracy by considering multiple factors that influence user preferences, the system can paint a more nuanced picture of individual needs and desires (Hasan et al., 2022). This leads to more accurate and relevant recommendations that are tailored to the specific user and their unique preferences across various criteria (Hasan et al., 2022).

Enhanced User Experience: Receiving recommendations that genuinely align with their needs and preferences leads to a more satisfying user experience. Users are more likely to be engaged with the system and trust its suggestions when they are relevant and personalized (A. Nair et al., 2023).

More personalized recommendations: A multi-criteria recommendation system can provide more personalized recommendations by leveraging various data points and criteria, multi-criteria systems have the potential to personalize recommendations to a much greater extent. This can lead to a more individualized and engaging user experience (Kaya & Kaleli, 2023).

In the tourism industry context, this study proposed a multi-criteria hybrid recommendation system to improve the recommendation system's accuracy, address user dynamic preferences, and offer a less computational model that has less training time to be suitable for real-world applications yet has higher accuracy. This system considers users' implicit feedback in addition to their past behavior and ratings, which can lead to better recommendations than baseline collaborative filtering algorithms.

This study proposes a multi-criteria hybrid recommendation system for several reasons. First, Users may have different preferences or criteria for making decisions, such as price, quality, or popularity. A multi-criteria recommendation system can consider multiple criteria and provide recommendations that satisfy diverse users' preferences. Secondly, some decision-making scenarios may need to be more complex to be addressed by a single criterion. A multi-criteria recommendation system can consider both criteria and provide recommendations that balance the trade-offs and help users with complex decision-making. The third reason is that the available information may be limited or uncertain in some cases,

making it difficult to make accurate recommendations based on a single criterion. A multi-criteria recommendation system can use multiple criteria to compensate for the uncertainty and provide more reliable recommendations. Lastly, by considering multiple criteria, a multi-criteria recommendation system can improve the accuracy of the recommendations.

The proposed system can offer personalized recommendations that cater to tourists' unique needs. The proposed multi-criteria recommendation system can also overcome the limitations of single-criteria recommendation systems that recommend items based on a single factor, such as popularity, or ratings. The proposed system can leverage multiple criteria to offer a wider range of options that match the tourist's unique preferences and constraints.

Furthermore, the proposed system can be designed to adapt to changing user needs and preferences over time. This goal can be achieved by incorporating feedback mechanisms that allow users to write queries according to their needs. This information can be used to refine the system's algorithms and improve its accuracy in making personalized recommendations. As an example, if a user wants a luxurious hotel on a budget, the system feedback mechanism searches through the textual information for such a user request and disseminates the most related item based on user criteria and item textual history.

In addition, the proposed method can be designed to have less computational complexity and less training time which can be useful for real-world applications. Training speed refers to how long it takes for a recommendation system to learn from data and become effective. If a system takes a long time to train on a small dataset, it might become impractical when dealing with much larger datasets. By focusing on faster training, the aim

is the proposed method can handle not only small datasets but also potentially much larger ones. This ensures the system is scalable and practical for real-world applications.

In conclusion, the nature of tourism data is complex and diverse, and developing effective recommendation systems for tourism requires the consideration of multiple types of data to provide personalized and accurate recommendations. By leveraging a multi-criteria hybrid recommendation system, tourists can have highly personalized and accurate recommendations. This can lead to increased tourist's satisfaction, improved decision-making, and a more competitive edge for tourism businesses.

2.8 Chapter Summary

This chapter comprehensively overviews the information dissemination framework and its various components. Additionally, this chapter discusses different types of recommendation systems and their relevance in various contexts. By exploring the different types of recommendation systems, readers will better understand how these systems can be applied to solve various problems related to information dissemination. Also, the multi-criteria recommendation system is explained and four states of arts researches about to multi-criteria recommendation systems related to tourism reviewed in details. The gaps of research presented and possible solution is proposed. The following chapter is about the proposed solution for this study.

CHAPTER 3

METHODOLOGY AND DESIGN OF PROPOSED METHOD

3.1 Introduction

This chapter delves into the novel multi-criteria recommendation method designed to enhance both accuracy and personalization. It consists of two main components of recommendation engine and query analysis. Unlike traditional approaches, the proposed multi-criteria recommendation method (ResNetMF) integrates residual networks (ResNet) with matrix factorization. This innovative fusion aims to capture richer, more nuanced data patterns reflecting user preferences and item characteristics. ResNet specifically addresses the vanishing gradient problem, a common obstacle in deep learning applications. By overcoming this challenge, ResNet allows the model to learn more effectively from complex datasets, potentially leading to significantly improved recommendation accuracy compared to existing methods. Furthermore, a user query analysis component is incorporated. This component analyzes user queries to further personalize recommendations and ensure they align with evolving user preferences. This two-pronged approach of capturing complex data patterns and tailoring recommendations aims to deliver a highly personalized and accurate recommendation experience.

3.2 Methodology

In today's information-saturated world, discovering the right product or service can be an overwhelming task. Fortunately, multi-criteria recommendation systems are here to help. They delve deeper, weaving a tapestry of personalized recommendations by considering a multitude of factors. This study proposes a multi-criteria recommendation

model which can consider different features of an item rather than rating to increase accuracy and also it is able to answer to the user dynamic preferences.

Imagine this scenario. A user planning a weekend getaway and seek the perfect hotel. A multi-criteria recommendation system kicks into action. First, it analyzes your user profile, identifying your preferences for location, budget, amenities, and even past travel experiences. This comprehensive understanding forms the foundation for generating an initial recommendation list.

However, the system's journey doesn't end there. The user then input specific query, perhaps "romantic getaways near hiking trails." The system doesn't simply scan for keywords. Instead, it breaks down your query to understand the nuances of your search. It recognizes the desire for both romance and outdoor adventure.

At the heart of the system lies its ability to analyze reviews and comments that unlocks a deeper understanding of user preferences. By delving into the experiences of past guests, the system gleans valuable insights from their words. It can identify reviews mentioning romantic ambiances or proximity to hiking trails, further refining its grasp of what truly matters to user.

Armed with this multi-layered analysis, the system then takes the initial recommendation list and re-orders it dynamically. Hotels that best match user specific query, considering both your profile and the nuances of your search, rise to the top of the list. This ensures that the recommendations the user receives are not just generic suggestions, but highly tailored to user unique needs and desires.

Finally, the re-ordered list is displayed for exploration. This personalized list empowers the user to make informed decisions, saving time and frustration in user search for the perfect hotel. The multi-criteria recommendation system can transform what could have been a tedious task into an efficient and enjoyable experience.

This system's strength lies in its ability to go beyond the surface level. It not only understands user preferences but also listens to the voices of past guests and interprets the subtle nuances of your queries. This multi-faceted approach ultimately leads to a more personalized and successful discovery journey. The flowchart below displays the processes of the proposed model at a glance.

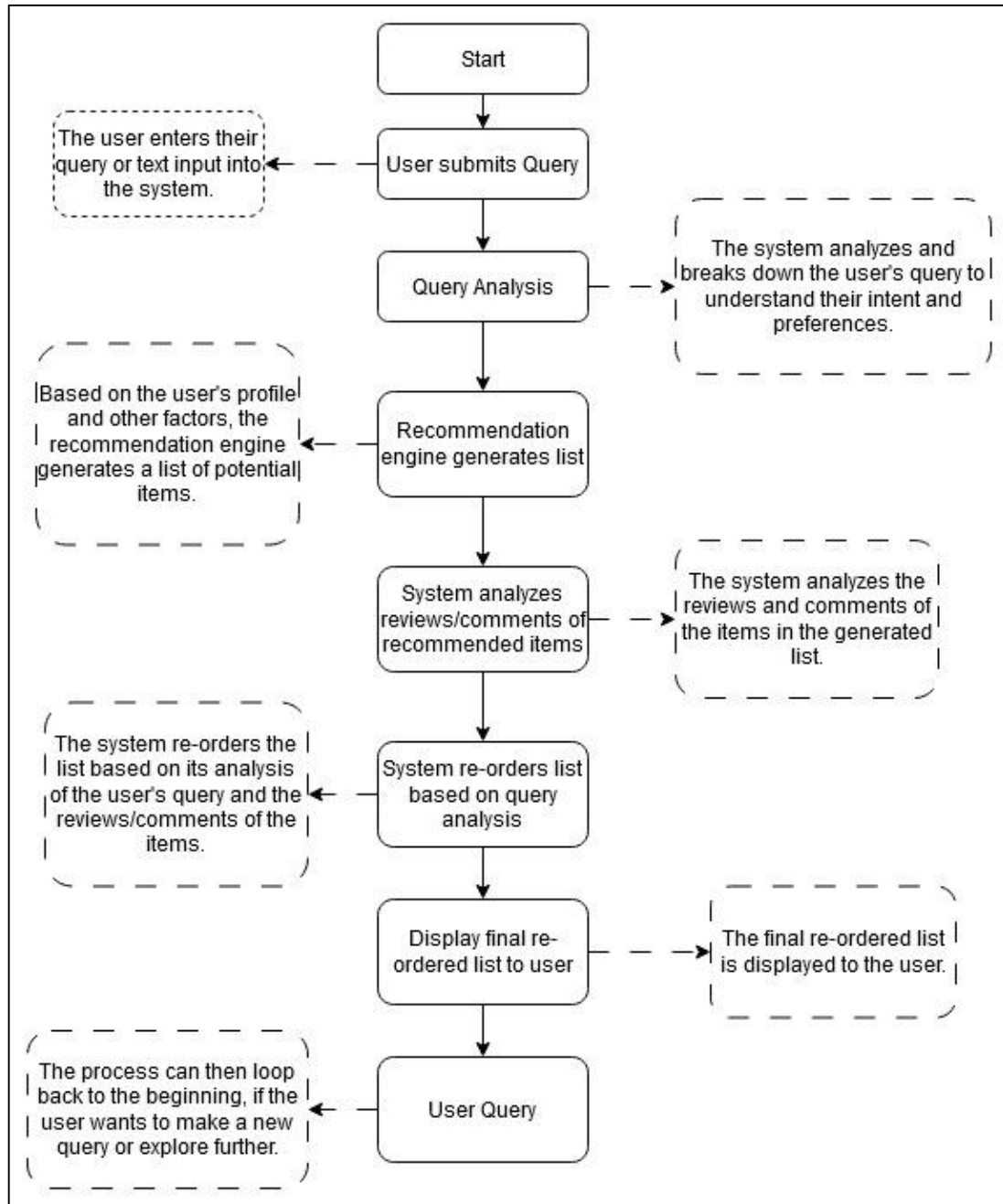


Figure 3.1: Proposed Method Flowchart

This flowchart illustrates the multi-criteria recommendation system designed to deliver highly personalized and relevant results. The system begins by carefully analyzing user preferences and behaviors. Based on this understanding, the recommendation engine generates an initial list of suggested items.

The unique aspect of this system lies in its ability to refine recommendations further. It delves into user queries, breaking them down to understand the specific needs and interests expressed. Additionally, it intelligently analyzes reviews and comments associated with the recommended items. Using this combined analysis, the system re-orders the initial list, prioritizing those items that best match the user's query.

The final output is a dynamically tailored list of recommendations. This approach ensures that the most relevant options are presented to the user, leading to a significantly enhanced and personalized user experience.

The proposed method offers significant advantages in terms of both computational efficiency and recommendation relevance. Imagine a vast system with 1,000 items. If the system were to analyze every item's reviews and comments based on every user query, it would be an incredibly time-consuming and resource-intensive process.

Fortunately, the proposed method employs a two-pronged approach to address this challenge. First, the recommendation engine steps in, narrowing down the potential options by suggesting only 100 items most likely to be relevant to the specific user. This initial filtering significantly reduces the number of reviews and comments the system needs to analyze based on the user's query.

This targeted approach not only translates to faster results and reduced computational strain but also leads to increased recommendation relevancy. By focusing on a smaller set of relevant items, the system can delve deeper, analyzing the reviews and comments with greater context. This allows the system to identify nuances and trends that might be missed in a broader search, ultimately leading to recommendations that are highly personalized and tailored to the user's unique needs and preferences.

To sum up, by utilizing the recommendation engine as the pre-filtering, it reduces the computational burden while simultaneously enhancing the quality and accuracy of the recommendations presented to the user. This results in a more streamlined and efficient user experience, allowing them to discover the perfect item without sacrificing valuable time or resources.

In the upcoming sections, there will be closer look at how the proposed method works. It will start by showing you a picture of the architecture. Then, each part will be examined and explained what it does in the system. This detailed explanation will help you understand how the method works and what makes it effective.

3.3 System Architecture

The proposed method not only should capture hidden patterns in data, but also it should be able to be responsive to user dynamic preferences. Therefore, this study came up with a solution that consists of two main components which are recommendation engine and user query analysis. The mechanism is that the first recommendation system lists top related items according to the user history. After that, the users need to give queries according to their preferences. Then, the system checks with matching words of the user query and re-orders the recommendation list accordingly. As an example, a user may ask the system for a jogging place on Wednesday and ask for a romantic date on Friday. This user preference changes in two days. The user's queries will be saved for future user mining and preferences. The diagram below shows the architecture of proposed method which has two main components.

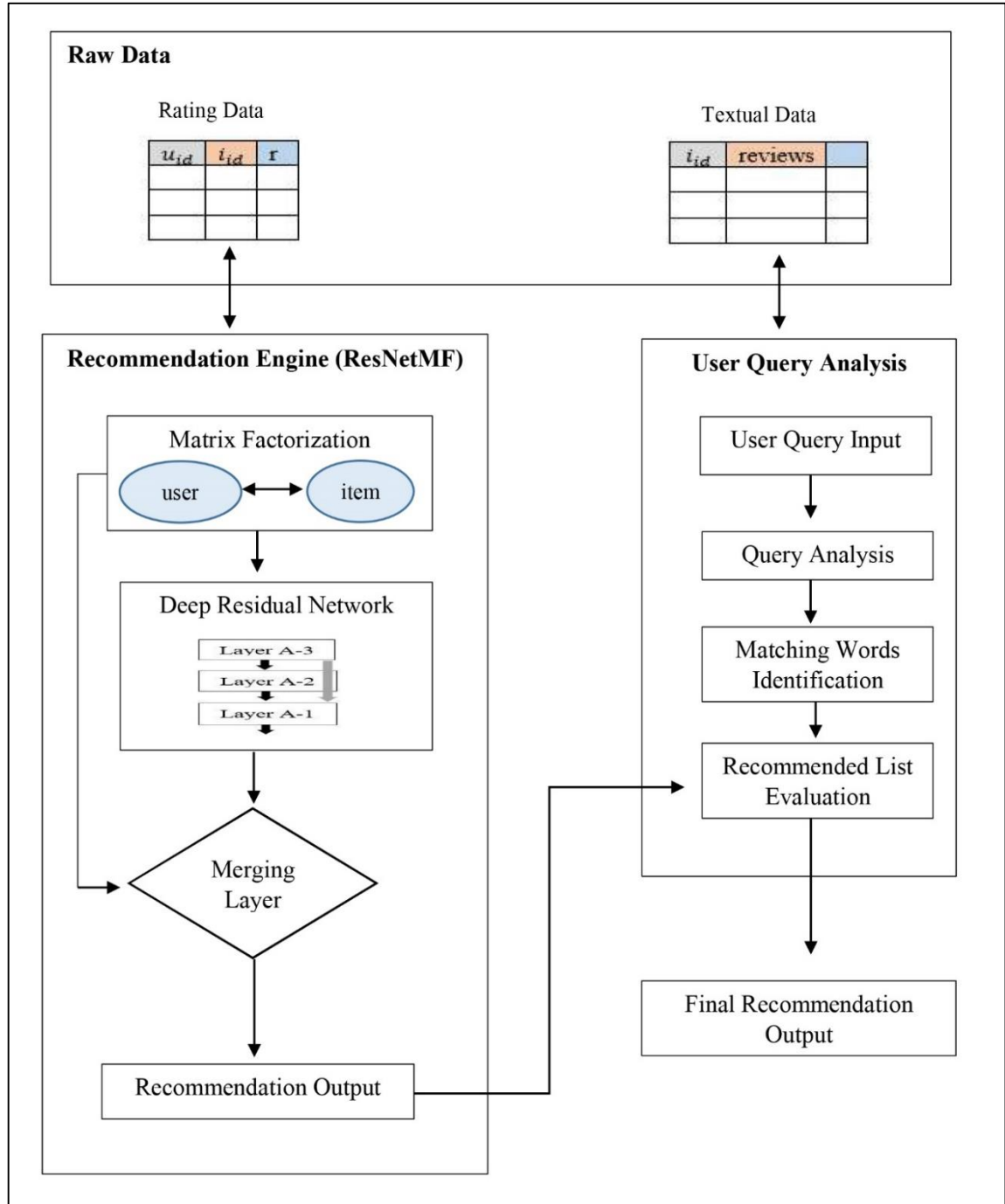


Figure 3.2: System Architecture

The first component is the recommendation engine. This study proposed a recommendation method named ResNetMF, that synergistically combines state-of-the-art matrix factorization and deep residual neural networks to detect non-linear patterns in the

data. Deep residual networks, originally developed for image recognition tasks, are a variant of CNN.

The proposed recommendation engine is a fusion of matrix factorization and deep residual networks to address the limitations of matrix factorization, which excels with linear data but struggles with non-linear patterns; by harnessing the strengths of both approaches, the objective is to learn the underlying data features more accurately and handle non-linear patterns effectively. Deep residual networks are adept at recognizing intricate patterns in images and can capture non-linear patterns, complementing the capabilities of matrix factorization.

The core idea behind ResNetMF is to design a neural network with multiple branches, each dedicated to a specific task. This architecture allows the network to simultaneously leverage the power of matrix factorization and deep residual neural networks, enabling it to capture both linear and non-linear patterns in the data.

In the ResNetMF model, different branches are constructed with distinct architectures to extract specific features from the input data. Some branches are designed to identify linear patterns using matrix factorization, which is known to work well with linear data. Matrix factorization decomposes the data into latent factors and learns the linear relationships between users and items. On the other hand, other branches utilize deep residual neural networks, which excel at capturing complex non-linear patterns, especially in image recognition tasks. These networks are convolutional neural network (CNN) variants and have shown remarkable success in uncovering intricate features in visual data.

By combining these different branches, the network aims to harness the complementary strengths of both approaches. Matrix factorization focuses on capturing

linear relationships, while deep residual networks specialize in detecting non-linear patterns. This integration allows the network to handle both patterns effectively, resulting in a more comprehensive understanding of the underlying data.

Instead of directly predicting the target score, the neural network aims to estimate the error introduced by the matrix factorization model. This shift in focus is motivated by the insight that improving the accuracy of the matrix factorization model can be achieved by learning and predicting the discrepancies or residuals between the predicted and actual scores. By modeling and predicting these errors, the network can refine the predictions made by the matrix factorization model, leading to more accurate recommendations.

An additional merging layer is introduced to merge the outputs of these two components. This layer combines the predictions from the matrix factorization model and the deep neural network, producing the final recommendation output. The merging layer uses summation and concatenation.

The second component of the system takes the initial recommendations from the ResNetMF engine and refines them further based on the user's specific query. This component, known as user query analysis, plays a crucial role in ensuring the recommendations are truly relevant to the user's needs.

Imagine you're browsing for running shoes after receiving a recommendation list. User query analysis kicks in here. The system starts by understanding your query. It breaks down your text input into individual words, capturing the essence of your search. It then delves into the reviews and descriptions associated with the recommended running shoes on the list.

By analyzing these texts, the system identifies matches between your query and the information associated with each shoe. The shoe with the highest number of matches is then prioritized and displayed prominently on the list. This ensures that the recommendations you see are not just generic suggestions, but specifically tailored to the nuances of your search.

User query analysis acts as a filtering mechanism, taking the initial recommendations and refining them further based on your specific needs and preferences. This additional layer of analysis ensures that the final list presented to you offers the most relevant and personalized options, ultimately leading to a more satisfying user experience.

Each components of the proposed method with their sub-components are explained in details in the following sections. First, the recommendation engine and its sub-component are explained following the processes of query analysis are explained in detail.

3.4 Recommendation Engine (ResNetMF)

This study proposes a multi-criteria recommendation system that leverages various item attributes beyond ratings to improve accuracy, uncover users' hidden preferences, and adapt to their evolving preferences over time.

To achieve this, the study employs a deep learning approach combining matrix factorization. Matrix factorization excels compared to traditional collaborative filtering by representing users and items through latent factors derived from user-item rating patterns (M. Wang, 2020). However, it struggles with additional data in multi-criteria scenario (M. Wang, 2020).

Therefore, this study proposes a multi-criteria recommendation system based on matrix factorization and deep residual networks (ResNetMF) to enhance recommendation

accuracy. ResNetMF incorporates a deep residual network, a deep learning method, with matrix factorization. This approach simplifies the training of neural networks and allows them to delve deeper into the data.

In this study the recommendation algorithm named as ResNetMF, where "MF" represents matrix factorization, "ResNet" refers to the residual neural network, and "R" denotes the recommendation list. The relationship between the components is mathematically represented by Equation 3.1, where R' is the sum of the matrix factorization output (MF(x)) and the ResNet output (ResNet(x)). Equation 3.2 defines ResNet(x) as the difference between R' and MF(x), indicating the error introduced by the matrix factorization model.

$$R' = MF(x) + Resnet(X) \quad \text{Equation 3.1}$$

$$ResNet(x) = R' - MF(x) = \text{error made by MF model} \quad \text{Equation 3.2}$$

The ResNetMF model offers a comprehensive framework that leverages matrix factorization and deep residual neural networks to capture linear and non-linear data patterns. By integrating these approaches, the model improves the accuracy of recommendations by refining the predictions made by the matrix factorization model using the insights captured by the deep neural network. This combination of linear and non-linear modeling enhances the network's ability to understand complex relationships in the data, leading to more effective and precise recommendations.

The subsequent sections of this study will delve into the intricate details of the proposed recommendation model's components, which includes explaining the core notations used, followed by an in-depth exploration of the role of matrix factorization and a

comprehensive discussion on the functions of deep residual networks within the proposed method.

3.4.1 Notations

The notation introduced in the text describes algorithms in recommender systems, which aim to assign scores to items based on user ratings. The notation starts by defining item 'b' score as $S(b)$. In a dataset with X items, 'b' can take values from 1 to X , representing each item in the system.

To calculate the average rating for an item 'b', denoted as B , the notation incorporates the sum of all ratings for item 'b' (represented as $|O_b|$, where $|O_b|$ is the set of all users who rated item 'b'). The sum is divided by the number of ratings for item 'b'. This provides a measure of the overall rating received by item 'b' based on user feedback.

$$S(b) = \frac{\sum_{a \in O_b} T_{ab}}{|O_b|} \quad \text{Equation 3.3}$$

$|O_b|$ = set of all users who rated item b

T_{ab} = rating user a gave to item b

However, to personalize the scores and consider the preferences of individual users, the notation is extended to include scores that depend on users 'a' and items 'b'. This is denoted as $S(a, b)$, where 'a' represents users and 'b' represents items. In the context of recommender systems, 'a' could represent different users in the system, and 'b' could represent various items such as tourism destinations, hotels, chairs, movies, music, or any other recommended items.

In this extended notation, 'a' ranges from 1 to Y, where Y represents the total number of users in the system. Similarly, 'b' ranges from 1 to X, representing the number of items. The rating given by user 'a' to item 'b' is denoted as T_{ab} . By incorporating the personalized aspect, the scores are now influenced by both the user and the specific item being rated.

$$S(a, b) = \frac{\sum_{a \in O_b} T_{ab}}{|O_b|} \quad \text{Equation 3.4}$$

$a = 1 \dots Y$, Y = number of users

$b = 1 \dots X$, X = number of items

By utilizing this notation, recommender systems can capture the preferences and behaviors of individual users, resulting in personalized item scores ($S(a, b)$). The expanded notation allows for a more nuanced understanding of user-item interactions and facilitates the generation of tailored recommendations based on user-specific preferences.

The notation provides a framework for understanding the relationships between users, items, ratings, and scores in recommender systems. It enables the design and development of algorithms considering global patterns and individual user preferences, leading to more accurate and personalized recommendations.

3.4.2 The Core Object of Study

As discussed earlier, the T_{ab} variable represents the rating given by user 'a' to item 'b', where 'a' ranges from 1 to Y and 'b' ranges from 1 to X. This information is typically organized in the form of a user-item matrix, denoted as $R_{Y \times X}$. This matrix is often called the user-item rating matrix or simply the user-item matrix, with dimensions of (Y×X).

For example, consider a movie recommendation system with a user-item matrix of size 1000×5000 . Each row in the matrix represents a unique user, and each column represents a specific movie. The entries in the matrix correspond to the ratings users give to movies they have watched. A simplified representation of this matrix is shown in Table 3.2.

Table 3.1: A Sample of User-Item Matrix

	Movie 1	Movie 2	Movie 3	Movie 4		Movie 5000
User 1	4	3	5	2		2
User 2		5	2	4		3
User 3	3	1	2			
.....						
User 1000	2	4	?	5		

In recommender systems, the user-item matrix is often sparse, meaning many entries must be filled in. This sparsity arises because not all users have rated all movies; each user has only provided ratings for a subset of the available movies. This sparse nature is evident in the example table above, where gaps represent missing ratings.

The primary objective of recommendation systems is to estimate the missing entries in the user-item matrix, enabling the system to make personalized user recommendations. By predicting the values of these missing entries, the system can fill in the gaps and suggest movies that users are likely to enjoy. For instance, if the system predicts that a user would rate a particular movie highly, it can recommend it to the user, even if they have not seen it before. By accurately estimating the missing ratings in the user-item matrix, recommendation systems can provide users with tailored and relevant movie recommendations, enhancing their movie-watching experiences.

3.4.3 Matrix Factorization

The proposed recommendation model utilizes matrix factorization, a distinct approach from other supervised learning algorithms, and operates like unsupervised learning methods. While supervised machine learning algorithms prioritize accuracy, matrix factorization focuses on representing a rating matrix (R) as a product of two smaller matrices (P and V) to approximate the original matrix.

Matrix factorization involves breaking down the rating matrix (R) into two matrices: P , representing users, and V , representing items. The matrix R is a sparse ($Y \times X$) matrix, where Y represents the number of users, and X represents the number of items. Since most users have not rated most items, the matrix is mostly filled with zeros. An example of a sample user and item matrix is provided in Table 3.3, showcasing the ratings given by users for different items.

Table 3.2: Sample User and Item Matrix

	Item 1	Item 2	Item 3	Item 4	Item 5
User 1	0	0	3	0	0
User 2	5	0	0	3	4
User 3	2	0	5	0	0
User 4	0	3	0	1	0

Factorization is a concept that involves expressing numbers as the product of other numbers. For instance, 10 can be represented as 5×2 . Breaking down numbers into constituent factors allows for a better understanding of their composition and relationships.

Matrix factorization builds upon this idea by representing the larger matrix R as the product of two smaller matrices, P and V . In other words, R can be expressed as $R' = PV$.

$$\mathbf{R}' = \mathbf{P} \mathbf{V}$$

Figure 3.3: Matrix Factorization Logic

The notation $\mathbf{R}'$ is used to represent the model of $\mathbf{R}$, as it is an approximation of $\mathbf{R}$ and does not provide the true values. The key point to understand about matrix factorization is that the sizes of matrices $\mathbf{P}$ and $\mathbf{V}$ are significantly smaller than the size of $\mathbf{R}$. Specifically, the matrix $\mathbf{R}$ has a size of $(Y \times X)$.

The study tackles memory limitations associated with large datasets by avoiding storing the entire matrix in memory. This is crucial, as storing a complete matrix for datasets like MovieLens (280,000 users, 58,000 movies) would be impractical. Instead, the proposed method, ResNetMF, selectively retains only the necessary portion for calculations.

Matrix factorization addresses this challenge by creating two significantly smaller matrices, $\mathbf{P}$ and $\mathbf{V}$, which capture the essential information from the original large matrix. These matrices offer a more efficient representation, significantly reducing memory usage compared to traditional methods.

The size of $\mathbf{P}$ and $\mathbf{V}$ is determined by the number of users (Y), items (X), and a chosen value (K) representing the number of features. For instance, with $K = 10$, $Y = 280,000$, and $X = 58,000$, the combined size of $\mathbf{P}$ and $\mathbf{V}$ ($Y \times K + X \times K$) is 3,380,000, significantly smaller than the original matrix size (16,240,000,000). This size reduction not only improves memory usage but also reduces computational requirements.

While the calculation demonstrates the size reduction, it's important to remember that the model doesn't multiply the entire matrices for predictions. Instead, it focuses on individual user-item recommendations, predicting the rating of a specific user for a specific item.

Consequently, the calculation of R' is less sparse due to using smaller matrices, as it simply involves the dot product of two vectors of size K . The following equation represents this approach and will also be utilized in coding.

$$R'_{ab} = R' [a, b] = P[a] \cdot V[b] = P[a] \cdot V[b] \quad \text{Equation 3.5}$$

Matrix factorization is a powerful approach for uncovering patterns and correlations within a matrix of user-item ratings. By decomposing the original rating matrix into two low-rank matrices, one for users and one for items, the model can approximate the original matrix by multiplying these two matrices together. This process enables the discovery of latent features or factors influencing user ratings without explicit labeling or identification.

For instance, in a movie recommendation system, these latent features could represent various aspects such as genre, actors, director, tone, and setting. Although not explicitly defined, the model can deduce these features by analyzing the ratings provided by users for different movies. As more data becomes available, the model becomes more adept at accurately capturing and representing these latent features.

To illustrate the mechanics of the matrix factorization model, let us consider a specific example. Suppose there is a movie ratings dataset from the MovieLens dataset that aims to predict a user's preference for a particular movie. Each element in a user's vector can represent a different movie genre, such as action, adventure, comedy, romance, horror, and

animation. Similarly, each element in a movie's vector represents the presence or intensity of each genre in the movie.

For example, user "A" enjoys action and adventure movies, comedy, and animation but dislikes romance movies. This user's preferences can be represented using a vector $P[a] = (1, 0.8, -1, 0.1, 1)$, where each element denotes their preference for a specific genre. Similarly, suppose the movie "B" contains substantial comedy, action, and animation but lacks romance. The features of this movie using a vector $V[b] = (1, 1.5, -1.3, 0, 1.2)$ can be represented, where each element represents the presence of a genre.

To predict how much user "A" would like movie "B," it simply calculates the dot product of the two vectors: $(1 * 1) + (0.8 * 1.5) + (1 * 1.3) + (0.1 * 0) + (1 * 1.2) = 4.7$. This resulting score indicates a high likelihood that user "A" will enjoy the movie. Based on this prediction, the system can recommend movie "B" to the user "A."

In summary, matrix factorization represents users and items as vectors of features, the model captures the essential factors that influence user preferences and accurately predicts their level of enjoyment for specific items. The model autonomously discovers these latent features by analyzing the data patterns, making it a powerful unsupervised learning approach. Unlike supervised machine learning algorithms that require explicit feature engineering, matrix factorization automatically uncovers meaningful features solely from the patterns in user-item ratings.

The features are not explicitly defined or clear-cut in a real Matrix Factorization model. For example, it is impossible to directly select a feature like "comedy" without inspecting and understanding its meaning within the data context. The model behaves more like unsupervised learning, examining the data and identifying patterns. This process is

similar to clustering, where the model can identify groups of data points that share similar characteristics.

The ability of matrix factorization to discover latent features solely from the relationships between users, items, and ratings makes it suitable for situations where input features are not well-defined or when the data is complex and manual feature engineering is challenging. The model can handle unstructured or high-dimensional data and uncover meaningful patterns that may need to be apparent to human experts.

In contrast to supervised machine learning approaches, where explicit features about users and items are required to train models like a random forest for predicting user ratings, Matrix Factorization does not rely on feeding such features into the model. Instead, it autonomously discovers these features by analyzing the relationships between users, items, and their ratings. This powerful capability allows the model to uncover meaningful features solely by examining the data without relying on predefined feature engineering. This capability makes it a valuable tool for various tasks, including user preference prediction and personalized recommendations.

3.4.4 Residual Neural Network

Recommender systems have greatly benefited from the advancements in neural networks. Neural networks enable the extraction of non-linear relationships and their incorporation into recommendation algorithms, leading to more powerful predictive capabilities and improved user experiences.

Deep residual learning, introduced by Kaiming He in 2015, is a deep learning technique primarily used in computer vision tasks. Residual neural network (ResNet) is a

powerful convolutional neural network that emerged from this approach and achieved remarkable performance in the ImageNet challenge. ImageNet is an open-source repository containing over 14 million images across over 21,000 categories. The residual neural network achieved powerful performance, with around 3.57% error which is incredible. ResNet's ability to stack many layers (up to 152) addresses the vanishing gradient issue, which hindered the performance of deep networks in the past.

The primary concept of ResNet involves the ability to stack many layers, possibly up to 152, while also addressing the vanishing gradient issue. Before the development of ResNet, adding more layers to a deep learning network often resulted in the vanishing gradient problem. The gradient is used to return to the network and update its weights. One of the challenges is that once all these layers stack on top of each other, the gradient becomes very small, and essentially, network performance becomes extremely poor. Moreover, the network cannot train anymore if there is a very deep convolutional neural network. In simple terms, ResNet allows stacking up to 152 layers while avoiding the vanishing gradient problem, which occurs when adding too many layers to a deep learning network. The gradient becomes too small, resulting in poor network performance and difficulty in training deep convolutional neural networks.

The ResNet overall idea is to use skip connection. The figure below is a visual representation of a network. The input feed to layer A-1, and there is also a skip connection. Moreover, it skips the convolution layer of A-1 and passes along to the next layer, which is Layer A, and it feeds the same input to Layer A (Skip connection is in the gray arrow in the figure below). Therefore, it can overcome vanishing gradient issues and stack hundreds of layers on top of each other. The residual connections in the neural network architecture allow

gradients to flow more directly through the network. These skip connections enable the network to learn residual mappings, which makes it easier for the gradients to propagate through the network during backpropagation.

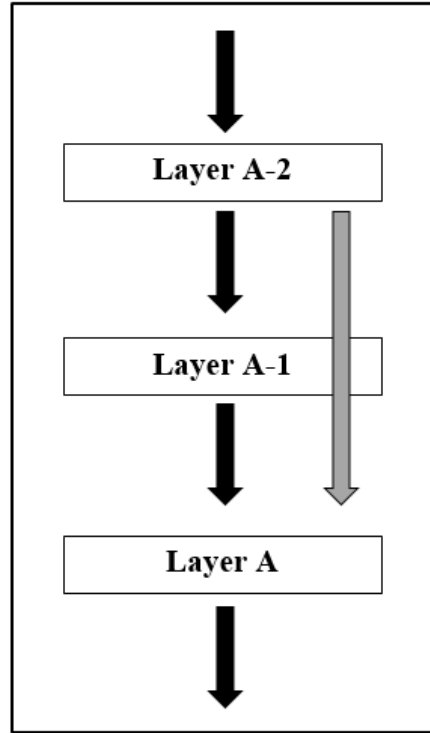


Figure 3.4: Deep Residual Neural Network Concept

Traditional recommender systems often rely on a single hidden layer of neurons to process user ratings, which may not accurately capture users' non-linear preferences. Therefore, to address this limitation, a proposed method involves building a multiple-branching deep neural network as a differentiable extension of matrix factorization-based recommendation systems. This framework enhances prediction accuracy by considering residual information beyond the single layer of neurons.

The input in a feed-forward neural network like ResNetMF consists of feature vectors. The ResNetMF eliminates the need for extensive data collection by utilizing matrix

factorization embedding. User and item feature vectors containing the necessary information are already prepared using matrix factorization, which decomposes a large, sparse matrix into smaller dense matrices (P and V) representing user and item latent features.

ResNetMF combines user and item latent features ($P[a]$ and $V[b]$) to form a single input feature vector. This vector is then inputted into a neural network consisting of convolutional and fully connected layers. By learning complex relationships between user and item features, ResNetMF improves the accuracy of predictions and overall performance in recommendation systems. Additionally, matrix factorization embedding eliminates the need for extensive data gathering, making the implementation of ResNetMF more efficient and practical.

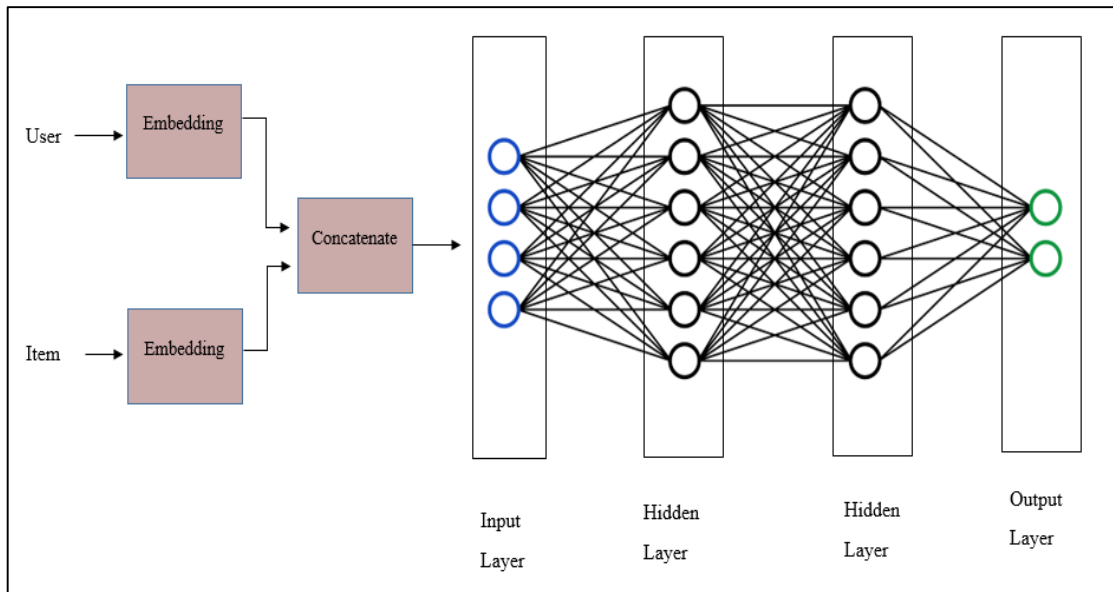


Figure 3.5: ResNet on Proposed System

ResNetMF leverages matrix factorization embedding to prepare user and item features, eliminating the need for extensive data collection. Matrix factorization decomposes a large, sparse matrix into two smaller, dense matrices representing latent user and item

features. These features capture user and item characteristics and serve as input for ResNetMF.

ResNetMF employs a distinct approach, merging latent features into a single input vector fed into a neural network with convolutional and fully connected layers. This allows the network to learn complex relationships between user and item features, significantly improving recommendation accuracy.

Through learning these non-linear relationships, the neural network embedded within ResNetMF generates more accurate predictions. Additionally, matrix factorization embedding eliminates the challenges of gathering extensive user and item data, making ResNetMF practical.

ResNetMF combines the strengths of neural networks and matrix factorization. This integration significantly enhances prediction accuracy and overall recommendation system performance. Furthermore, ResNetMF's implementation is efficient due to not requiring extensive data gathering.

To summarize, ResNetMF's key strengths lie in its ability to capture complex user-item relationships through a neural network and its use of matrix factorization embedding, which eliminates extensive data collection and leverages informative latent features. This combination leads to higher prediction accuracy, improved overall performance, and efficient training, making ResNetMF an effective and practical solution for recommendation systems.

3.4.5 Overall Mechanism of Proposed Recommendation Method (ResNetMF)

The proposed approach can be categorized as a feature augmentation type of hybrid recommendation system, utilizing the strengths of different algorithms to enhance recommendation accuracy. It combines matrix factorization, discovering hidden, latent factors in the data, with a residual neural network to predict user preferences.

The proposed recommendation system does not require manual data collection, as it uncovers these explicit factors by searching for hidden, latent factors within the data. Matrix factorization is employed to identify linear data patterns and create feature vectors representing users and items. These resulting vectors are then fed into the residual neural network. Matrix factorization excels at capturing collaborative filtering patterns and latent factors within user-item interactions. Conversely, ResNet thrives in learning complex, non-linear relationships from additional user and item features.

By combining these techniques, the proposed system takes a two-step approach. First, it employs matrix factorization to decompose the user-item matrix into lower-dimensional user and item embedding matrices. These matrices capture the essence of user-item interactions. Subsequently, the system leverages a ResNet model that utilizes the latent features obtained from matrix factorization as input. This ResNet model learns to map these features to the desired recommendations.

The model is designed to uncover previously inconceivable relationships in the underlying data using a residual neural network. The residual neural network facilitates the generation of additional features that describe observations better than the original data. These additional features aid in determining which items should be added or removed from the top recommendation list based on user preferences.

The combined architecture of matrix factorization and residual network for recommendation tasks consists of two main components:

Matrix Factorization Component:

- i. User Latent Factors (U): A matrix that captures the user preferences, with dimensions (number of users, number of latent factors).
- ii. Item Latent Factors (I): A matrix that captures the item characteristics, with dimensions (number of items, number of latent factors).

Residual Network Component:

- i. Input Layer: Accepts the input data, which could be user and item features or other contextual information.
- ii. Feature Extraction Layers: Multiple layers of convolutional, pooling, and activation operations to learn high-level features from the input data.
- iii. Residual Blocks: Stacked residual blocks consisting of convolutional layers and skip connections to capture residual information and enable deeper learning.
- iv. Global Pooling Layer: Aggregates the features across spatial dimensions and generates a fixed-length feature vector.
- v. Fully Connected Layers: One or more fully connected layers to transform the extracted features.
- vi. Output Layer: Produces the final recommendation scores or ratings.

The proposed approach is summarized by the equation 3.6. Here, R represents the predicted recommendation score, U and I signify the user and item embedding matrices from matrix factorization, and $F(X)$ denotes the learned features extracted by the ResNet model from the input features (X)

$$R = U \times I + F(X) \quad \text{Equation 3.6}$$

The dot product operation ($\times$) between U and I computes the interaction between user and item embeddings. In contrast, the addition operation ($+$) combines this matrix factorization interaction with the features learned by the residual network. This combined approach allows for extracting meaningful patterns from the data through matrix factorization and learning complex relationships and additional features through the residual network. By integrating these components, the final recommendation score is obtained, providing enhanced accuracy in recommendation systems.

By leveraging both matrix factorization and ResNet, the system can extract meaningful patterns from data and capture complex relationships, leading to enhanced recommendation accuracy.

In conclusion, the combined architecture of matrix factorization and residual network offers a robust framework for recommendation systems. By incorporating feature augmentation, this approach significantly enhances prediction accuracy and enables efficient model training. This translates to personalized and relevant recommendations across various domains.

3.5 User Query Analysis

This section explores the user query analysis part, a crucial process for selecting and providing useful information based on users' queries.

In the proposed method, the emphasis is on analyzing text input from users. Within this section, the system carefully evaluates user queries and retrieves relevant data based on their specific needs. It is important to note that the recommendation system lies at the heart of the overall system, and this section solely focuses on selecting information from the recommended list.

The selection process follows: First, the user query undergoes analysis. Subsequently, the system breaks down the query into individual words. Next, if available, the system examines the reviews and descriptions associated with the recommended list. Finally, the system disseminates the item with the most matches to the user, aligning with their query.

Let us consider a user query such as "Luxury Clothes Shopping." In the recommendation system, the initial step involves generating a list of 10 items based on the user's criteria. The query is subsequently broken down into four distinct parts: "Luxury," "Clothes," "Shopping," and "Luxury Clothes Shopping." The system then searches the reviews of these recommended items to identify matches and calculates the total count of occurrences. Ultimately, the item with the highest match score is selected and disseminated to the user.

Here's an example scenario illustrating this process:

User Query: "Luxury Clothes Shopping"

Step 1: The recommendation system generates a list of 10 items based on the user's criteria.

Step 2: The query is broken down into separate words: "Luxury," "Clothes," "Shopping."

Step 3: The system examines the reviews of each item from the generated recommendation list, comparing them to the query words. The table below presents the word counts for each item:

Table 3.3: Case Scenario Example

Items	Item 1 (Word Count)	Item 2 (Word Count)	Item 3 (Word Count)
Luxury Clothes Shopping	1	0	1
Luxury	1	0	3
Clothes	3	2	5
Shopping	5	6	7
Total Word Count	10	8	16

Step 4: The item with the highest number of word matches, in this example, Item 3, is selected and disseminated to the user as the top recommended item.

In the example provided, it is noteworthy that despite item 3 being ranked third on the recommendation list, it is still disseminated to the user due to its highest number of matches. This matching process offers a significant advantage in terms of efficiency, as it avoids the need to scan the entire database. Instead, the system focuses on searching within a smaller set of examples generated by the recommendation system, specifically tailored to the user's specified criteria.

Three distinct scenarios can arise during the item selection, each with implications. The first scenario occurs when a single item stands out with the most matches, as illustrated in the example above. This item is disseminated to the user in such cases, aligning with their query intent and maximizing relevance.

Table 3.4: First Scenario Example

Recommendation Order	Item	A	B	C	D	E	F	G	H	I	J
	Word count	10	7	16	5	12	9	8	6	2	4

In the given example, the system analyzes the query when the user query is "Luxury Clothes Shopping" and generates a list of 10 recommendations. It counts the matches for each item in the list. As a result, the recommended order is modified. In this case, the item with the most matches, item C, is disseminated to the user as the top option. This change in recommendation ordering is based on the matching process, which considers the relevance of each item to the user query, ensuring that the most suitable options are presented first.

Table 3.5: First Scenario new Ordering

New Recommendation Order	Items									
	C	E	A	F	G	B	H	D	J	I

The second scenario arises when two or more items share equal matches. When multiple items score equally in a user query, the system prioritizes the item with the higher rank on the recommendation list. For instance, if the user searches for "Luxury Clothes Shopping" and the system generates 10 recommendations, items E and J might have an equal

number of matches. However, if item E occupies a higher position on the list, it will be prioritized and presented to the user. This prioritization ensures that the most relevant and well-ranked item is delivered, enhancing the system's accuracy and effectiveness.

Table 3.6: Second Scenario Example

Recommendation Order	Item	A	B	C	D	E	F	G	H	I	J
	Word count	10	8	12	5	16	9	8	6	2	16

By prioritizing higher-ranked items, the system acknowledges the importance of the recommendation list's order, ultimately leading to user satisfaction and improved overall performance in delivering personalized and accurate recommendations.

Table 3.7: Second Scenario new Ordering

New Recommendation Order	Items									
	E	J	C	A	F	B	G	H	D	I

The third scenario occurs when the system fails to find matching words within its database or when all items in the recommendation list have equal word repetitions for the given user query. In both cases, the system prioritizes user experience by disseminating the first item in the recommendation list as a default choice. This ensures the user receives some information and has a starting point for exploration, even if a perfect match cannot be made. This approach enhances the system's ability to cater to a wide range of user queries and preferences, contributing to a more comprehensive user experience.

Table 3.8: Third Scenario Example

Recommendation Order	Item	A	B	C	D	E	F	G	H	I	J
	Word count	0	0	0	0	0	0	0	0	0	0

In Case 3, For example, if the user searches for "Luxury Clothes Shopping" and the system cannot find any matching words (scenario 1) or all items have an equal number of matches (scenario 2), the first item on the list (item A) will be presented to the user. This ensures the user receives a recommendation, even without a perfect match.

The recommendation list is crucial in guiding the system to determine which information should be disseminated to the user. It goes beyond the mere suggestion of a generic set of items. The recommendation list is carefully curated based on various factors, such as user preferences, historical behavior, demographic information, and contextual relevance. The system can provide each user with a personalized and tailored experience by incorporating these elements.

A key advantage lies in the ability to dynamically reorder results. Sophisticated algorithms prioritize items likely to resonate with the user's specific needs, making them more prominent and increasing engagement. This focused approach enhances efficiency by eliminating generic suggestions and prioritizing relevant options based on user criteria. It leads to higher user satisfaction as they encounter recommendations directly aligned with their interests, increasing the likelihood of them finding the information they seek and engaging with the system more frequently.

It is worth noting that the continuous refinement and evaluation of the recommendation system are essential to maintain its performance and adapt to evolving user needs. Regularly collecting user feedback, analyzing usage patterns, and incorporating new data sources and algorithms are crucial to ensure the system remains accurate, reliable, and capable of delivering high-quality recommendations.

3.6 Conclusion

This chapter provides a comprehensive and in-depth exposition of the proposed method. It offers readers a detailed overview of the method, elucidating its underlying principles and operational mechanisms. Each constituent element of the method is meticulously expounded upon, accompanied by thoroughly exploring the algorithms employed. Furthermore, the chapter meticulously examines the notations and formulas utilized throughout the implementation process. In particular, the user query analysis and ResNetMF process are elucidated precisely, shedding light on its intricacies and procedures.

The subsequent chapters will focus on the development, results, analysis, and comparative evaluation of the proposed method to other existing approaches. Special attention will be dedicated to the experimental design process, presenting the advancements made, followed by an in-depth analysis of the results. This will facilitate a comprehensive assessment of the effectiveness and utility of the proposed method.

CHAPTER 4

EXPERIMENTAL DESIGN

4.1 Introduction

This chapter outlines the experimental design to evaluate the effectiveness of the proposed multi-criteria recommendation method using ResNetMF. Building on its unique combination of residual networks and matrix factorization, the aim is to demonstrate its real-world impact through rigorous testing. The methodology will encompass defining relevant evaluation metrics, establishing appropriate baseline approaches, finding the right datasets, and configuring the experimental setup. This detailed approach ensures transparency, reproducibility, and robust validation of this study findings. Planned experiments will involve comparing ResNetMF to various baselines and evaluating its performance across different scenarios. Ultimately, the goal is to provide compelling evidence of the method's ability to deliver both accurate recommendations and an enhanced user experience.

4.2 Experimental Design and Benchmarking

To systematically assess the effectiveness of the proposed multi-criteria recommendation system, ResNetMF, this section details the key steps involved in the evaluation process. Here, the chosen benchmarks used to evaluate the system's performance will be defined. These benchmarks include the benchmark dataset, evaluation metrics, and baseline algorithms.

In recommendation systems, benchmarks encompass standard datasets, appropriate evaluation metrics, and baseline algorithms for performance comparison. MovieLens, a widely used dataset, serves as the benchmark dataset for this study. Accuracy metrics like

MAE and RMSE, along with classification metrics (Precision, Recall, and F1-Score), are used for evaluation. To assess performance, the proposed algorithm will be compared with deep learning methods and other established approaches like collaborative filtering and content-based filtering. Detailed explanations of each benchmark are provided throughout the chapter.

Dataset Selection: System performance is evaluated on two diverse multi-criteria datasets of the MovieLens and Tripadvisor.com crawl. MovieLens is a widely used benchmark dataset containing movie ratings, timestamps, and user demographic information. This allows us to assess ResNetMF's effectiveness in a well-established domain. Tripadvisor.com Crawl is a custom dataset crawled from Tripadvisor.com, containing user reviews, ratings, and various travel-related attributes for attraction place in Kuching City. This dataset broadens the evaluation scope by encompassing a different domain with richer multi-criteria information. By selecting diverse datasets, the aim is to ensure a comprehensive evaluation of ResNetMF's generalizability and adaptability to different recommendation scenarios.

Comparison Methods Selection: To establish a clear benchmark for performance comparison, several existing recommendation methods as baselines are selected to compare the performance of the proposed system. These methods include deep learning methods such as Convolutional Neural Networks (CNN), and collaborative filtering methods like Singular Value Decomposition (SVD). It can objectively assess its relative effectiveness and potential advancements in recommendation accuracy and personalization by comparing ResNetMF against these established methods.

Evaluation Metrics: The primary focus of the evaluation is on recommendation accuracy. To measure this, the following metrics are employed. The Mean Absolute Error (MAE) metric calculates the average absolute difference between the predicted ratings and the actual user ratings. Lower MAE values indicate higher accuracy. Root Mean Squared Error (RMSE) metric calculates the square root of the average squared difference between the predicted ratings and the actual user ratings. Lower RMSE values indicate higher accuracy. In addition to these, precision and recall metrics might be considered, which provide insights into the proportion of relevant and correctly retrieved recommendations.

Pre-Experimental Phase: Before conducting the main experiments, a pre-experimental phase is crucial to fine-tune and optimize the experimental setup. This phase involves conducting pilot experiments with various hyperparameter configurations (e.g., learning rate, number of epochs) to identify optimal settings for training the ResNetMF model. These steps ultimately contribute to a more optimized experimental setup for the main experiment. The final experimental setup is then established based on the insights gained from the pre-experimental phase. This includes data splitting, and hyperparameter selection. Data splitting is about dividing the datasets into training and testing sets to ensure the model generalizes well to unseen data. Hyperparameter selection is about utilizing the findings from pilot experiments to select the optimal hyperparameters for training.

Outlining these steps fosters transparency and reproducibility in the evaluation process. By following this comprehensive experimental design, the study aims to conduct a rigorous and systematic evaluation of ResNetMF. This will enable meaningful comparisons with existing methods and provide concrete evidence regarding its effectiveness in enhancing recommendation accuracy and personalization.

4.3 Datasets

The initial goal of this project was to develop a method for sharing intelligence information customized for Sarawak state. To achieve this, a comprehensive dataset from TripAdvisor, encompassing reviews, comments, attraction descriptions, and responses derived from meticulously crafted questionnaires specifically targeted at the Sarawakian population was curated. However, due to the unexpected COVID-19 outbreak, collecting data from the intended localities within Sarawak became infeasible.

Faced with this challenge, the decision was made to focus on creating a dataset for Kuching city, the capital of Sarawak. While not the originally planned location, Kuching city shares many characteristics and demographics with other urban areas within Sarawak. This strategic shift allows the project to maintain its objective of having a dataset relevant to the local population and fulfills the research goal of creating an intelligence information dissemination method that can be adapted and generalized for use throughout Sarawak.

In addition to the Kuching city data, the widely recognized MovieLens dataset, a popular choice among recommendation systems researchers, was also employed. This dataset is renowned for its extensive collection of ratings across a diverse range of movies. This is one of the most popular multi-criteria datasets for recommendation research.

While the content of the Kuching city dataset that focusing on tourism attractions in Kuching differs from the MovieLens dataset that focusing on movie recommendations, both share some underlying similarities that contribute to the chosen approach. Both datasets involve users with specific preferences, such as preferences for certain locations, types of attractions (e.g., historical sites, museums), or categories of movies (e.g., comedies, action films). Additionally, both datasets utilize similar rating scales.

Although the MovieLens dataset focuses on movie recommendations, it provides a valuable benchmark for evaluating the core functionalities of our recommendation system. The MovieLens dataset, known for its large size and standardization, enables rigorous testing of the recommendation performance and the ability to handle vast data. Analyzing our system's effectiveness on a standardized dataset like MovieLens allows us to isolate and evaluate the core recommendation mechanisms, such as user preference learning and item recommendation strategies. This evaluation is crucial because if the system can effectively learn user preferences and recommend relevant movies based on those preferences on MovieLens, it increases the likelihood that it can be adapted to learn user preferences and recommend relevant items, such as preferred tourist attractions, in the context of the Kuching city dataset. Even though the content differs, the core functionalities of understanding user preferences and recommending relevant items remain applicable.

To sum up, Using MovieLens as a benchmark allows us to assess the generalizability of the core recommendation functionalities. A system performing well on this standardized dataset is more likely to possess robust and transferable recommendation capabilities. This two-pronged approach ensures our system is effective for both the specific Sarawak intelligence information dissemination task and potentially adaptable to other domains.

4.3.1 Kuching Dataset

One of the datasets used in this study was obtained by conducting a systematic crawl of TripAdvisor, specifically targeting data related to Kuching city. This comprehensive dataset encompasses over 22,519 user ratings from the top 45 attractions in the city. These user ratings played a crucial role in training the recommendation models, providing valuable examples for the models to learn from.

Furthermore, a survey capturing the ratings of 200 Kuching residents for the same attractions was conducted and feed to the system alongside the tripadvisor dataset. The reason for rating survey is to ask random sample of this population to write queries to the system and rate their satisfaction to fulfil third objective. The study intends to have more tailored recommendation based on those user ratings before they submit any queries to the system as this study not focusing on cold start issue of new users. Therefore, the study intends to have ratings of users on attractions before they submit any queries. This means this study interested in how the system performs for users with some existing preferences. This allows the system to potentially personalize recommendations for these users even before they submit any queries. This could lead to more accurate suggestions based on their known preferences. Evaluating user satisfaction is another reason for survey rating. By having users write queries and rate their satisfaction with the recommendations, this study can assess how well the system meets user needs (fulfilling the third objective). This comprehensive data collection serves a dual purpose. Not only does it train our recommendation models, but it also provides a robust foundation for the query analysis testing phase and evaluate the third research objective.

Beyond ratings, the crawl also captured user comments, reviews, and attraction descriptions. This wealth of textual data was stored in a structured database, allowing efficient access and utilization during query analysis part 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. User comments and reviews can highlight specific aspects of attractions that might be of interest to users, allowing us to present this information alongside the core recommendations.

Table 4.1: Kuching Dataset Features and Schema

Feature	Detail
User Ratings	The rating given by users for the destination (out of 5 stars)
User Reviews	Textual reviews 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.).

In this study all the features in above table is utilized for experimentation. This comprehensive dataset serves as a valuable resource for understanding user preferences, exploring tourist attractions in Kuching city, and developing recommendation systems. However, it is important to note that the dataset cannot be shared due to usage restrictions imposed by one of the involved parties.

4.3.2 MovieLens Dataset

The MovieLens dataset, curated by the GroupLens Research lab at the University of Minnesota, is a widely recognized resource in the field of recommendation systems. Encompassing over 27 million ratings, 1.1 million tag applications, and information from 58,000 movies and 280,000 users, it offers a rich repository of diverse data, including movie ratings, tags, and user information. This data allows researchers to explore various aspects of recommendation systems and develop and evaluate new algorithms. In this study, the 27 million rating version of the MovieLens dataset is utilized to evaluate the performance of the recommendation engine, ResNetMF. Also, since the MovieLens dataset is huge, the

training speed of the ResNetMF can easily be measured with other baseline algorithms to evaluate the second objective of this study.

The MovieLens dataset consists of four files: Ratings, Movies, Tags, and Links. Each file contains data relevant to specific tasks. The schema for the MovieLens dataset is summarized in the table below.

Table 4.2: MovieLens Dataset Schema

File Name	File Description	Columns	Columns Description
Rating	This file contains user ratings for movies.	userId	Integer, unique identifier for the user
		movieId	Integer, unique identifier for the movie
		rating	Float, rating given by the user to the movie (typically on a scale from 0.5 to 5.0)
		timestamp	Integer, Unix timestamp when the rating was made
Movies	This file contains details about the movies.	movieId	Integer, unique identifier for the movie
		title	String, title of the movie
		genres	String, genres of the movie, separated by
Tags	This file contains tags provided by users for movies.	userId	Integer, unique identifier for the user
		movieId	Integer, unique identifier for the movie
		tag	String, tag applied by the user
		timestamp	Integer, Unix timestamp when the tag was applied
Links	This file provides links between MovieLens IDs and IDs in other movie databases.	movieId	Integer, unique identifier for the movie in the MovieLens database
		imdbId	Integer, unique identifier for the movie in the IMDb database
		tmdbId	Integer, unique identifier for the movie in the TMDb database

In this study, the Ratings file was used to train and evaluate recommendation systems. The Movies file provided context and metadata about the movies, which was useful for content-based filtering and data exploration. The remaining dataset files, such as Tags and Links, were not utilized in this experiment.

4.4 Evaluating ResNetMF with Diverse Recommendation Algorithms

In order to assess the effectiveness of the recommendation engine, ResNetMF, it is decided to compare it with a diverse range of well-established recommendation algorithms. These algorithms are categorized into four groups of Matrix Factorization (MF) techniques, Collaborative Filtering (CF) techniques, Content-Based (CBF) techniques, and deep learning techniques.

4.4.1 Matrix Factorization (MF) Techniques

These are established baselines for recommendation systems and provide a foundation for comparison.

SVD (Singular Value Decomposition): This fundamental technique decomposes the user-item rating matrix into two lower-dimensional matrices, capturing latent factors that explain user preferences and item characteristics (Przystupa et al., 2021). It serves as a baseline to understand how ResNetMF, which combines MF with residual networks, improves upon a basic MF technique.

SVD++ and Non-Negative Matrix Factorization (NMF): Both SVD++ and NMF are extensions of SVD that offer different advantages. SVD++ incorporates side information like timestamps or implicit ratings for improved accuracy for recommendations, allowing us to compare how ResNetMF leverages such information if available. NMF constrains the

decomposed matrices to non-negative elements, providing a different interpretability perspective compared to ResNetMF, which might not have the same constraints.

4.4.2 Collaborative Filtering (CF) Techniques

KNN-based methods (KNNbaseline, KNNbasic, KNN with means, KNNwithzscore): These techniques identify k-nearest neighbors (similar users) for each target user and recommend items that these neighbors have liked. They offer different variations in calculating similarity, allowing us to assess how ResNetMF, which leverages global latent factors, performs against local neighborhood-based recommendations.

4.4.3 Content-Based (CBF) Techniques

NormalPredictor: This basic method recommends items with the average rating across all users, providing a simple baseline to understand how ResNetMF incorporates both user preferences and item characteristics for better recommendations.

Co-clustering: This technique simultaneously clusters users and items based on their rating patterns, identifying groups with similar preferences and item characteristics (Y. Wu et al., 2016). Comparing with co-clustering allows us to evaluate if ResNetMF captures similar patterns but potentially in a more efficient and scalable manner.

4.4.4 Deep Learning Techniques

Comparing ResNetMF with these powerful methods designed to capture complex patterns in recommendation data reveals how this approach performs against current deep-learning trends.

Convolutional Neural Network (CNN): it is a type of neural network architecture can be used for recommendation tasks, especially when dealing with structured item features

(e.g., images, and text reviews) that can be processed by convolutional layers (Dudekula et al., 2023). Comparing with CNNs allows us to understand how ResNetMF performs when dealing with unstructured rating data and how it leverages residual networks for learning complex representations.

Convolutional Matrix Factorization (CMF): This demonstrates how the ResNetMF method integrates Matrix Factorization with a deep learning architecture compared to a pre-existing hybrid approach. Convolutional matrix factorization combines convolutional layers with matrix factorization, potentially capturing local and global patterns in the rating data (Chen et al., 2019).

Autoencoders: Autoencoders learn compressed representations of the data and then try to reconstruct the original data (Macias et al., 2021). Comparing these methods allows us to assess if ResNetMF, using residual networks, achieves similar benefits but potentially with a different architectural design and a more focused goal of recommendation generation.

Recurrent Neural Network (RNN): These are powerful tools for handling sequential data. In recommendation systems, it process each user interaction one after another (Liu et al., 2019). This allows them to build a hidden state, a kind of internal memory, that captures information about the user's recent choices. RNN can capture these dynamics by processing the user's interactions in order, potentially leading to more accurate recommendations that reflect evolving preferences (Liu et al., 2019).

Neural Collaborative Filtering (NCF): This approach combines matrix factorization, a well-established technique for capturing user-item relationships, with deep learning architectures (Niu et al., 2021). This allows NCF to learn complex, non-linear patterns in

recommendation data beyond simple user preferences. It can leverage factors like item attributes, or implicit feedback to create more nuanced recommendations (Niu et al., 2021).

To summarize, these diverse algorithms provide a comprehensive comparison set for ResNetMF. They encompass various baselines (MF, basic CF, CBF), alternative deep learning architectures, and different strategies for capturing user preferences and item characteristics. Analyzing ResNetMF's performance compared to these established techniques yields valuable insights into its strengths, weaknesses, and potential to outperform or complement existing recommendation methods.

4.5 Evaluation Metrics

This study aims to create a personalized and accurate recommendation system using ResNetMF. To ensure the system recommends both relevant and accurate items, the focus is on accuracy and relevancy metrics for evaluation.

The recommendation system in this study is a supervised regression problem. It is a supervised learning since the system is trained on labeled data. The labels in this case are the user ratings on a 1 to 5 scale. Accurately predicting user preferences is crucial for recommender systems. In this study, the system aspires to guess a rating value on a scale of 1 to 5. Since these ratings are continuous, a simple good/bad classification would not suffice. To address this, this study frames the evaluation as a regression problem. In regression, models learn to predict continuous values based on features. This perfectly aligns with the system's goal that is learning from user and item features to predict user ratings on a continuous scale, ultimately leading to more accurate recommendations.

Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) are selected. These metrics measure the difference between predicted and actual ratings, with lower values indicating higher accuracy that is closer predictions to user preferences (Hodson, 2022). They are commonly used for regression tasks like this study and address the challenge of handling missing entries in the dataset (Hodson, 2022). Their formulas are:

$$\text{RMSE} = \sqrt{\frac{1}{|O|} \sum_{a,b \in O} (T_{ab} - \hat{T}_{ab})^2} \quad \text{Equation 4.1}$$

$$\text{MAE} = \frac{1}{|O|} \sum_{a,b \in O} |(T_{ab} - \hat{T}_{ab})| \quad \text{Equation 4.2}$$

O = set of pairs (a, b) where user a has rated item b

Precision and Recall are other selected metrics. These metrics address the relevancy of recommendations. Precision measures the ratio of relevant items among the recommended items, ensuring the system suggests interesting items (Singh, 2021). Recall measures the proportion of relevant items recommended compared to all relevant items in the system, capturing how well the system recommends potentially useful items (Singh, 2021). F1-score combines both into a single metric.

To effectively evaluate the relevancy of our recommendations, the concept of a confusion matrix is employed. This matrix helps to categorize the system's performance by analyzing how well it aligns with user preferences for various items.

True Positive (TP) are the wins. Items that are both recommended by the system and preferred by users. They represent successful recommendations that align with user interests. False Positive (FP) are the false alarms. Items that are recommended by the system, but users do not actually prefer. False Negative (FN) are the missed opportunities. Items that are not

recommended by the system, but users actually prefer. True Negative (TN) are the correct rejections. Items that are neither recommended by the system nor preferred by users.

Table 4.3: Confusion Matrix

Actual Recommendations	Predicted Recommendations	
	Selected	Not Selected
Actual Recommendations	Selected	True-Positive (TP)
	Not Selected	False-Positive (FP)
Actual Recommendations	Selected	False-Negative (FN)
	Not Selected	True-Negative (TN)

The formulas for precision, recall, and F1-score are:

$$\text{Precision} = \text{TP} / (\text{TP} + \text{FP}) \quad \text{Equation 4.3}$$

$$\text{Recall} = \text{TP} / (\text{TP} + \text{FN}) \quad \text{Equation 4.4}$$

$$\text{F1-score} = 2 * (\text{Precision} * \text{Recall}) / (\text{Precision} + \text{Recall}) \quad \text{Equation 4.5}$$

While other accuracy metrics like Normalized Discounted Cumulative Gain (NDCG) and Mean Reciprocal Rank (MRR) exist for evaluating recommendation systems, they are not chosen for this study. This is because the system inherently sorts the recommendation list based on the user's query at the end. Therefore, focusing on accuracy and relevancy through the chosen metrics is sufficient for evaluating the overall performance of the recommendation system in this context.

In summary, this study utilizes MAE, RMSE, Precision, Recall, and F1-score to evaluate the effectiveness of the ResNetMF-based recommendation system in achieving its goal of providing both accurate and relevant recommendations to users.

4.6 Pre-Experimental Phase

This section delves into the data preparation steps undertaken to ensure clean and representative datasets for both the MovieLens and Kuching datasets used in this study. Here, common preprocessing techniques applied to both datasets, followed by additional steps specific to the MovieLens data due to its inherent characteristics are discussed. Following data preparation, the process of hyperparameter tuning for certain algorithms is explored. Hyperparameter tuning involves adjusting specific settings within these algorithms to optimize their performance for the recommendation task.

4.6.1 Data Preprocessing and Cleaning

The initial step involves preprocessing and cleaning both the MovieLens and Kuching datasets. These steps are identical for all recommendation algorithms applied and ensure a clean, representative dataset for optimal performance.

Missing Value Handling: Missing values can significantly impact the performance of recommendation algorithms. In this case, both MovieLens and Kuching datasets have missing values removed entirely, which is a common approach.

Duplicate Removal: Duplicate entries can skew the results and inflate the importance of certain users or items. Removing these duplicates ensures a cleaner and more representative dataset. Duplicate entries are removed in both datasets.

Sparsity Reduction (MovieLens Only): The MovieLens dataset might contain users or movies with very few ratings (sparse entries). These entries can negatively affect the accuracy of recommendation algorithms. Here, users or movies with an exceptionally low

number of ratings are removed from the MovieLens dataset to improve the statistical significance of the results.

Timestamp Removal (MovieLens Only): The timestamp feature in the MovieLens dataset likely signifies when a rating was made. In this study, the focus is on the user-item interactions and preferences themselves, not the specific time of rating. Therefore, the timestamp feature is excluded from the analysis of the MovieLens dataset.

4.6.2 Hyperparameter Tuning

After data preprocessing and cleaning, some algorithms require hyperparameter tuning. The hypertuning phase aimed to identify the optimal configurations for diverse recommendation algorithms by leveraging a smaller version of the MovieLens dataset. Since, this is standard dataset, the parameters can be easily transferable to other domains. The process involved tailoring different parameters for each algorithm to maximize their performance. The following algorithms underwent hyperparameter tuning.

ResNetMF: A grid search approach was employed to systematically evaluate combinations of parameters like the number of latent factors (K), learning rate, and regularization penalty. This exploration identified the optimal configuration: $K=17$ latent factors, a learning rate of 0.01, and a regularization penalty of 0. This configuration effectively captured the underlying relationships within the data, allowing the model to make accurate recommendations.

CNN: A random search approach was employed to explore various CNN architectures. This exploration focused on parameters like the number of convolutional layers, filter sizes, and training epochs. Interestingly, the method yielded a promising CNN

architecture. A single convolutional layer utilizing a 3x3 filter size, trained for 15 epochs. This configuration effectively learned the spatial features embedded within the data.

KNN-based methods: Grid search was utilized to fine-tune parameters crucial for KNN's performance, including the number of neighbors (K) considered for predictions, the similarity metric used (e.g., Pearson correlation), and the minimum number of common items/users required to calculate similarity. This search identified the optimal configuration as K=40 neighbors, the Pearson similarity metric, and a minimum of 1 common item/user. This configuration ensured that the model considered the most relevant neighbors when making recommendations.

SVD++ and NMF: A grid search approach was employed to fine-tune the hyperparameters of SVD++ and NMF, two established matrix factorization methods. This process optimized factors like the number of latent factors (K), training iterations (epochs), learning rate, and regularization parameter. The search identified an optimal configuration for both methods, utilizing 25 latent factors, 15 training epochs, a learning rate of 0.01, and a regularization parameter of 0.02. This configuration effectively captured the underlying structure within the data, enabling the models to generate accurate recommendations.

Furthermore, all algorithms were evaluated using different data split ratios for training and testing sets (80/20, 70/30, 60/40). The 80/20 split, allocating a larger portion of data for training, consistently yielded the highest accuracy across all algorithms. This outcome underscores the significance of data distribution for training effectiveness. By having a larger training set, the models were able to learn more robust patterns from the data, leading to improved recommendation accuracy.

4.7 Conclusion

This chapter outlines a comprehensive and transparent experimental design to evaluate the effectiveness of the ResNetMF recommendation method. Employing a diverse set of established algorithms as baselines and evaluation metrics, the study aims to assess both accuracy and relevancy of recommendations through rigorous evaluation. The hyperparameter tuning phase plays a crucial role in optimizing the performance of ResNetMF and other algorithms. This design allows for meaningful comparisons with existing methods and paves the way for advancements in recommendation systems.

CHAPTER 5

RESULTS AND ANALYSIS

5.1 Introduction

This chapter provides a comprehensive evaluation of the proposed method. This includes the comparison of ResNetMF with other widely-used algorithms such as convolutional matrix factorization (CMF), convolutional neural networks (CNN), and autoencoders. The efficiency and effectiveness of ResNetMF will be tested on selected datasets, and the results will be analyzed in detail. It presents and analyzes the experimentation results, including the comparison with other algorithms and the query analysis process. Finally, the chapter concludes with observations on the results and provide concluding remarks.

5.2 Results and Discussion

This section is dedicated to providing a comprehensive analysis and evaluation of the experimental results obtained from implementing the ResNetMF method and other algorithms used for comparison. The objective is to examine and interpret the outcomes to gain insights into each algorithm's performance, effectiveness, and comparative advantages. Through a meticulous examination of the results, valuable conclusions can be drawn, enabling a deeper understanding of the strengths and limitations of the ResNetMF approach compared to the other algorithms.

5.2.1 Experiment Number 1

The first experiment was conducted on the MovieLens dataset, a widely recognized benchmark in recommendation systems research. This experiment investigates the

effectiveness of ResNetMF, a deep learning approach for recommendation systems. Its performance against other deep learning methods (CNN, Autoencoders, CMF, RNN, NCF) and various baseline algorithms (SVD, KNN) is compared.

The deep learning algorithms were first evaluated. The setup ensured a randomized data distribution through shuffling. Two essential accuracy metrics (MAE and RMSE) and three relevancy metrics (precision, recall, F1-score) were utilized to assess performance.

The table below summarizes the average values of MAE and RMSE for each algorithm across 15 epochs, providing a clear comparison of their accuracy and prediction quality.

Table 5.1: RMSE and MAE of Deep Learning Algorithms

Algorithm	RMSE	MAE	Note
Convolutional Neural Network (CNN)	0.8156	0.6657	Moderate Scores
Recurrent Neural Network (RNN)	0.8427	0.7106	Lowest Scores
Neural Collaborative Filtering (NCF)	0.7933	0.6303	Good Scores
Convolutional Matrix Factorization (CMF)	0.8235	0.6378	Moderate Scores
Autoencoder	0.6101	0.4749	Best Scores
ResNetMF	0.7348	0.5731	Good Scores

The best-performing algorithm in terms of both RMSE and MAE is the Autoencoder, while the worst-performing algorithm is the Recurrent Neural Network (RNN).

ResNetMF demonstrates higher performance in both MAE and RMSE compared to CMF, CNN, RNN, and NCF. It achieves a notable 14.67% improvement in RMSE compared to RNN, and surpasses others by 7.95% (NCF), 11% (CNN) and 12.05% (CMF). While NCF achieves closer accuracy to ResNetMF, its training time is significantly slower.

However, the autoencoder exhibits higher accuracy on MAE and RMSE metrics. This advantage comes at a cost, though, as the autoencoder's training time is significantly slower than ResNetMF's (average of 1982.31 seconds compared to 258.9 seconds). This highlights the trade-off between accuracy and efficiency. ResNetMF offers a well-balanced solution, achieving high accuracy with faster training times. The table below further emphasizes ResNetMF's advantage in terms of training speed.

Table 5.2: Training Speed of Deep Learning Methods

Algorithm (Average Training Time)	Duration (Seconds)	Note
Convolutional Neural Network (CNN)	372.8	Moderate Training Speed
Recurrent Neural Network (RNN)	1899.145	Slow Training Speed
Neural Collaborative Filtering (NCF)	1575.12	Slow Training Speed
Convolutional Matrix Factorization (CMF)	305.2	Fast Training Speed
Autoencoder	1982.31	Slowest Training Speed
ResNetMF	258.9	Fastest Training Speed

ResNetMF exhibits notably faster training times compared to all tested deep learning methods. It is 17.88% faster than CMF, 43.98% faster than CNN, and much faster than other algorithms. This advantage opens up possibilities for real-world applications where time constraints are crucial.

ResNetMF, achieves higher performance compared to other deep learning-based approaches in terms of precision, recall, and F1-score. This demonstrates the increased accuracy of ResNetMF. Specifically, ResNetMF achieves a precision of 7.16, recall of 0.4409, and F1-score of 0.8076. These values outperform CNN (precision: 6.49, recall: 0.4017, F1-score: 0.7565), RNN (precision: 6.02, recall: 0.398, F1-score: 0.747), NCF

(precision: 7.01, recall: 0.413, F1-score: 0.778), and CMF (precision: 6.6, recall: 0.41, F1-score: 0.772).

Table 5.3: Precision, Recall, F1-Score of Deep Learning Methods

Algorithm	Precision	Recall	F1-Score	Note
CNN	6.49	0.4017	0.7565	-
RNN	6.02	0.398	0.747	Lowest Scores
NCF	7.01	0.413	0.778	-
CMF	6.6	0.41	0.772	-
Autoencoder	7.9	0.495	0.9316	Best Scores
ResNetMF	7.1673	0.4409	0.8076	-

Let us interpret each evaluation metric within the context of the ResNetMF. The Precision value of 7.1673 signifies that, on average, 7.1673 out of the top 10 recommended items are deemed relevant or useful to the user. The recall value of 0.4409 suggests that ResNetMF successfully identified and recommended approximately 44.09% of the relevant items to the user out of all the available items in the system. The F1-score of 0.8076 indicates a harmonious balance between precision and recall, achieving a relatively high level of performance in both measures.

Among deep learning methods, ResNetMF achieved competitive accuracy (MAE, RMSE) while excelling in training speed. While Autoencoder offered higher accuracy, its significantly slower training times limit their practicality.

This experimental analysis proceeds to compare ResNetMF with various baseline algorithms commonly used in recommendation systems. These include matrix factorization techniques (SVD, SVD++, NMF), Collaborative filtering techniques (KNN-based methods, SlopeOne), and Content-based techniques (NormalPredictor, Co-clustering).

The table below encapsulates all the experimented algorithms' RMSE and MAE values. Notably, the proposed ResNetMF algorithm outshines its counterparts, displaying substantially better MAE and RMSE scores, thereby converging closer to the target rating. This compelling evidence unequivocally establishes the optimal performance of ResNetMF compared to the other benchmark algorithms. These findings emphasize the effectiveness of ResNetMF and accentuate its potential as a highly effective recommendation algorithm.

Table 5.4: RMSE and MAE of Baseline Algorithms

Algorithm Type	Algorithms	RMSE	MAE	Note
Matrix Factorization Techniques	SVD	0.879	0.677	-
	SVD++ (SVDpp)	0.869	0.667	-
	Non-Negative Matrix Factorization (NMF)	0.9788	0.7169	-
Collaborative Filtering Techniques	KNNbaseline	0.882	0.674	-
	KNNbasic	0.958	0.735	-
	KNN with means	0.906	0.692	-
	KNNwithzscore	0.904	0.686	-
	SlopeOne	0.911	0.696	-
	BaselineOnly	0.876	0.676	-
Content-Based Techniques	NormalPredictor	1.422	1.136	Lowest Scores
	Co-clustering	0.952	0.737	-
Proposed Method	ResNetMF	0.7348	0.5731	Best Scores

As above table shows, ResNetMF consistently achieves the lowest RMSE and MAE values, indicating higher accuracy in predicting user ratings. Compared to the best-performing baseline (SVDpp) attaining the best RMSE value of 0.869 and MAE value of

0.667 at its peak, ResNetMF offers a significant improvement of 18.10% and 19.27% in RMSE and MAE, respectively.

The table below presents a comparative assessment of ResNetMF against the peak results of other benchmark algorithms regarding RMSE and MAE to provide a quantitative analysis. The percentage values highlight the degree of improvement attained by the proposed method compared to the best-performing algorithms in each metric. This analysis clearly and concisely represents ResNetMF's performance and underscores its significance in achieving more precise and reliable predictions.

Table 5.5: ResNetMF Improvement in RMSE & MAE

Algorithm Type	Algorithms	ResNetMF Percentage Improvement Compared	
		RMSE	MAE
Matrix Factorization Techniques	SVD	19.57 %	18.08 %
	SVD++ (SVDpp)	18.21 %	16.36 %
	Non-Negative Matrix Factorization (NMF)	33.2 %	25.06 %
Collaborative Filtering Techniques	KNNbaseline	20.01 %	17.56 %
	KNNbasic	30.37 %	28.32 %
	KNN with means	23.33 %	20.78 %
	KNNwithzscore	23.03 %	19.68 %
	SlopeOne	23.93 %	21.42 %
	BaselineOnly	19.17 %	18.02 %
Content-Based Techniques	NormalPredictor	93.28 %	98.07 %
	Co-clustering	29.57 %	28.53 %
Proposed Method (it is baseline to compare)	ResNetMF	0%	0%

ResNetMF achieved higher value in precision, recall and F1-score metrics than all other benchmark algorithms. As it is displayed in the table below. Even though matrix factorization based methods achieved high results compare to other algorithms, but, their training speed was lower than ResNetMF. For example, SVDpp requires significantly longer training (3524.86 seconds per epoch) compared to ResNetMF (258.9 seconds per epoch).

Table 5.6: Precision, Recall and F1-Score@10 Results of tested algorithms

Algorithms	Precision@10	Recall@10	F1-Score@10	Note
SVD	6.2154	0.339	0.7282	-
SVD++ (SVDpp)	6.4302	0.371	0.7404	-
NMF	6.6583	0.367	0.6999	-
KNNbaseline	4.4572	0.392	0.7463	-
KNNbasic	5.1326	0.342	0.7348	-
KNN with means	5.2785	0.389	0.7296	-
KNNwithzscore	4.9613	0.4	0.7477	-
SlopeOne	4.1254	0.393	0.7361	-
BaselineOnly	3.8501	0.392	0.733	-
NormalPredictor	4.0115	0.316	0.552	Lowest Scores
Co-clustering	4.521	0.354	0.6944	-
ResNetMF	7.1673	0.4409	0.8076	Best Scores

The results from the initial experiment unequivocally demonstrate that ResNetMF outperforms all other benchmark algorithms and commonly used algorithms across all evaluation metrics. Furthermore, combining a deep residual network with matrix factorization led to enhanced performance in the matrix factorization algorithm.

To illustrate, before incorporating the deep residual network, the matrix factorization (SVD) algorithm achieved an F1-score of 0.7282. However, after implementing the proposed ResNetMF method, the F1-score notably improved to 0.8076. This study validates the

proposed method's effectiveness and sheds light on its ability to enhance the performance of matrix factorization algorithms.

These results demonstrate ResNetMF's effectiveness. It surpasses various baseline methods in terms of accuracy (RMSE, MAE, precision, recall, F1-score) while maintaining faster training times compared to other algorithms like SVDpp.

This compelling evidence establishes ResNetMF as a promising approach for recommendation systems. It offers a well-balanced solution, achieving high accuracy with faster training times, making it ideal for real-world applications.

5.2.2 Experiment Number 2

In experiment number 2, the Kuching dataset was utilized to investigate the performance of the recommendation algorithms. This experiment revealed higher RMSE values than the previous experiment on the MovieLens dataset. This discrepancy can be attributed to the Smart Kuching dataset contains a larger number of blank ratings

ResNetMF outperforms baseline algorithms (CMF, CNN, RNN, NCF, and SVDpp) in both Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) metrics. The average MAE of ResNetMF (0.7172) indicates its ability to make more accurate predictions with a lower average error compared to other algorithms (CMF: 1.039, CNN: 1.08, RNN: 1.17, NCF: 0.911, SVDpp: 1.5609). These results demonstrate ResNetMF's effectiveness in minimizing the difference between predicted and actual ratings, making it a strong choice for recommendation systems using the Kuching dataset.

ResNetMF's performance is further validated by its lower Root Mean Squared Error (RMSE) of 1.2991, demonstrating its proficiency in reducing the overall magnitude of prediction errors. In comparison, other algorithms yielded higher RMSE values: CMF

(1.4637), CNN (1.460), RNN (1.501), NCF (1.351), and SVDpp (1.951). These results further emphasize ResNetMF's effectiveness in accurately estimating user preferences and mitigating the discrepancies between predicted and observed ratings on the Kuching dataset.

Table 5.7: RMSE and MAE of Deep Learning Methods

Algorithm	RMSE	MAE	Note
CNN	1.460	1.08	-
RNN	1.501	1.17	Lowest Scores
NCF	1.351	0.911	-
CMF	1.4637	1.039	-
Autoencoder	0.981	0.61	Best Scores
ResNetMF	1.2991	0.7172	-

However, the autoencoder achieved better results with the MAE result of 0.61 and an RMSE of 0.981. However, the autoencoder is slower in training even though it almost took the same time to finish the training as ResNetMF. It is important to note that the training duration comparison was not feasible in this experiment. ResNetMF, CMF, and CNN had similar training times, primarily due to the considerably smaller size of the Kuching dataset compared to the MovieLens dataset. This disparity in dataset size could have influenced the training duration of the algorithms.

The higher achievement of ResNetMF over the baseline algorithms in terms of both MAE and RMSE values underscores its effectiveness in addressing the challenges posed by the sparser Kuching dataset. By leveraging the combined power of matrix factorization and residual networks, ResNetMF demonstrates enhanced recommendation capabilities, enabling more precise predictions.

Moreover, the ResNetMF, demonstrated higher accuracy over all the baseline algorithms, achieving results with a peak RMSE of 1.144 and the MAE of 0.663. These performance metrics indicate ResNetMF's ability to make highly accurate predictions, surpassing the performance of all other evaluated algorithms. Notably, among the considered baseline models, the SVDpp algorithm exhibited the highest level of accuracy, achieving a peak RMSE of 1.336 and the MAE of 1.082. The results suggest that the SVDpp algorithm excels in predicting future time series data compared to the other baseline models.

The table below provide a comprehensive overview of the comparative performance. This table representation facilitates a clear understanding of each algorithm's RMSE and MAE values. It becomes evident from the table that ResNetMF consistently outperforms the other algorithms. The gap between ResNetMF's performance and that of the baseline algorithms further reinforces the efficacy of the proposed method.

Table 5.8: RMSE and MAE of Baseline Algorithms

Algorithm Type	Algorithms	RMSE	MAE	Note
Matrix Factorization Techniques	SVD	1.369	1.145	-
	SVD++ (SVDpp)	1.336	1.082	-
	Non-Negative Matrix Factorization (NMF)	1.433	1.136	-
Collaborative Filtering Techniques	KNNbaseline	1.377	1.159	-
	KNNbasic	1.468	1.277	-
	KNN with means	1.371	1.127	-
	KNNwithzscore	1.344	1.093	-
	SlopeOne	1.413	1.179	-
	BaselineOnly	1.474	1.286	-
Content-Based Techniques	NormalPredictor	2.5	2.113	Worst Scores
	Co-clustering	1.391	1.12	-
Proposed Method	ResNetMF	1.2991	0.7172	Best Scores

The proposed ResNetMF method outperformed other benchmark algorithms in the second experiment. Notably, the Precision, Recall, and F1-Score of ResNetMF were higher than all the other algorithms as summarized in the table below.

Table 5.9: Precision, Recall and F1-Score@10 Results

Algorithm Type	Algorithms	Precision@10	Recall@10	F1-Score@10	Note
Matrix Factorization Techniques	SVD	5.731	0.641	0.683	-
	SVD++ (SVDpp)	5.741	0.639	0.685	-
	Non-Negative Matrix Factorization (NMF)	5.734	0.574	0.644	-
Collaborative Filtering Techniques	KNNbaseline	4.701	0.64	0.669	-
	KNNbasic	4.725	0.716	0.72	-
	KNN with means	4.71	0.624	0.664	-
	KNNwithzscore	4.705	0.635	0.668	-
	SlopeOne	4.681	0.666	0.673	-
	BaselineOnly	3.624	0.659	0.641	Worst Scores
Content-Based Techniques	NormalPredictor	3.998	0.525	0.592	-
	Co-clustering	4.696	0.645	0.669	-
Deep Learning Techniques	CNN	6.18	0.701	0.656	-
	RNN	0.581	0.699	0.634	-
	NCF	0.7021	0.722	0.712	-
	CMF	6.74	0.709	0.691	-
	Autoencoders	8.21	0.8001	0.81	Best Scores
Proposed Method	ResNetMF	7.93	0.742	0.766	-

The proposed ResNetMF method achieves the lowest RMSE (1.2991) compared to all other algorithms. The closest competitor, SVD++, has an RMSE of 1.336, which is higher than ResNetMF. The MAE for ResNetMF is 0.7172, which is significantly lower than all other algorithms. The next best, SVD++, has an MAE of 1.082. ResNetMF achieves a Precision@10 of 7.93, which is second only to Autoencoders (8.21). High precision indicates that a significant proportion of the recommended items are relevant, showing that ResNetMF provides high-quality recommendations. ResNetMF has a Recall@10 of 0.742, which is lower than Autoencoders (0.8001) but still competitive. The F1-Score@10 for ResNetMF is 0.766, which is lower than Autoencoders (0.81) but higher than all other algorithms.

The proposed ResNetMF algorithm yields positive outcomes due to its ability to achieve the lowest RMSE and MAE, indicating high recommendation accuracy. It also maintains competitive precision, recall, and F1-Score metrics, demonstrating its effectiveness in providing high-quality recommendations. These results justify the algorithm's good performance in the context of the given research objectives, achieving significant improvements over traditional and some advanced recommendation methods.

The proposed model outperforms all other baseline algorithms and alternative methods, such as CMF and CNN, regarding prediction and recommendation accuracy. These results offer strong evidence supporting the efficiency of the proposed model compared to existing algorithms and baseline methods.

5.2.3 Key Findings from Recommendation Experiments

These two controlled experiments investigate the effectiveness of ResNetMF, a deep learning approach for recommendation systems. Both experiments evaluate its performance on various metrics compared to existing recommendation algorithms and baseline methods.

The first experiment utilizes the MovieLens dataset, a widely recognized benchmark in recommendation system research. This experiment assesses ResNetMF against other deep learning methods (CNN, RNN, NCF, CMF, Autoencoders) and various baseline algorithms (SVD, KNN). The evaluation focuses on accuracy metrics (MAE, RMSE) and relevancy metrics (Precision, Recall, F1-score).

The results of Experiment 1 demonstrate that ResNetMF outperforms all deep learning methods tested. It achieves better accuracy in terms of both MAE and RMSE while maintaining competitive relevancy scores. Notably, while Autoencoders exhibit slightly higher accuracy, their significantly slower training times limit their practicality. This highlights a key advantage of ResNetMF which is achieving high accuracy with faster training speeds, making it ideal for real-world applications with time constraints.

Furthermore, Experiment 1 compares ResNetMF with various baseline algorithms commonly used in recommendation systems. These include matrix factorization techniques, collaborative filtering techniques, and content-based techniques. The results show that ResNetMF significantly outperforms all baseline methods in terms of both MAE and RMSE, indicating higher accuracy in predicting user ratings.

The second experiment investigates the performance of ResNetMF on the Kuching dataset, which had more missing ratings compared to the MovieLens dataset. This experiment aims to assess how ResNetMF handles data with a higher percentage of missing values.

The results of Experiment 2 confirm the effectiveness of ResNetMF observed in Experiment 1. It achieves better results than baseline algorithms in terms of both MAE and RMSE. However, due to the increased sparsity of the Kuching dataset, the overall RMSE

and MAE values are higher compared to Experiment 1. This highlights that the overall accuracy might be impacted by the extent of missing ratings.

The two-experiment approach employed in this study provides a comprehensive evaluation of ResNetMF's performance. The findings demonstrate its effectiveness across different datasets. Notably, ResNetMF achieves high accuracy in predicting user ratings with faster training times compared to several existing methods. These qualities make ResNetMF a promising recommendation algorithm for real-world applications.

5.2.4 Dissemination Experiment (Query Analysis)

Tourists visiting the historic city of Kuching faced challenges in planning their itineraries due to information overload. There was an abundance of historical sites, museums, events, and dining options, making it overwhelming for visitors. Imagine you're on vacation in a new city, overwhelmed by countless restaurants and attractions. A good recommendation system can be your best friend, filtering through options and presenting the most relevant choices. Beyond measuring effectiveness, a crucial aspect often overlooked is user satisfaction with the recommendations. This study tackles this challenge by focusing on measuring this dimension of user experience.

In order to tackle this issue, this study proposed a combined recommendation and query analysis system which consists of a multi-criteria recommendation engine alongside a query analysis mechanism in which users write their query and system must interactively answer. This system is able to answer user dynamic user preferences. Once the recommendation list is generated, the system determines the recommendation orders based on user query. This mechanism allows users to have more engagement with this system.

However, the study recognizes the need to establish a method for measuring the dissemination performance of the recommendation system. Unlike widely used benchmark measurements like root mean square error (RMSE) that assess prediction accuracy, evaluating the dissemination performance presents a distinct challenge. While the recommendation system is crucial in narrowing down the options and filtering the results, the final decision-making lies with human users and their cognitive processes. Assessing the impact of this decision-making on the overall recommendation experience poses difficulties in terms of measurement.

The study highlights the challenge of measuring dissemination performance due to the involvement of human decision-making in the recommendation process. Considering the complexity of measuring the effectiveness of query analysis (dissemination), the study focuses on assessing user satisfaction with the recommended results. A method that involves selecting 50 random users from the Kuching dataset and requesting them to provide at least 5 queries to the system is proposed. Subsequently, the users are asked to rate their satisfaction and perceived relevance of the results on a scale of 1 to 10. A rating of 1 indicates low satisfaction and relevance, while a rating of 10 signifies high satisfaction and result relevance.

The study collected a total of 286 queries from users, and the average rating derived from these responses was 7.25. The result indicates that, on average, 72.5% of the users expressed satisfaction with the disseminated information. These findings suggest that the recommendation system and the information selection process performed well, as most users found the recommended results satisfactory.

However, it is crucial to acknowledge the limitation of the study's sample size of 50 users and only Kuching City attraction data utilized, which may impact the generalizability of the results. Future research should include a larger and more diverse user sample to obtain more representative insights.

In conclusion, this study proposes a user-centered recommendation system that tackles information overload for tourists in Kuching. By measuring user satisfaction with the recommendations, the study demonstrates the system's effectiveness in delivering relevant information. While future research can explore larger user groups and data sources, this initial evaluation paves the way for developing more engaging and informative recommendation systems for travellers.

5.3 Contribution

This study introduces ResNetMF, a novel deep learning approach for recommendation systems. The research investigates its effectiveness through controlled experiments on two datasets: MovieLens and Kuching dataset. The results demonstrate that ResNetMF outperforms existing deep learning methods (CNN, RNN, NCF, CMF, Autoencoder) and various baseline algorithms like SVD and KNN in terms of accuracy. This is measured by metrics like Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE), indicating ResNetMF's ability to predict user ratings more accurately.

The experimentation results show that the ResNetMF can find more latent features that match better to users' preferences than baseline algorithms. It performs better than other baseline methods in finding latent dimensionality and latent features in all datasets which leads to higher recommendation accuracy.

One key advantage of ResNetMF is its training speed. While some methods achieve slightly higher accuracy (Autoencoder), they come at the cost of significantly slower training times. ResNetMF achieved high accuracy without sacrificing training speed. This makes it ideal for real-world deployments where timely recommendations are crucial.

The experiments also explore how ResNetMF handles data with a higher percentage of missing values. While overall accuracy decreases when using the Kuching dataset, ResNetMF still performs better than baseline algorithms in terms of MAE and RMSE. This highlights its effectiveness in dealing with data imperfections and providing accurate recommendations.

The study goes beyond just accuracy by emphasizing a user-centered approach. The recommendation system incorporates a query analysis mechanism that allows users to express their preferences and interact with the system. This helps the system personalize recommendations and cater to dynamic user needs.

To measure the system's effectiveness in delivering relevant information to users, the study proposes a user satisfaction survey. The results show that on average, 72.5% of users in a sample group were satisfied with the recommendations provided by the system on the Kuching dataset. This indicates that the combined recommendation and query analysis approach successfully tackles information overload and delivers user-satisfying results.

In conclusion, this research introduces ResNetMF, a promising recommendation system that offers improved accuracy, and faster training speeds. Additionally, it emphasizes a user-centered approach through query analysis and user satisfaction evaluation, paving the way for more engaging and informative recommendation systems for travelers.

5.4 Conclusion

This chapter demonstrates the effectiveness of the proposed ResNetMF method through experimentation, results, and analysis. The study introduces a novel multi-criteria approach that utilizes latent feature extraction in recommender systems by combining matrix factorization with deep residual networks. This method significantly improved recommendation accuracy by considering multiple factors influencing user decision-making. Consequently, users received more relevant recommendations, leading to a more positive user experience and increased trust in the system. ResNetMF's ability to find different criteria through latent dimensionality analysis further contributed to this improved user experience. The chapter then delves into the user experience by exploring information selection during dissemination. Since dissemination effectiveness relies on user cognition and choice, a universally accepted measurement method hasn't been established. To evaluate user interaction with the system's recommendations, the study leveraged direct user feedback. This approach provided valuable insights, highlighting the system's effectiveness in helping users select information. While a larger sample size could provide even more comprehensive data, the current results offer encouraging evidence of the system's positive impact. Future research can explore expanding the user feedback pool to further refine our understanding of user behavior within the system.

CHAPTER 6

CONCLUSION AND FUTURE WORKS

6.1 Introduction

The tourism industry has experienced significant growth in recent years, leading to a vast amount of data and information related to tourism. This industry significantly impacts economic growth, as studies have shown. While this data offers valuable insights for travellers, it also creates a challenge of information overload. Recommendation systems, a sub-discipline of information filtering, have emerged as a solution to this challenge by delivering personalized and relevant information to users. Existing recommendation systems in tourism can struggle to effectively address this challenge due to limitations in incorporating user preferences dynamically, and considering different criteria of data. Despite various studies attempting to address the challenges of recommendation systems in the tourism industry, there is still considerable scope for improvement.

This study aimed to improve the accuracy of tourism recommendation systems by proposing a multi-criteria method. The proposed method consists of two main components of the recommendation engine (ResNetMF) which is a hybrid multi-criteria recommendation system that combines a common recommendation technique, matrix factorization, with deep learning methods and a query analysis part for interactive user engagement to answer user dynamic preferences. The experimentation results demonstrate that the proposed approach effectively has more accuracy on recommendations and has high user satisfaction, which is a significant accomplishment for this study.

The recommendation system reduces the scope, and the decision on dissemination takes place with the user query analysis technique. However, there were challenges, limitations, and future work needed to improve current trends.

This chapter is structured as follows. Section 6.1 is an introductory summary of the research. Section 6.2 is the achievements part; Section 6.3 is about the encountered limitations. Future work is in Section 6.4. Moreover, Section 6.5 is about the summary and conclusion.

6.2 Achievements

This study successfully developed a multi-criteria recommendation system, ResNetMF, for the tourism industry. ResNetMF leverages a deep neural network in conjunction with matrix factorization to significantly improve the accuracy of recommendation tasks. The proposed method outperforms all tested algorithms, achieving better scores on accuracy metrics with lower training speed.

This study successfully achieved its first objective. It is demonstrating the effectiveness of the proposed multi-criteria recommendation algorithm, ResNetMF. The proposed multi-criteria recommendation algorithm has higher accuracy in terms of RMSE in the MovieLens dataset. ResNetMF outperforms the other tested deep learning methods by margins ranging from 7.95% (NCF) to 12.05% (CMF) in terms of RMSE and average improvement of 11.41% improvements compare to tested deep learning methods. Furthermore, ResNetMF surpasses traditional recommendation techniques by even greater margins, achieving improvements ranging from 19.57% compared to SVD to a 93.28% improvement over the NormalPredictor method and average of 30.3% improvement in RMSE compared to the traditional methods exceeding the target of at least 8% improvement.

The second objective of evaluating training speed was also successfully achieved. The proposed multi-criteria recommendation algorithm, ResNetMF, demonstrates significantly faster training speeds compared to several established baseline methods. Notably, ResNetMF exhibits a 17.88% improvement over CMF and a 43.98% improvement over CNN. Furthermore, ResNetMF boasts a training time of 258.9 seconds, which is considerably faster than both RNN with 1899.145 seconds and NCF with 1575.12 seconds. These results translate to speed improvements, exceeding the initial target of achieving at least a 10% reduction in training time compared to the baselines.

Furthermore, this study designed and integrated a query analysis mechanism into the multi-criteria recommendation system. This mechanism allows for user interaction and dynamic preference expression. Users can submit queries that are analyzed to identify items matching their criteria and preferences. The system analyzes user queries to identify information that aligns with user criteria and preferences. This method effectively filters out irrelevant information, resulting in a more focused set of recommendations that directly address user dynamic preferences. The core principle lies in analyzing the user's query and searching the database for matching items. The system then counts the matches and disseminates the related items to the user. An experiment to gauge the effectiveness of the information dissemination method yielded positive results, with over 72.5% of users expressing satisfaction with the information they received. Therefore, the third research objective is achieved which was intended to have at least 50% satisfaction.

Overall, the proposed method of disseminating information is a step in the right direction toward developing more effective recommendation systems in the tourism

industry. However, more research is needed to improve the method further and address its limitations.

6.3 Limitations

This study encountered several limitations that highlight opportunities for future research. The initial plan involved collecting real-world Sarawak tourism data to ensure adherence to proper dissemination practices. However, due to the COVID-19 outbreak, acquiring the desired dataset proved infeasible. While the survey data (Kuching Dataset) formed part of the original plan, the creation of a knowledge graph using the collected data was postponed for future work.

Despite achieving significant improvements in evaluation metrics, the proposed method necessitates real-world validation. Testing the system in operational tourism settings is crucial for confirming its effectiveness and performance.

The proposed multi-criteria recommendation system demonstrated promising results on the tested datasets. However, for a comprehensive evaluation, it requires more data and user interaction. The effectiveness and efficiency of any recommendation system are heavily influenced by the quality, quantity of data available, and the level of user engagement. To fully assess the proposed system's performance, future studies should utilize larger datasets that reflect real-world tourism information and incorporate a greater number of users interacting with the system.

This extended evaluation would help identify any potential limitations or shortcomings of the system, allowing for further refinement and improvement in its accuracy and effectiveness for user recommendations.

The dissemination aspect of the study also presented limitations. While the proposed method for user-criteria and query-based information dissemination achieved a 72.5% user satisfaction rate, challenges arose in measuring user cognition, a notoriously difficult aspect. A new measurement approach was designed specifically for this study, yielding positive results. However, the sample size for the user satisfaction survey was limited to 50 participants. This limited sample size may affect the validity and generalizability of the findings to a larger population. Therefore, future studies should expand the sample size to provide a more robust evaluation of the proposed dissemination method.

Additionally, further research is needed to investigate the effectiveness and feasibility of the proposed method in real-world tourism settings. This exploration can be pursued as part of future work.

6.4 Research Contribution

This study introduces a multi-criteria recommendation system, ResNetMF, which is a deep learning approach for recommendation systems that leverages the strengths of both residual networks and matrix factorization. The key contributions of ResNetMF are as follow.

Recommendation accuracy improved. ResNetMF achieves better accuracy in predicting user ratings compared to existing deep learning methods (CNN, CMF) and various baseline algorithms (SVD, KNN) across multiple evaluation metrics (MAE, RMSE, Precision, Recall, F1-Score). This is particularly evident in Experiment 1, where ResNetMF outperforms all other tested algorithms on the MovieLens dataset.

Faster Training Speed achieved. While achieving high accuracy, ResNetMF maintains significantly faster training times compared to other methods, especially Autoencoders. This efficiency makes it ideal for real-world applications with time constraints. Experiment 1 clearly demonstrates this advantage, showcasing ResNetMF's faster training compared to CNN and CMF.

The proposed system incorporates a query analysis mechanism, allowing users to interact with the recommendation engine through queries. This interactive element can help to answer to user dynamic preferences. This study acknowledges that the user satisfaction cannot be measure by traditional recommendation system metrics like RMSE and proposes a user-centered approach to evaluate dissemination performance. By measuring user satisfaction with the recommended results, the study provides valuable insights into user experience. The positive average rating (7.25) on a scale of 1 to 10 indicates user satisfaction with the information provided by the system.

To sum up, ResNetMF presents a promising approach for recommendation systems, offering a well-balanced solution that combines high accuracy with faster training times. This study sheds light on its potential for real-world applications, particularly in tourism where efficient and user-centric recommendation systems are crucial.

6.5 Future Works

The proposed work outperformed other tested algorithms. The future improvement is going in two directions. The first direction is the dissemination (query analysis), and the second part is the recommendation engine part. The proposed method is analyzing the user query and then selecting the best based on the user-recommended list based on the query.

It will adapt the GPT-3 (Generative Pre-Trained Transformer 3) technologies. GPT-3 is an advanced natural language processing (NLP) model developed by OpenAI. It belongs to a family of models known as transformers, which are particularly adept at understanding and generating human-like text. If you have heard of the trendy new machine learning models such as BERT, GPT-3, or T5, all of these models are based on transformers which is a neural network architecture. GPT-3 was trained on almost all public web data, estimated at 45 terabytes of text data. Thus, it is decided to adapt the GPT-3 transformers into the proposed method for better accuracy and more robust performance for future development. These neural network architecture transformers will be used in analyzing and mining the description of the textual data. Therefore, further research is needed to explore the potential benefits and limitations of using neural network architecture transformers in the proposed recommendation system.

The proposed method (ResNetMF) achieved good performance. However, there is much room for improvement in the recommendation part. First, the proposed method is needed to address the cold start issue, which can arise for new users and items. In order to solve the cold start issue, it is decided to develop a knowledge graph alongside the proposed method. Two knowledge graphs of users and items can be created. The first part is to create a knowledge graph of different tourist attractions. Thus, when a new item is added to the proposed system, it can be easily categorized. The second part is to create a knowledge graph of users based on the demographic information of users. Therefore, it is easier to connect users and recommend better to users.

Also, it is intended to use global and local factorization methods for the matrix factorization part to improve multi-criteria systems. A group of researchers from Wuhan

University proposes this technique. However, this technique is used in e-commerce platforms. It is planned to adapt their method to the proposed method.

6.6 Summary

Tourism has become one of the most popular industries in recent years. Many tourists look for information before planning their travel. However, we are facing an era of information overload. Recently, the tourism industry has become one of the popular subjects of recommendation research. The tourism industry demands a well-optimized recommendation system.

This study aimed to design and develop an intelligence information dissemination technique for the tourism industry using a multi-criteria hybrid recommendation system. The proposed method has more accuracy and improved recommendation tasks by recommending more relevant items to users.

The results show that the proposed method outperforms other tested algorithms regarding accuracy metrics such as MAE, RMSE, Precision, Recall, and F1-Score. The proposed method works better on datasets of varying sizes, has better training and testing speed, and can select the most relevant item to the user query. The study suggests that future work could involve using GPT-3 transformers to improve the performance of the recommendation system.

In conclusion, this study provides valuable insights into improving the efficiency and effectiveness of recommendation systems for the tourism industry. In the future, the study recommends exploring the use of GPT-3 transformers to improve the recommendation system's ability to select the most relevant item to the user query.

REFERENCES

- Adilaksa, Y., & Musdholifah, A. (2021). Recommendation System for Elective Courses using Content-based Filtering and Weighted Cosine Similarity. *2021 4th International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, 51–55. <https://doi.org/10.1109/ISRITI54043.2021.9702788>
- Alamdari, P. M., Navimipour, N. J., Hosseinzadeh, M., Safaei, A. A., & 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>
- Arif, Y. M., Nurhayati, H., Nugroho, S. M. S., & Hariadi, M. (2022). Destinations Ratings Based Multi-Criteria Recommender System for Indonesian Halal Tourism Game. *International Journal of Intelligent Engineering and Systems*, 15(1), 282–294. <https://doi.org/10.22266/ijies2022.0228.26>
- Behera, G., & Nain, N. (2022). Handling data sparsity via item metadata embedding into deep collaborative recommender system. *Journal of King Saud University - Computer and Information Sciences*, 34(10), 9953–9963. <https://doi.org/10.1016/j.jksuci.2021.12.021>
- Burke, R. (2002). [No title found]. *User Modeling and User-Adapted Interaction*, 12(4), 331–370. <https://doi.org/10.1023/A:1021240730564>
- Caserio, C., & Trucco, S. (2016). Relationship between Information System and Information Overload. A Preliminary Analysis. *INTERNATIONAL JOURNAL OF MANAGEMENT & INFORMATION TECHNOLOGY*, 11(5), 3040–3050. <https://doi.org/10.24297/ijmit.v11i5.5669>

- Castellano, G., Fanelli, A. M., & Torsello, M. A. (2011). NEWER: A system for NEuro-fuzzy WEb Recommendation. *Applied Soft Computing*, 11(1), 793–806. <https://doi.org/10.1016/j.asoc.2009.12.040>
- Ceri, S., Bozzon, A., Brambilla, M., Della Valle, E., Fraternali, P., & Quarteroni, S. (2013). An Introduction to Information Retrieval. In S. Ceri, A. Bozzon, M. Brambilla, E. Della Valle, P. Fraternali, & S. Quarteroni, *Web Information Retrieval* (pp. 3–11). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-39314-3_1
- Chala, O., Novikova, L., Chernyshova, L., & Kalnitskaya, A. (2020). Method for Detecting Shilling Attacks Based on Implicit Feedback in Recommender Systems. *EUREKA: Physics and Engineering*, 5, 21–30. <https://doi.org/10.21303/2461-4262.2020.001394>
- Chen, H., Fu, J., Zhang, L., Wang, S., Lin, K., Shi, L., & Wang, L. (2019). Deformable Convolutional Matrix Factorization for Document Context-Aware Recommendation in Social Networks. *IEEE Access*, 7, 66347–66357. <https://doi.org/10.1109/ACCESS.2019.2917257>
- Dudekula, K. V., Syed, H., Basha, M. I. M., Swamykan, S. I., Kasaraneni, P. P., Kumar, Y. V. P., Flah, A., & Azar, A. T. (2023). Convolutional Neural Network-Based Personalized Program Recommendation System for Smart Television Users. *Sustainability*, 15(3), 2206. <https://doi.org/10.3390/su15032206>
- Falk, K. (2019). Practical recommender systems. Manning Publications.
- Fayyaz, Z., Ebrahimian, M., Nawara, D., Ibrahim, A., & Kashef, R. (2020). Recommendation Systems: Algorithms, Challenges, Metrics, and Business Opportunities. *Applied Sciences*, 10(21), 7748. <https://doi.org/10.3390/app10217748>

- Forouzandeh, S., Rostami, M., & Berahmand, K. (2022). A Hybrid Method for Recommendation Systems based on Tourism with an Evolutionary Algorithm and Topsis Model. *Fuzzy Information and Engineering*, 14(1), 26–50. <https://doi.org/10.1080/16168658.2021.2019430>
- Gai, S., Zhao, F., Kang, Y., Chen, Z., Wang, D., & Tang, A. (2019). Deep Transfer Collaborative Filtering for Recommender Systems. In A. C. Nayak & A. Sharma (Eds.), *PRICAI 2019: Trends in Artificial Intelligence* (Vol. 11672, pp. 515–528). Springer International Publishing. https://doi.org/10.1007/978-3-030-29894-4_42
- García-Crespo, Á., López-Cuadrado, J. L., Colomo-Palacios, R., González-Carrasco, I., & Ruiz-Mezcua, B. (2011). Sem-Fit: A semantic based expert system to provide recommendations in the tourism domain. *Expert Systems with Applications*, 38(10), 13310–13319. <https://doi.org/10.1016/j.eswa.2011.04.152>
- Gavalas, D., Konstantopoulos, C., Mastakas, K., & Pantziou, G. (2014). Mobile recommender systems in tourism. *Journal of Network and Computer Applications*, 39, 319–333. <https://doi.org/10.1016/j.jnca.2013.04.006>
- Gupta, S., & Kant, V. (2022). A model-based approach to user preference discovery in multi-criteria recommender system using genetic programming. *Concurrency and Computation: Practice and Experience*, 34(11), e6899. <https://doi.org/10.1002/cpe.6899>
- Hasan, E., Ding, C., & Cuzzocrea, A. (2022). Multi-criteria Rating and Review based Recommendation Model. *2022 IEEE International Conference on Big Data (Big Data)*, 5494–5503. <https://doi.org/10.1109/BigData55660.2022.10020741>

- Hodson, T. O. (2022). Root-mean-square error (RMSE) or mean absolute error (MAE): When to use them or not. *Geoscientific Model Development*, 15(14), 5481–5487. <https://doi.org/10.5194/gmd-15-5481-2022>
- Hong, M., & Jung, J. J. (2021). Multi-criteria tensor model for tourism recommender systems. *Expert Systems with Applications*, 170, 114537. <https://doi.org/10.1016/j.eswa.2020.114537>
- Huang, C., Yan, D., & Liang, S. (2023). The Relationship between Information Dissemination Channels, Health Belief, and COVID-19 Vaccination Intention: Evidence from China. *Journal of Environmental and Public Health*, 2023, 1–11. <https://doi.org/10.1155/2023/6915125>
- Hwang, S., & Chae, D.-K. (2022). An Uncertainty-Aware Imputation Framework for Alleviating the Sparsity Problem in Collaborative Filtering. *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 802–811. <https://doi.org/10.1145/3511808.3557236>
- Ifada, N., Sophan, M. K., & Putri, N. F. D. (2023). A MinMax Item-based Method for Multi-Criteria Recommendation Systems. *Procedia Computer Science*, 227, 1020–1029. <https://doi.org/10.1016/j.procs.2023.10.611>
- Kaya, T. T., & Kaleli, C. (2023). Robustness Analysis of Multi-Criteria Top-n Collaborative Recommender System. *Arabian Journal for Science and Engineering*, 48(8), 10189–10212. <https://doi.org/10.1007/s13369-022-07568-w>
- Khozooii, N. S., Haratizadeh, S., & Keyvanpour, M. R. (2013). An Analytical Framework for Web Information Filtering Techniques. *International Journal of Hybrid Information Technology*, 6(6), 345–358. <https://doi.org/10.14257/ijhit.2013.6.6.31>

- Klaib, A. F., & Nuser, M. S. (2019). Evaluating EHR and Health Care in Jordan According to the International Health Metrics Network (HMN) Framework and Standards: A Case Study of Hakeem. *IEEE Access*, 7, 51457–51465. <https://doi.org/10.1109/ACCESS.2019.2911684>
- Ko, H., Lee, S., Park, Y., & Choi, A. (2022). A Survey of Recommendation Systems: Recommendation Models, Techniques, and Application Fields. *Electronics*, 11(1), 141. <https://doi.org/10.3390/electronics11010141>
- Kuanr, M., & Mohapatra, P. (2021). Assessment Methods for Evaluation of Recommender Systems: A Survey. *Foundations of Computing and Decision Sciences*, 46(4), 393–421. <https://doi.org/10.2478/fcds-2021-0023>
- Kumar, N., Pallavi, K. V., & Hanji, B. R. (2023). Personalized Travel Recommendation System Using Average Cumulative Rating Matrix Factorization Technique: Concept and Framework. *Vietnam Journal of Computer Science*, 10(02), 159–195. <https://doi.org/10.1142/S2196888822500361>
- Li, G., Hua, J., Yuan, T., Wu, J., Jiang, Z., Zhang, H., & Li, T. (2019). Novel Recommendation System for Tourist Spots Based on Hierarchical Sampling Statistics and SVD++. *Mathematical Problems in Engineering*, 2019, 1–15. <https://doi.org/10.1155/2019/2072375>
- Li, Z. (2018). Towards the next generation of multi-criteria recommender systems. *Proceedings of the 12th ACM Conference on Recommender Systems*, 553–557. <https://doi.org/10.1145/3240323.3240326>
- Liu, E., Chu, Y., Luan, L., Li, G., & Wang, Z. (2019). Mixing-RNN: A Recommendation Algorithm Based on Recurrent Neural Network. In C. Douligeris, D. Karagiannis, & D. Apostolou (Eds.), *Knowledge Science, Engineering and Management* (Vol.

- 11775, pp. 109–117). Springer International Publishing. https://doi.org/10.1007/978-3-030-29551-6_10
- Macias, E., Serrano, J., Vicario, J. L., & Morell, A. (2021). Novel Imputation Method Using Average Code from Autoencoders in Clinical Data. *2020 28th European Signal Processing Conference (EUSIPCO)*, 1576–1579. <https://doi.org/10.23919/Eusipco47968.2020.9287343>
- Malhan, M., & Dewani, P. P. (2023). Customer engagement on social networking sites: An experimental analysis in the tourism and hospitality sector. *Current Issues in Tourism*, 26(12), 1915–1940. <https://doi.org/10.1080/13683500.2022.2071685>
- 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. *Journal of Hospitality and Tourism Technology*, 7(3), 281–299. <https://doi.org/10.1108/JHTT-10-2014-0064>
- Musa, J. M., & Zhihong, X. (2020). Item Based Collaborative Filtering Approach in Movie Recommendation System Using Different Similarity Measures. *Proceedings of the 2020 6th International Conference on Computer and Technology Applications*, 31–34. <https://doi.org/10.1145/3397125.3397148>
- Nair, A., Baghel, A., More, A., & Shetty, T. (2023). Enhancing Book Recommendations: A Comprehensive Exploration of Cosine Similarity in Collaborative Filtering for Personalized Reading Experiences. *International Journal For Multidisciplinary Research*, 5(6), 10825. <https://doi.org/10.36948/ijfmr.2023.v05i06.10825>
- Nair, S. C. (2022). A Comprehensive Survey On Recommendation Systems Based On Collaborative Filtering. *2022 International Conference on Computing*,

- Communication, Security and Intelligent Systems (IC3SIS)*, 1–5.
<https://doi.org/10.1109/IC3SIS54991.2022.9885627>
- Nam, L. N. H. (2022). Towards comprehensive approaches for the rating prediction phase in memory-based collaborative filtering recommender systems. *Information Sciences*, 589, 878–910. <https://doi.org/10.1016/j.ins.2021.12.123>
- Natarajan, S., Vairavasundaram, S., Natarajan, S., & Gandomi, A. H. (2020). Resolving data sparsity and cold start problem in collaborative filtering recommender system using Linked Open Data. *Expert Systems with Applications*, 149, 113248. <https://doi.org/10.1016/j.eswa.2020.113248>
- Niu, B., Ma, J., & Yang, Z. (2021). A Comparative Study of CF And NCF In Children's Book Recommender System. 2021 3rd International Workshop on Artificial Intelligence and Education (WAIE), 43–47. <https://doi.org/10.1109/WAIE54146.2021.00017>
- Parasuraman, D., & Elumalai, S. (2021). Improving the accuracy of item recommendations by combining collaborative and content-based recommendations: A hybrid approach. *International Journal of Advanced Intelligence Paradigms*, 19(3/4), 262. <https://doi.org/10.1504/IJAIP.2021.116361>
- Pazzani, M. J. (1999). A Framework for Collaborative, Content-Based and Demographic Filtering. *Artificial Intelligence Review*, 13(5/6), 393–408. <https://doi.org/10.1023/A:1006544522159>
- Przystupa, K., Beshley, M., Hordiichuk-Bublivska, O., Kyryk, M., Beshley, H., Pyrih, J., & Selech, J. (2021). Distributed Singular Value Decomposition Method for Fast Data Processing in Recommendation Systems. *Energies*, 14(8), 2284. <https://doi.org/10.3390/en14082284>

- Pujahari, A., & Sisodia, D. S. (2020). Model-Based Collaborative Filtering for Recommender Systems: An Empirical Survey. *2020 First International Conference on Power, Control and Computing Technologies (ICPC2T)*, 443–447. <https://doi.org/10.1109/ICPC2T48082.2020.9071454>
- Shambour, Q. (2021). A deep learning based algorithm for multi-criteria recommender systems. *Knowledge-Based Systems*, 211, 106545. <https://doi.org/10.1016/j.knosys.2020.106545>
- Shambour, Q. Y., Abualhaj, M. M., & Abu-Shareha, A. A. (2023). Restaurant Recommendations Based on Multi-Criteria Recommendation Algorithm. *JUCS - Journal of Universal Computer Science*, 29(2), 179–200. <https://doi.org/10.3897/jucs.78240>
- Shambour, Q. Y., Abu-Shareha, A. A., & Abualhaj, M. M. (2022). A Hotel Recommender System Based on Multi-Criteria Collaborative Filtering. *Information Technology and Control*, 51(2), 390–402. <https://doi.org/10.5755/j01.itc.51.2.30701>
- Singh, K. U. (2021). A Multi-Criteria Movie Recommendation System based on User Preferences and Movie Features. *Mathematical Statistician and Engineering Applications*, 70(1), 348–360. <https://doi.org/10.17762/msea.v70i1.2317>
- Sinha, B. B., & Dhanalakshmi, R. (2022). Evolution of recommender paradigm optimization over time. *Journal of King Saud University - Computer and Information Sciences*, 34(4), 1047–1059. <https://doi.org/10.1016/j.jksuci.2019.06.008>
- Soleymani, H., Saeidnia, H. R., Ausloos, M., & Hassanzadeh, M. (2023). Selective dissemination of information (SDI) in the age of artificial intelligence (AI). *Library Hi Tech News*. <https://doi.org/10.1108/LHTN-08-2023-0156>

- Sunilkumar, C. N. (2020). A review of movie recommendation system: Limitations, Survey and Challenges. *ELCVIA Electronic Letters on Computer Vision and Image Analysis*, 19(3), 18–37. <https://doi.org/10.5565/rev/elcvia.1232>
- Thannimalai, V., & Zhang, L. (2021). A Content Based and Collaborative Filtering Recommender System. *2021 International Conference on Machine Learning and Cybernetics (ICMLC)*, 1–7. <https://doi.org/10.1109/ICMLC54886.2021.9737238>
- Wan, Y. (2022). Tourism intelligent recommendation system based on big data mining. *Proceedings of the 2022 6th International Conference on Electronic Information Technology and Computer Engineering*, 1040–1044. <https://doi.org/10.1145/3573428.3573615>
- Wang, M. (2020). Applying Internet information technology combined with deep learning to tourism collaborative recommendation system. *PLOS ONE*, 15(12), e0240656. <https://doi.org/10.1371/journal.pone.0240656>
- Wang, R. Y., & Strong, D. M. (1996). Beyond Accuracy: What Data Quality Means to Data Consumers. *Journal of Management Information Systems*, 12(4), 5–33. <https://doi.org/10.1080/07421222.1996.11518099>
- Wu, X. (2022). Comparison Between Collaborative Filtering and Content-Based Filtering. *Highlights in Science, Engineering and Technology*, 16, 480–489. <https://doi.org/10.54097/hset.v16i.2627>
- Wu, Y., Liu, X., Xie, M., Ester, M., & Yang, Q. (2016). CCCF: Improving Collaborative Filtering via Scalable User-Item Co-Clustering. *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, 73–82. <https://doi.org/10.1145/2835776.2835836>

- Yien, T. Y., Lim, N., Hon, K., Andam, J. L., & Tang, M. B. (2021). The Determinants of Quality in Accounting Information System: A Case of SMEs in Sibu Sarawak. *Borneo Journal of Social Sciences and Humanities*, 3, 112–124. <https://doi.org/10.35370/bjssh.2021.3.1-09>
- Yu, L., Huang, Z., Qiu, P., Gao, J., & Lu, F. (2020). An interpretable attraction recommendation method based on knowledge graph. *SCIENTIA SINICA Informationis*, 50(7), 1055–1068. <https://doi.org/10.1360/SSI-2019-0268>
- Zeeshan, Z., Ul Ain, Q., Bhatti, U. A., Memon, W. H., Ali, S., Nawaz, S. A., Nizamani, M. M., Mehmood, A., Bhatti, M. A., & Shoukat, M. U. (2021). Feature-based multi-criteria recommendation system using a weighted approach with ranking correlation. *Intelligent Data Analysis*, 25(4), 1013–1029. <https://doi.org/10.3233/IDA-205388>
- Zhang, H., Zhao, H., Liu, Q., Xu, T., Chen, E., & Huang, X. (2018). Finding potential lenders in P2P lending: A Hybrid Random Walk Approach. *Information Sciences*, 432, 376–391. <https://doi.org/10.1016/j.ins.2017.12.017>
- Zhang, S., Yao, L., Sun, A., & Tay, Y. (2020). Deep Learning Based Recommender System: A Survey and New Perspectives. *ACM Computing Surveys*, 52(1), 1–38. <https://doi.org/10.1145/3285029>
- Zhang, Y., Lu, X., Shi, Y., & Zhang, D. (2023). Hybrid Algorithm for Item Collaborative Filtering Based on Matrix Factorization. *2023 4th Information Communication Technologies Conference (ICTC)*, 276–284. <https://doi.org/10.1109/ICTC57116.2023.10154673>
- Zhang, Z., Fang, A., Cui, L., Pan, Z., Zhang, W., Tan, C., & Wang, C. (2021). Towards Exploring the Influence of Community Structures on Information Dissemination in

Sina Weibo Networks. *Discrete Dynamics in Nature and Society*, 2021, 1–12.
<https://doi.org/10.1155/2021/8325302>

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>