COLLABORATIVE FILTERING MODEL FOR CHINESE MUSIC GENRE RECOMMENDATION SYSTEM

Pusat Sumber FTSM

LUO WENHAO

UNIVERSITI KEBANGSAAN MALAYSIA

COLLABORATIVE FILTERING MODEL FOR CHINESE MUSIC GENRE RECOMMENDATION SYSTEM

LUO WENHAO

Pusat Sumber FTSM

PROJECT SUBMITTED IN PARTIAL FULFILMENT FOR THE DEGREE OF MASTER OF DATA SCIENCE

FACULTY OF INFORMATION SCIENCE AND TECHNOLOGY
UNIVERSITI KEBANGSAAN MALAYSIA
BANGI

2024

MODEL PENAPIS KOLABORATIF UNTUK SISTEM CADANGAN GENRE MUZIK CHINESE

LUO WENHAO

Pusat Sumber FTSM

PROJEK YANG DIKEMUKAKAN UNTUK MEMENUHI SEBAHAGIAN DARIPADA SYARAT MEMPEROLEH IJAZAH SARJANA SAINS DATA

FAKULTI TEKNOLOGI DAN SAINS MAKLUMAT
UNIVERSITI KEBANGSAAN MALAYSIA
BANGI
2024

DECLARATION

I hereby declare that the work in this thesis is my own except for quotations and summaries which have been duly acknowledged.

09 July 2024 LUO WEN HAO
P126919

Pusat Sumber FTSM

ACKNOWLEDGEMENT

This dissertation, in addition to the results of my intensive study about data science in my master's degree, has been spiced up with the help and support from a group of very friendly people. Therefore, I write this paragraph to express my sincere gratitude and admiration to all of you!

First and foremost, I would like to express my sincere gratitude to my supervisor Prof. Dr. Azuraliza Abu Bakar, who has always listened to my ideas and given me direction on how to successfully implement my project. She took the time to meet with me even when she was busy and carefully suggested changes to each chapter. I am grateful for her openness to accept my ideas and give me the space to create my own and her generosity in providing guidance on feasible solutions when things get difficult. Moreover, I will thank all the professors who participated in the data science course. Their high-quality teaching has helped me master knowledge in the field of data science and has become the technical framework for building this project and providing theoretical support for feasibility assessment of project implementation.

Secondly, I would also like to thank Dr. Wandeep Kaur, the coordinator of the Master of Data Science research report, for clearly presenting the important information about the project including the process, writing requirements, timelines, etc. in every briefing session and urging us to submit the milestones on time. This became an important basis for me to develop an efficient and high-quality project proposal and timeline. Furthermore, she was in charge of creating the Turnitin for each student's report, as well as providing next steps once the results were in. Without their conscientious attitude, our project submission would not have gone as smoothly as it did, so I would like to thank her and the Qursiah team from the bottom of my heart.

Thirdly, I would like to thank the Faculty of Information Science and Technology of Universiti Kebangsaan Malaysia for its support and strong learning atmosphere for students to conduct academic research. The strong faculty team gives each student specialized training and has set up academic paper related courses to improve students' academic research ability.

Lastly, I would like to give a special thanks to my family for their constant support and love. You have been a strong motivation for me to move forward, providing unwavering encouragement and love when I was lost. It is because of your support that I have been able to study abroad and pursue my education without fear until now.

Pusat Sumber FTSM

ABSTRAK

Jumlah maklumat yang meningkat dengan pantas boleh memenuhi keperluan pengguna yang pelbagai, tetapi ia juga menghadapi cabaran penyaringan data. Dalam menghadapi sumber muzik yang besar, pengguna selalunya tidak dapat membuat pilihan yang pantas dan sesuai. Oleh itu, sistem pengesyoran muzik telah menjadi alat penyelesaian yang berkesan dalam konteks ini dan telah digunakan oleh banyak platform media penstriman yang besar. Walau bagaimanapun, sistem pengesyoran muzik matang semasa masih menghadapi cabaran termasuk kekurangan pengesyoran yang diperibadikan, permulaan sejuk, kesederhanaan data dan masalah ekor panjang. Saya memilih kekurangan pemperibadian, kesederhanaan data dan permulaan sejuk sebagai penyataan masalah untuk projek ini dan menyelesaikannya. Di samping itu, terdapat lebih sedikit hasil penyelidikan dalam bidang algoritma pengesyoran muzik Cina setakat ini. Oleh itu, matlamat kertas ini adalah untuk menggunakan algoritma pembelajaran mesin untuk mencapai sistem pengesyoran muzik Cina yang diperibadikan dengan ketepatan yang tinggi. Objektif khusus saya adalah tiga lipatan; (i) untuk mencadangkan ciri dan skor tentang item muzik Cina untuk mencapai cadangan yang diperibadikan. (ii) untuk mencadangkan kaedah berasaskan kandungan untuk menyelesaikan masalah cold-start. (iii) untuk merumuskan pengiraan persamaan yang cekap untuk mengatasi masalah sparsity data. Empat algoritma pembelajaran mesin disiasat dalam kertas ini, termasuk "Algoritma jiran terdekat K" (KNN), "Penguraian Nilai Tunggal" (SVD), "Model Faktor Terpendam" (LFM) dan "Pemfaktoran matriks bukan negatif" (NMF). Keputusan eksperimen menunjukkan bahawa pengesyoran berasaskan kandungan menyelesaikan masalah permulaan sejuk, ketepatan yang tinggi menunjukkan bahawa pemperibadian telah diselesaikan, dan keterlanjuran data dipertingkatkan dengan ketara dengan penggunaan persamaan kosinus. Antaranya, algoritma "KNN" untuk item berasaskan sistem muzik Cina menunjukkan prestasi terbaik. Apabila bilangan pengesyoran (k) adalah besar, ketepatan (k=25) dan ingat kembali (k=20) dipilih sebagai metrik penilaian prestasi. Apabila bilangan pengesyoran (k) adalah kecil, ketepatan (k=10) dipilih sebagai indeks penilaian prestasi.

Pusat Sumber FTSM

ABSTRACT

The rapidly increasing amount of information can meet the diverse needs of users, but it also faces the challenge of data screening. In the face of massive music resources, users often cannot make quick and appropriate choices. Therefore, music recommendation system has become an effective solution tool in this context and has been applied by many large streaming media platforms. However, the current mature music recommendation system still has challenges including lack of personalized recommendation, cold start, data sparsity and long tail problems. I chose lack of personalization, data sparsity, and cold start as the problem statement for this project and solved it. In addition, there are fewer research results in the field of Chinese music recommendation algorithms so far. Thus, the aim of this project is to apply machine learning algorithms to achieve a personalised Chinese music recommendation system with high accuracy. Our specific objectives are three folds; (i) to propose features and score about Chinese music items to achieve personalized recommendation. (ii) to propose a content-based method to solve the cold-start problem. (iii) to formulate an efficient similarity computation to overcome the data sparsity problem. Four machine learning algorithms are investigated in this project, including "K-nearest neighbours algorithm" (KNN), "Singular Value Decomposition" (SVD), "Latent Factor Model" (LFM) and "Non-negative matrix factorisation" (NMF). The experimental results show that content-based recommendation solves the cold-start problem, high accuracy indicates that personalization has been solved, and data sparsity is significantly improved with the application of cosine similarity. Among them, the "KNN" algorithm for based-items Chinese music system performs the best. When the number of recommendations (k) is large, precision (k=25) and recall (k=20) are chosen as the performance evaluation metrics. When the number of recommendations (k) is small, accuracy (k=10) is chosen as the performance evaluation index.

Pusat Sumber FTSM

TABLE OF CONTENTS

		Page
DECLARATION		iii
ACKNOWLEDGEMENT		iv
ABSTRAK		iv
ABSTRACT		vi
TABLE OF CONTENTS		vii
LIST OF TABLES		x
LIST OF ILLUSTRATIONS		xi
LIST OF ABBREVIATIONS		xii
CHAPTER I	**INTRODUCTION**	
1.1	Introduction	1
1.2	Research Background	3
	1.2.1 Recommendation System	3
	1.2.2 Collaborative Recommender System	4
	1.2.3 Music Recommendation System	6
	1.2.4 Methods in Music Classification/ detection/ identification Models	8
	1.2.5 Proposed Solution	9
1.3	Problem Statement	10
1.4	Research Questions	10
1.5	Research Objectives	11
1.6	Project Scope	11
1.7	Significance of this Project	11
1.8	Summary	12
CHAPTER II	**LITERATURE REVIEW**	
2.1	Introduction	13
2.2	Recommendation System	13
2.3	Types of Recommender Systems Practice in Music	14
	2.3.1 Collaborative Music Recommender System	14
	2.3.2 Content-based Music Recommender System	15
	2.3.3 Hybrid Music Recommender System	16
	2.3.4 Chinese Music Recommender System	16

Pusat Sumber FTSM

2.4	Methods Support in Collaborative Music Recommender System	17
2.5	Machine Learning	19
2.6	Technical/Framework Support	20
2.7	Current challenges	21
	2.7.1 Cold start problem	21
	2.7.2 Lack of personalization	22
	2.7.3 Gray Sheep problem	22
	2.7.4 Lack of scalability	23
	2.7.5 Shilling Attacks problem	24
2.8	Summary	24
CHAPTER III	**METHODOLOGY**	
3.1	Introduction	25
3.2	Phase 1: Data collection and Development	26
	3.2.1 Data	26
	3.2.2 Data description	27
	3.2.3 Preprocessing Data	29
3.3	Phase 2: Chinese music feature extraction to achieve personalized recommendations	31
3.4	Phase 3: Design the content-based collaborative filtering algorithms to solve the cold-start problem	33
	3.4.1 Content-based filtering method	33
	3.4.2 Data processing framework	34
	3.4.3 Web framework	35
	3.4.4 Database	35
3.5	Phase 4: Developing a music recommendation method for Music item similarity calculation to overcome the data sparsity problem	36
	3.5.1 Music score	36
	3.5.2 Music recommendation algorithm	39
	3.5.3 Syntax code	42
	3.5.4 Music item similarity measurement	42
3.6	Phase 5: Method evaluation	46
	3.6.1 Precision	47
	3.6.2 Recall	47
	3.6.3 Accuracy	48
	3.6.4 Coverage	48
3.7	Summary	49

Pusat Sumber FTSM

CHAPTER IV	**RESULTS AND EVALUATION**	
4.1	Introduction	50
4.2	Comparing performance metrics of different similarity measures	50
4.3	Compare performance indicators for different recommended item quantities	52
4.4	Compare performance metrics of the one user under different interest tags	54
4.5	Compare the performance metrics of different recommendation algorithms	57
4.6	Summary	60
CHAPTER V	**CONCLUSION**	
5.1	Objective 1: Chinese music tags and scores for improved personalized recommendation	61
5.2	Objective 2: Content-based method to solve the cold-start problem	62
5.3	Objective 3: Similarity computation to overcome the data sparsity problem	63
5.4	Limitations	63
5.5	Future Work	64
REFERENCES		67
APPENDICES		
Appendix A	CODING	74
Appendix B	QUESTIONNAIRE	84

Pusat Sumber FTSM

LIST OF TABLES

Table No.		Page
Table 3.1	The songs sample size for different interest tags	29
Table 3.2	Music scoring rules	36
Table 3.3	Music score function	37
Table 3.4	Music score result	37
Table 4.1	Evaluation's results under different K-value	54
Table 4.2	Accuracy, recall, coverage result for 1 user	54
Table 4.3	Best performance tool for evaluating different tag	56
Table 4.4	Average accuracy results for 10 users	58
Table 4.5	Average recall result for 10 users	58
Table 4.6	Average coverage result for 10 users	59

Pusat Sumber FTSM

LIST OF ILLUSTRATIONS

Figure No.		Page
Figure 1.1	Global music streaming subscription market	2
Figure 1.2	Famous Recommendation System application	4
Figure 1.3	Netflix's use of collaborative filtering	5
Figure 1.4	Item-based collaborative filtering	6
Figure 1.5	NetEase CloudMusic webpage Screenshot	7
Figure 3.1	Process of Chinese music recommendation	26
Figure 3.2	Survey of 10 user	28
Figure 3.3	Chinese music feature extract process	32
Figure 3.4	Cosine similarity space representation of similar items 1 and 2	44
Figure 3.5	Confusion Metrics	46
Figure 4.1	Similarity result for Cosine and Pearson	52
Figure 4.2	Performance evaluate result in different k-items case	53

Pusat Sumber
FTSM

LIST OF ABBREVIATIONS

KNN	K-nearest neighbours algorithm
SVD	Singular Value Decomposition
LFM	Latent Factor Modeln
NMF	Non-negative matrix factorisation
CF	Collaborative Filtering
ML	Machine learning
CNN	Convolutional neural network
RNN	Recurrent neural network
SVM	Support vector machines
ALS	Alternating least squares
TP	Ture positive
FP	False positive
FN	False negative
TN	Ture negative
Et al.	and others
n.d.	no date

Pusat Sumber FTSM

CHAPTER I

INTRODUCTION

1.1 INTRODUCTION

The application of recommendation technology responds to streaming media's need for filtering overloaded data. Music is one of the popular fields of streaming media. Richter (2023) found that the global music industry has seen eight consecutive years of sustained growth after 20 years of decline, thanks to the growth of digital music and streaming services. digital music accounted for the largest share of global music revenues in 2021, and streaming services alone accounted for 67% of the industry's total revenues. In fact, digital media has become the mainstream channel for people to consume music. According to Global Streaming Statistics presented by Silber (2019), the number of global music streaming subscribers in 2018 was 180.3 million. As show in the Figure 1.1 by McCain (2023), ‘Spotify’ (Sweden) has the most subscribers in the streaming market in the world with a 31% share. ‘Apple Music’ (USA) came in second with 15 percent, while ‘Amazon Music’ (U.S.A) and ‘Tencent Music’ (China) each had 13 percent. Over time, the accumulation of old songs and new songs has built a huge number of music libraries. Moreover, in recent years, as ‘TikTok’, ‘YouTube’ and other platforms have promoted the new wave of music creation, more and more online singers have appeared, resulting in an unprecedented increase in the number of new songs released in the music library. This echoes the latest trend in the music industry where streaming reigns supreme and independent artists or self-published singers are becoming more and more popular.

Pusat Sumber FTSM

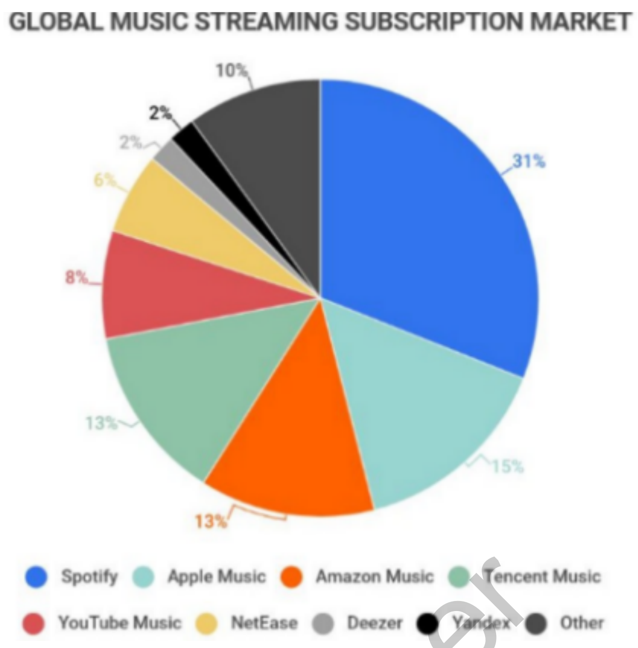


Figure 1.1 Global music streaming subscription market

Source: Zippia by McCain (2023)

Pusat Sumber FTSM

However, the larger the number of songs, the easier it is for users to fall into the confusion of choices. Specifically, due to the subjectivity of personal preferences and information overload, it is becoming more and more difficult for users to find their favorite music. In order to solve this problem, major streaming media platforms have launched music recommendation systems. In fact, the purpose of a music recommendation system is to provide users with convenience and a way to enjoy music. They are designed to ensure that the user's streaming experience feels enhanced. Specifically, music recommendation systems can predict the user's preference for specific elements and provide users with matching playlists after data filtering. This can greatly assist the user to find the desired song list and avoid wasting time caused by trial and error. In addition, the business benefits given by the recommendation system are also conducive to increasing the company's income, tapping potential consumers and stabilizing loyal customers. In previous researchers exploring the business value of recommendation systems, Panniello et al (2014) showed that recommendation systems increase the chances of converting viewers into consumers, while Jannach & Jugovac (2019) found that firms can lead to consumer demand and consumption upgrades by providing valuable information or item recommendations thereby.

Recommendation system is called machine learning application with modern significance. It uses a large number of data to build models to achieve prediction, reduce the search scope and recommendation function. With the mature application of recommendation technology, the existing recommendation systems are divided into Content-based recommender systems, collaborative recommender systems and hybrid recommender systems. In item-based recommendation, content is defined by items and the attributes or characteristics of items are used as the basis for recommendation. Collaborative recommendation can predict the interaction between users and projects in the future and take the similarity of user preference behavior as the basis of recommendation. Although recommendation systems have made great progress in the context of the rapid development of the Internet and mobile data technology, there are still flaws in terms of the current application results. In the survey by Kundu (2020), even though music recommendation systems are widely used in the commercial scope, there is no single perfect recommendation system. Therefore, how to improve the recommendation system to provide more extensive and effective recommendation requirements has become a topic of concern for developers. The aims to improve to the recommendation system are driven from both academic research and industrial application field this project, which is implement a content-based and collaborative filtering algorithm to solve some problems and limitations in the existing music recommendation system. Otherwise, provide more accurate music recommendation services, so that users can find more new music and enjoy personalized recommendation experience.

Pusat Sumber FTSM

1.2 RESEARCH BACKGROUND

1.2.1 Recommendation System

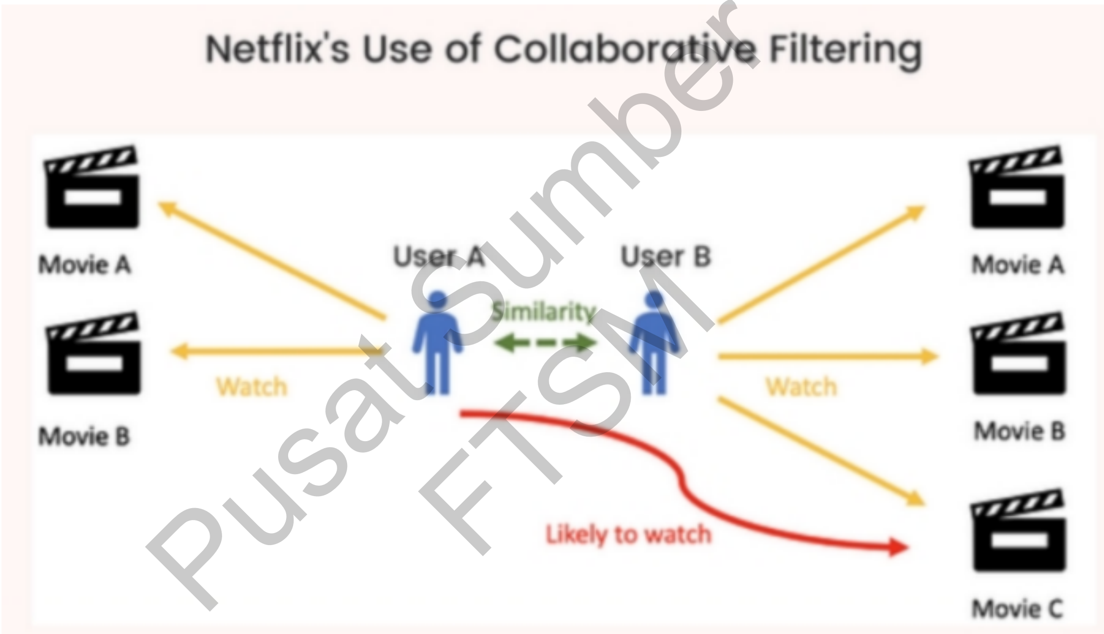
In the report by Qomariyah (2018), recommender systems are also known as information filtering systems, which aim to provide personalized suggestions and recommendations based on users' interests and preferences. In retrospect, Dong et al (2022) found a pioneering model for recommender system research named GroupLens by Prof. John Riedl, which works based on user-user collaborative filtering. As a matter of fact, the tremendous advancement in Internet technology and e-commerce has driven increased interest in the topic of Recommender system research. Moreover, with the commercial value of recommendation system being explored and valued, its application fields are not only in entertainment, medical treatment, education, e-commerce and other industries. Among them, the Amazon case has become the beginning of the prosperity of recommendation system industrial applications. Amazon which has a forward-looking vision, plays a leading role in the application of this field. Amazon introduced and patented 'Collaborative Filtering' (CF) in the late 1990s and realized an increase in sales. The successful Amazon case made recommender systems popular, and other online businesses began to implement RS on their websites. Currently, it has been widely used in various fields and successfully integrated into people's daily life. such as movie, shopping, book and music recommendation, etc. Moreover, recommendation system has been recognized by major online platforms including 'Amazon', 'Netflix', 'Google', 'TaoBao', 'TikTok', etc., as show in Figure 1.2. Regardless of the variability in the application domains of recommender systems, the goal is always to help users discover content or products that may be of interest to them and provide a better experience.

Pusat Sumber FTSM

Figure 1.2 Famous Recommendation System application

1.2.2 Collaborative Recommender System

Collaborative filtering system is a commonly used recommender system approach to find similarities by analyzing user behavior and other user's preferences and make recommendations based on these similarities. This approach is based on the concept of user behavior and group intelligence and can provide personalized recommendations. Collaborative filtering algorithms are usually categorized into two types including user-based collaborative filtering and item-based collaborative filtering. On the one hand, a user-based collaboration system finds similar user groups by comparing the behavior and preference matching between users, and then recommends items that a user has never touched or liked to the user. Specifically, if user A and user B have similar interests and purchase histories, the system can recommend items that user B likes to user A. As show the user based Collaborative Recommender System case from Netflix in Figure 1.3. Users A and users B have similar preferences from the viewing records of movie A and movie B. Therefore, when user a uses Netflix again, he will receive a push from movie C.



Pusat Sumber FTSM

Figure 1.3 Netflix's use of collaborative filtering
Source: Towards Data Science present by Skovhoj (2022)

On the other hand, the item-based collaborative filtering system compares the similarities between items and extracts the features of the users like items to recommend other items with similar features to the user. Specifically, if user A likes item X, the system can recommend items Y, Z, etc. like item X to user A. As show the items based Collaborative Recommender System case from Qutbuddi (2020) in Figure 1.4. User A has purchased Movie A, Movie B and Movie C. Movie D is recognised as matching User A's preference by the similarity between the items. Therefore, User A receives a recommendation message pushed by the platform to purchase Movie D.

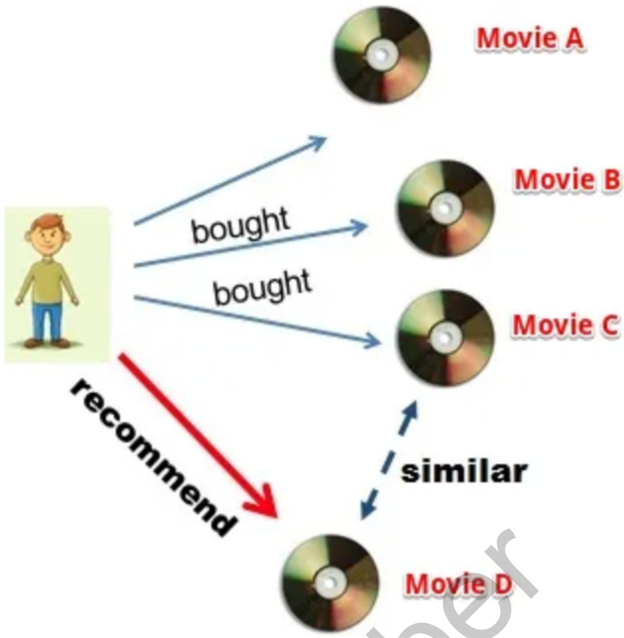


Figure 1.4 Item- based collaborative filtering
Source: Qutbuddin (2020)

Pusat Sumber FTSM

1.2.3 Music Recommendation System

Music recommendation system is a domain-specific recommendation system dedicated to providing users with personalized music recommendations. It needs to consider factors such as music characteristics, user preferences, habits, and social networks to provide accurate and diverse music recommendations. A music recommendation system is a recommendation system dedicated to recommending music to users. It provides users with personalized, accurate and diverse music recommendations by analyzing information such as users' music preferences, historical playback records, social network activities, and music characteristics. According to the report by Jannach et al (2018), as early as 1994, the Ringo system was created to recommend albums and artists to users. However, due to the undeveloped internet and information technology, the system's recommendations could only be exported via email. Moreover, the artist and album recommendation services of the three tasks supported by the system are retained in the music recommendation system developed later. Today's streaming-enabled web platforms and apps offer opportunities for the cross-phase development of music recommendation systems, which can provide users with direct recommendations and instant playback. User needs are expanding from basic preference recommendations to more accurate and personalised recommendations. Evidence from the recommendation system of “NetEase CloudMusic”, one of the major music streaming platforms in China, is shown in Figure 1.5. The "Song List Square" module on the webpage is used to provide users with a list of recommended songs and a one-click return to the parent webpage. The image shows an example of the platform's recommendation based on the user's time of day. The user uses the platform at 6pm, so the platform pushes songs for relaxing or driving during off-duty hours.

Figure 1.5 NetEase CloudMusic webpage Screenshot

Pusat Sumber FTSM

Typically, music recommendation system should fulfill the following 3 requirements:

1. User characteristic analysis

The system will analyze the user's personal information, music preferences, historical playback records, ratings and favorites, etc., to understand the user's music taste and preference. This can be achieved through techniques such as statistical methods, machine learning and collaborative filtering.

2. Music feature analysis

The system will extract and analyze music features, such as music genre, emotion, rhythm, style and singer. These features can be used to calculate the similarity between music and recommend related music.

3. Collaborative filtering recommendation

Music recommendation systems often use collaborative filtering algorithms to recommend music. It recommends music that one user likes to other users with similar interests based on similarities and behavioral patterns between users.

1.2.4 Methods in Music Classification/detection/identification Models

Music classification models play an important role in music recommendation systems. Today, common methods that have been practiced include content-based recommendation, collaborative filtering, deep learning, etc.

1. Feature Extraction

The first step in music classification is to extract features about the music. These features can include audio features (such as spectrum, rhythm, pitch, etc.), lyric features (such as word frequency, sentiment analysis, etc.) and metadata (such as artist, album, genre, etc.). The method of feature extraction can be based on traditional signal processing algorithms, or deep learning techniques such as 'convolutional neural network' (CNN) practice in study by Jha et al (2022) and 'recurrent neural network' (RNN). In study, Zhang (2022) practiced a music recommendation system based on 'Convolutional Neural Network'.

Pusat Sumber FTSM

2. Machine Learning Classification Algorithms

Once the features are extracted, various machine learning algorithms can be used for music classification, as the work by Sindhuja et al. (2021). Commonly used algorithms include ‘support vector machines’ (SVM), ‘decision trees’, ‘random forests’ and ‘naive Bayesian classifiers’, among others. These algorithms can learn the patterns and characteristics of music samples through the training set, and then classify new music samples into the corresponding categories.

3. Deep Learning Models

In recent years, deep learning has achieved remarkable results in music classification, refer survey work by Liu et al (2020). Deep learning models such as ‘convolutional neural network’ (CNN) and ‘recurrent neural network’ (RNN) are widely used in music classification tasks. These models can better capture the temporal and frequency domain structure of music, extract richer and higher-level musical features, thus improving classification performance.

Pusat Sumber FTSM

1.2.5 Proposed Solution

In order to improve the accuracy and personalization of the music recommendation system, this project proposes the use of a collaborative filtering algorithm. This algorithm will enable more accurate and personalized music recommendations based on analysis of similarities between items and other items in same user’s rating results, as well as the items scoring including user's preferences according tag selection, user behavior and popularity. In other words, the recommendation system realizes personalized recommendation by learning users' rating score for items, that is, the nearest neighbor model. In the method based on the nearest neighbor model, for a new item, the nearest neighbor method first finds the k items that are most similar to those judged by user u, and then judges the degree of their preference for the new item according to user u's preference for the k items. The key aspect in the application of this method is how to calculate the similarity between items through the attribute vector of items. If the vector space model is used to represent the similarity of projects, cosine similarity is selected as the best solution. In addition, python will be used as the programming language of the project and developed in combination with the mainstream web framework flask and data processing framework spark. As a supplement, database tools such as ‘MySQL’ will be used to store and manage data.

1.3 PROBLEM STATEMENT

1. Past studies have highlighted that one of the challenges faced by current music recommendation systems is the lack of personalization, present by Fayyaz et al (2020). Personalization emphasizes a unique recommendation list based on individual preferences, and there is inter-user variability. The system's lack of a mechanism to focus on the user's past preference or behavior may lead to a lack of accuracy in the recommendation results.

2. Previous studies have addressed that the music recommendation system is continuously accompanied by the cold start problem, from study by Dahdouh et al. (2019) and Fayyaz et al. (2020). Cold start is divided into two sub-problems containing product cold start and user cold start. Specifically, introducing new items with insufficient ratings or new users without a preference record may result in inaccurate or unresponsive recommendation forms. For new users, the recommendation list may be blank or inappropriate due to lack of historical data. Moreover, new projects will not initially be activated by the referral mechanism.

3. Kuar and Mohapatra (2021) argue that data sparsity is a common issue in collaborative recommendation systems, as recommendation mechanisms rely on absolute data item ratings or limited user response to ratings for all items. In other words, a certain type of project is overshadowed by popular and popular projects due to insufficient but high ratings, or people only rate projects related to past behavior. Ultimately, data sparsity will prevent users with niche preferences from finding the item through recommendation lists.

1.4 RESEARCH QUESTIONS

RQ1: How to identify and model to improved personalized recommendations?

RQ2: How to design and formulate a method that solves the cold start problem?

Pusat Sumber FTSM

RQ3: How to overcome the data sparsity problem by considering the similarity?

1.5 RESEARCH OBJECTIVES

The aim of this research is to applicate the collaborative filtering algorithm and method for calculating preference similarity in music recommendation systems to solve the problem about cold start, lack of personalization & diversification and data sparsity, to achieve more accurate, personalized and diverse music recommendations.

RO1: To propose Chinese music item tags and scores for improved personalized recommendation.

RO2: To propose a content-based method to solve the cold-start problem.

RO3: To formulate an efficient similarity computation to overcome the data sparsity problem.

Pusat Sumber FTSM

1.6 PROJECT SCOPE

The scope of this project development is limited to recommendation system for Chinese music media domain and collaborative filtering algorithm is used to implement its functionality. The technologies and frameworks used in this music recommendation system include python, Flask, database, Spark etc. Python is used for the construction of the website, Flask is the web framework used for it, Spark acts as a data processing role and the dataset is sourced from python crawler. This paper focuses on the experiments application of machine learning algorithms in recommendation performance test, including KNN, SVD, LFM, NMF.

1.7 SIGNIFICANCE OF THIS PROJECT

Throughout the existing music recommendation systems have been developed and matured, but there are still many challenges leading to a degraded user experience. This project aims to improve collaborative filtering algorithms in music recommendation systems to solve personalized and diverse recommendation, cold start and data sparsity problems. The system delivered in this research will have a positive impact on improving user experience and satisfaction with recommendation functions. Expected contributions include (i) providing accurate and personalized music recommendations. Furthermore, providing users with richer and more satisfying music choices. (ii) Increase the practice case of developing Chinese music recommendation systems, propose combination of popular algorithms and techniques, improve existing music recommendation methods. Moreover, achieve higher accuracy and effectiveness in personalized recommendations. (iii) Provide more intelligent, accurate and diverse music recommendation results through iterative design and evaluation optimization. (iv) to create an intuitive, convenient and customizable user interface that enhances user experience and satisfaction. By providing a highly interactive and customizable user interface, we expect to further promote user acceptance and usage of the music recommendation system

1.8 SUMMARY

This chapter presents an overview of recommender systems and their application in the music domain with its development background, as a way to express the relevance of the existence of this research. Moreover, summarizes the methods of recommendation system, such as feature extraction, scoring and machine learning algorithm for classification. Additionally, according to a review of the article to find out the challenges that are prevalent in today's recommender systems include lack of personalization, cold starts and data sparsity. It also briefly outlines the solution advocated in this project which is to implement a item-based music recommendation system using a nearest neighbor model that learns user preferences, with cosine similarity as a computational tool. The main objectives of this research are to propose a content-based and collaborative filtering Chinese music recommendation model to solve the problems of cold start, data sparsity and lack of personalization.

Pusat Sumber FTSM

CHAPTER II

LITERATURE REVIEW

2.1 INTRODUCTION

The literature review chapter mainly conducts a systematic review of previous literature related to the title. This chapter mainly focuses on the relevant literature on music recommendation systems, analyzing it from the aspects of historical development, introduction, theory, technical composition, and research results. This provides a research basis for developing an achievable theoretical and technical framework that matches this project. It also expanded the research on the development history and main technical categories of implementation in the field of Chinese music systems, laying the foundation for the future prospects of this branch of research. Additionally, explaining the limitations and critical analysis of existing journal disclosures helped this project identify research questions and objectives.

Pusat Sumber FTSM

2.2 RECOMMENDATION SYSTEM

The birth and development boom of recommender systems is closely related to the rapid development of Internet technology. The initial experimental stage of the recommender system can only be exported by e-mail. The discovery and attention of commercial added value make the recommendation system widely used in various industries. Nowadays, with the rapid development of website platforms and applications, instant and personalized recommendations are everywhere. Roy and Dutta (2022) argue that recommender systems can effectively filter online information. In order to meet the changing Internet access needs of contemporary computer users, including search habits, personality trends and trendy pursuits, etc. Specifically, the recommendation system guides users to find interesting information or realizes comfortable and personalized information push in a simple and fast way. Technological innovation and the prevalence of online service industries have promoted the wide application of recommender systems, for example, e-commerce, logistics, games, application stores, music, academic research resources, etc. Otherwise, the effectiveness of the system in solving the problem of information overload is recognized. Therefore, it is very important to build a high-quality and specific recommendation system to improve the quality of online information services. Moreover, recommender systems are classified into 3 types: item-based recommender systems, collaborative recommender systems, and hybrid recommender systems.

First of all, content-based recommendation systems are often used in books, movies and other fields. Usually, the collected description or feature data items will be put into the corresponding project configuration files, for example, singer, lyricist, release time can be defined as feature data of music. When it is recognized that the user holds positive comments on certain items, the user's personalized system customization can be realized by aggregating the data of these items in the profile.

Second, collaborative filtering-based recommendation systems are implemented by identifying similarity metrics between users. Specifically, a high degree of similarity can help the system to quickly identify areas of common interest of users and recommend them to other users with high similarity through information sharing.

Third, a hybrid recommendation system is a collection of two or more technologies. It is able to address the limitations of individual technical applications and effectively improve the performance and accuracy of recommendation applications.

2.3 TYPES OF RECOMMENDER SYSTEMS PRACTICE IN MUSIC

2.3.1 Collaborative Music Recommender System

In fact, the feasibility of music recommendation system based on collaborative filtering is demonstrated by Sun (2022). The author proposes a big data music personalized recommendation method based on big data analysis, which integrates user behavior, behavior context, user information, and music work information. Specifically,

Pusat Sumber FTSM

implementing proposals based on a series of user behavior records means that the model needs to learn the impact of different attributes on user interests. By comparing characteristic data such as singers, song styles, dates, and album names with data about users' online behaviors to determine their similarity, a set of project files with the highest similarity for this user group is finally exported as a material for music information sharing. Gil et al (2016) proposed a collaborative filtering music recommendation model depending on playing coefficients for artists and users in an earlier study. It is a model-based CF algorithm that uses user ratings to calculate item similarity instead of user similarity to deal with the scalability problem, consider that items are similar if they are liked/disliked by the same user. Therefore, assuming that users are expected to have similar preferences for similar items, recommendations can be provided to users. It turns out that the model outperforms other collaborative filtering methods, including those that exploit user attributes.

2.3.2 Content-based Music Recommender System

Looking back at previous music recommendation systems, it is not difficult to find that content-based music recommendation systems have also been demonstrated. Under the content-based music recommendation system edit by Niyazov et al (2021), the development characteristic is choosing content of musical compositions acoustic similarity as the recommendate basis. and built this model using two different techniques, on the one hand, using the common method of acoustic signature analysis. On the other hand, deep learning and computer vision methods are applied to optimize the results of the push system. Specifically, the method of forming a vector representation of a musical composition by extracting some acoustic features from an audio signal, such as music 'HZ ', score, rhythm, etc. The application of artificial neural networks improves the quality of recommendation results. Similarity of musical pieces is determined using a distance metric between vector representations of songs. The recommendation list consists of the 10 nearest neighbors of the test song selected in the training sample vector space. Moreover, Kaitila (2017) proposed a content-based music recommendation model that is relatively easy to implement, which is to build a preferred topic to automatically identify the basic topics that exist in the text source.

Pusat Sumber FTSM

Specifically, this model classifies data according to the content of music off-table features (i.e., lyrics and song annotations).

2.3.3 Hybrid Music Recommender System

Hybrid model in the application of music recommendation system Sun (2022) provides an empirical model, which implements the model based on content and collaborative filtering and various big data algorithms on the cloud computing platform Hadoop. Specifically, this model combines user behavior, behavior context, user information and music information, and a collaborative filtering proposal algorithm based on user behavior. Furthermore, the semantic similarity of lyrics and the co-occurrence similarity of songs are calculated based on the user's music download history. Therefore, both music similarity and tag similarity will be used as recommendation basis. The test results show the efficiency, scalability, stability and ability of this music recommendation system to meet users' personalized music needs. Zhang and Liu (2022) proposed a hybrid music recommendation model based on music gene and improved knowledge graph. The recommended references of this model include music genes, tags, knowledge graphs, geography, emotion and style preferences, etc. Specifically, by analyzing the massive user behavior records stored in music websites, the hybrid recommendation algorithm based on music genes and improved knowledge graphs can take into account the attributes of items by using the inherent semantic information of the items themselves and combining user tag information. At the same time, the algorithm utilizes two knowledge graphs to enhance the semantic information of item and user tags. Moreover, captures low-order and high-order features through a knowledge graph convolutional network. Finally, combined with the music gene recommendation model, personalized music recommendations are made for users.

2.3.4 Chinese Music Recommender System

Existing research on the topic of Chinese music recommendation systems is very limited. A survey of music personalisation recommendation systems in Xie and Ding (2018) suggests that many music recommendation systems have Cultural Metadata-based tagging, with the general idea being to use social tags provided by users to comprehensively assess the relevance of a singer or song. The tags reflect not only the

Pusat Sumber FTSM

results of categorising and characterising the content of the soundtrack ontology, but also the preference of using the soundtrack, which is highly flexible and open-ended. However, based on the differences in cultural backgrounds and ways of thinking between China and the West, music ontologies for Western music and listeners are not applicable to Chinese songs and Chinese-based music. Therefore, the study of constructing music ontologies based on Chinese music or applying user modelling with ontology technology has become popular in recent years. And in the limited literature found on Chinese music systems, authors are more inclined to propose a non-traditional solution. Fu et al. (2016) proposed a song recommendation algorithm based on sentiment analysis of Chinese vocabulary. By extracting the emotional information of the lyrics in the analysis to obtain the emotional theme, and thus the emotional theme of the song will be used as the sole criterion for recommendation in this paper. Combining the emotional gene sequences of lyrics to get various emotional information extraction coefficients and introduce a recommendation algorithm based on sentiment analysis. Similarly, Chen and Tang (2018) used the technical combination of content and sentiment analysis when building a Chinese song recommendation system, with lyrics as the data basis. Sun and Tang (2018) proposed the solution of Combining Valence-Based and Polarity-Based Sentiment Analysis on Lyrics, agreeing that lyrics are the data support for the project.

Pusat Sumber FTSM

2.4 METHODS SUPPORT IN COLLABORATIVE MUSIC RECOMMENDER SYSTEM

Anuprabha et al. (2020) considered 'collaborative filtering (CF)' and 'alternating least squares algorithm (ALS)' to be the most important techniques for building a music recommendation engine. On the one hand, 'collaborative filtering (CF)' is defined as a recommendation method based on recorded user behaviour or user ratings of items, presented by Joorabloo et al (2020). It can be seen that the recommendation system of collaborative filtering depends on either intuitive data (item ratings) or hidden data (user behavior). Building a scoring matrix before calculating similarity in collaborative filtering systems is crucial. Sun (2022) summarized that in addition to calculating the similarity based on the feature information of the songs, the core idea of recommendation also needs to refer to the listening time, click times, collections, downloads and other behaviors of users in the music library. Wu et al (2019) experimental results show that the recommendation algorithm based on user rating probability and item type can effectively improve accuracy and solve data sparsity. Specifically, the user's subjective behavior of item rating and selection can more directly capture the current degree of match with interests and obtain future rating changes to adjust the recommendation list.

And the basis of type discrimination is also based on users or items. The core step in user-based collaborative filtering is to compute the similarity of purchase decision behaviour between users. For example, most consumers who buy European, American and Chinese records tend to buy Chinese records. Thus, the recommendation system will recommend more Chinese paid records for this group of consumers, while reducing the relative recommendation rate of European and American records. The core step of item-based collaborative filtering is to calculate the similarity of scoring results between items. One the other hand, 'ALS' works by randomly filling the values of the user matrix before optimizing the music value, which can be implemented in spark. Spark appears frequently in the recommendation system presented by Hejing et al (2020) and Xie et al (2016). Spark is famous as a powerful big data processing framework. At its core, it serves as a distributed execution engine that supports a variety of workloads, including batch, streaming, and machine learning. Dahdouh et al. (2019) proposes that spark plays the role of data processing in the process of building the Recommender System. Moreover, its powerful functions and prominent advantages have become the reason why developers frequently choose it. Spark supports a variety of data management systems including 'MySQL', 'HDFS', 'HBase', 'Hive', 'Cassandra' and so on. The 'Spark API' provides a rich collection of libraries to support the development of cases of various levels of complexity. Finally, spark is programmer friendly because it supports popular programming languages including 'Scala', 'Java', 'Phyton', 'R'. In addition, experimental results built by the author and comparative studies with Weka and RStudio software show that in-memory computation with spark is the fastest and scalable solution. Tao et al. (2019) confirmed that the collaborative filtering recommendation algorithm based on Spark can complete iterative computing tasks excellently. Specifically, the calculation speed of the Spark distributed computing model is faster, and the calculated data is stored in the memory, which can prove that Spark is more suitable for large data system projects that need to run calculations

Pusat Sumber FTSM

repeatedly. In addition, spark introduced 'resilient distribution dataset (RDD)', which is a read-only and efficient fault-tolerant distributed shared memory model, which can realize efficient data sharing in the parallel computing stage.

2.5 MACHINE LEARNING

Machine learning has been described as an important computer science artefact in the era of the fourth industrial revolution and is a key component of the blueprint for the application and development of artificial intelligence. With its powerful ability to automatically learn and augment experience, machine learning has been fruitful in the field of data analysis and computation. Especially in applications that enhance system intelligence and functionality, machine learning is considered by some developers as the preferred solution. Nowadays, machine learning applications are no longer limited to cyber security, agriculture, e-commerce, smart homes, social media and more. Sarker (2021) states that data properties and characteristics and learning algorithms become critical in evaluating the effectiveness and efficiency of machine learning solutions. Among them, there are 4 major categories of data characteristics including structured data, unstructured data, semi-structured data and metadata. And there are 4 recognised types of machine learning algorithms: supervised learning, unsupervised learning, semi-supervised learning and reinforcement learning. In fact, algorithms with different principles can achieve different purposes. For example, supervised learning algorithm is suitable for classification and regression tasks, unsupervised learning is suitable for clustering and feature engineering tasks, semi supervised learning is suitable for machine translation and data annotation tasks and reinforcement learning is suitable for tasks with high complexity such as autonomous driving and manufacturing. Furthermore, the difference of data characteristics will directly lead to different results derived from similar algorithms. Therefore, it is a challenge to find machine learning algorithms that can match practical scenarios and all kinds of data.

Pusat Sumber FTSM

Algorithms are widely studied as common kernels for recommendation systems. Thomas and John (2020) state that 'k-nearest neighbor (KNN)' is currently the most common and easy to implement machine learning algorithm which focuses on user-to-user, item-to-item and user-to-item similarity measures. Thus, it can produce accurate results, but the lack of extensibility and sparsity becomes a limitation in the application of this algorithm. Guan (2024) summaries the traditional machine learning algorithms of user-item matching, clustering and association rules as solutions for recommendation systems. Firstly, the user-item matching approach is inspired by user requirements and item characteristics, also known as collaborative filtering algorithms. Specifically, after collecting the feature information, the matching degree is calculated and scored, and the scoring results become the basis for data filtering to export the recommendation list. Secondly, as a common unsupervised learning method, the principle of the recommendation mechanism implemented by clustering algorithm is to form clusters of similar data points, while the data points from different clusters are not similar. The well-known K-means algorithm and the hierarchical clustering algorithm are common examples of clustering algorithms. Thirdly, association rule algorithm is defined as a data mining method for finding relationships between itemsets from large scale datasets which contain multiple features and attributes. The core of its implementation for recommendation is filtering by setting a minimum threshold value. "Apriori algorithm", "FP-Growth algorithm" are common examples of clustering algorithms. In addition, Ding and Jiang (2021) summarised "Convolutional Neural Network (CNN)"-based recommendation algorithms applications regarding images, news, text and hybrids and suggested future optimisation recommendations. Such as heterogeneous data processing, privacy security and image feature extraction.

Pusat Sumber FTSM

2.6 TECHNICAL/FRAMEWORK SUPPORT

As a container for collecting data, the construction process of the website has received attention. The application of python in this field has achieved remarkable results. Nagpal and Gabrani (2019) summarize why Python is considered one of the fastest growing programming languages in their study and affirm its position at the forefront of data science applications, research and development. Furthermore, Idris et al. (2020) discuss Python in terms of history, strengths, and features, and expand on explanations about major python-based web frameworks - django and flask. In theory, python has won the favor of a large number of developers due to its low learning cost and strong readability. The current advantage of python lies in the library and community, which has benefited from the fact that python has been used in the programming industry for more than 20 years. In addition, scholars also expound on Django and Flask. Both provide support for storing rate-limited data in memory, in cache, or in the backend. However, Django provides a complete MVC framework that includes everything. Moreover, Flask is one of the micro-frameworks that follow the rules, but it also means that the design of Flask can be more flexible. Therefore, Flask will generally perform faster than Django. Even though Flask is based on the support of hundreds of queries per second, it will not slow down the operation speed. And Flask can freely integrate with NoSQL databases such as MongoDB and DynamoDB. Based on the strong scalability of Django, which can run a separate database server (MySQL). Both respond to the needs of big data projects. In addition, Python will also be used as a data acquisition tool. Hejing et al. (2020) verified the feasibility of using the Python-based scrapy framework to crawl website data. Moreover, it is emphasized that the various crawler frameworks and application libraries based on Python are very mature, so its reliability as a data acquisition tool is affirmed.

Pusat Sumber FTSM

2.7 CURRENT CHALLENGES

2.7.1 Cold start problem

In the study by Schedl et al (2018), the challenge from the Cold start problem that has been plaguing system developers stems from a lack of supportable data. Specifically, when new users register to the system or new projects are added to the catalog, the system does not have enough data associated with those projects/users. Ultimately, traditional recommender systems are not effective in recommending existing items for new users or new items for existing users.

However, the cold start problem raises a sub problem about data sparsity. In the study by Kuanr and Mohapatra (2021), Data sparsity is a common challenge in collaborative recommendation systems. If a data item has insufficient data but a very high rating, the data item may not be displayed in the recommendation list. This may cause users with relatively uncommon preferences or new users to be ignored. This reflects the limitation caused by the limited response of users to the rating or preference data for each music track, also known as the data sparsity problem. Obviously, the data sparsity in the process of calculating the similarity between users in the collaborative filtering algorithm creates troubles, which leads to the inability to accurately predict the user's preference. Moreover, Kuanr and Mohapatra (2021) suggest that solvable solutions include using algorithms to find pass-through connections between users through previous actions and feedback. Alternatively, the sparsity problem can be solved using preference relations rather than absolute ratings.

2.7.2 Lack of personalization

Kuanr and Mohapatra (2021) pointed out that creating a personalized website is the key to improving user interest, it should be an adaptive framework and try to provide each user with a unique instant personalized recommendation service. However, the lack of personalization is still evident as far as the system is concerned. Specifically, the recommendation mechanism cannot fully capture the personalized needs and preferences of each user. The song list exported by the recommendation system may deviate from the actual preferences of users, with low accuracy leads to lower user satisfaction with the recommendation results. Fayyaz et al (2020) point out that lack of personalization is equivalent to lack of diversity. Recommender systems can provide recommendations for similar items or more diverse items. In fact, the diversity of recommendations enables users to discover objects that are not easily found by themselves. Personalization is the main method to achieve diversified recommendation. Specifically, personalization refers to the uniqueness of recommendation lists across users, also known as inter-user diversity. Indeed, Milano et al. (2020) found that creating personalization will trigger ethical challenges. Given that most commercially known and successful recommender systems are based on hybrid or collaborative filtering techniques and build user models to generate personalized recommendations. Thus, user privacy is seen as an unavoidable challenge. In fact, the use of data within the scope of compliance is acquiesced by users, which is largely based on trust in the enterprise.

2.7.3 Gray Sheep problem

In the report from Mohamed et al (2019), "Gray sheep" derive from the behaviour of a subset of users with unusual preferences, i.e., they cannot be matched to neighbouring users with a high degree of similarity based on their feedback. It is their peculiar and

Pusat Sumber FTSM

unstable evaluations that cause the overall efficiency of collaborative filtering-based recommendation systems to decrease. This is due to the fact that traditional collaborative filtering recommendation systems are designed to cater for stable raters, and there is no way for systems to make accurate and effective decisions for people with elusive preferences. Therefore, it is widely accepted that the core of the solution to "gray sheep" is to adopt a new methodology to define the rating rules. Faiki et al (2019) proposed a new approach to find virtual neighbours for users with unique preferences to solve the "gray sheep" problem arising in collaborative filtering recommendation systems. Specifically, the method effectively increases the number of similar neighbours by considering users with different tastes from the target user as neighbours and performing similarity calculation. This is a reversal of the common method of calculating similarity, where similarity calculations are performed between users with the same taste. In fact, this method is successful in improving the prediction efficiency of recommendation system for users with "gray sheep" feature.

2.7.4 Lack of scalability

Present by Roy and Dutta (2022), the lack of scalability is common in recommendation systems that use collaborative filtering technology. The main reasons are the large demand for data from model training sets and the explosive growth of the number of items and users imported into the system. Whether it can effectively expand the processing capacity of the system under the pressure of explosive data loading to achieve stable performance has become a major challenge. It is well known that preferences are not constant and recommendation systems in various application domains need to update the recommendation results in real time to respond to the user's preference needs in the current state. Therefore, the scalability problem is also known as a product of the big data era. And the solution can start from the algorithm and system framework design. On the one hand, developers can use lower dimensional and clustering algorithms to reduce the query range from the entire database to a small cluster. On the other hand, the distributed storage and computing framework is used to support the storage and fast access of large-scale data. Moreover, this method can achieve high-speed and high-performance computing.

Pusat Sumber FTSM

2.7.5 Shilling Attacks problem

Many scholars have found the "shilling attacks problem" to be another important threat to recommendation performance, which comes from ratings given by dishonest users. In fact, the core of the operation of the recommendation mechanism is based on real ratings and the false information produced by pranksters, beneficiaries or competitors will undoubtedly disrupt the output of effective recommendation information. For example, some of the beneficiaries will specialize in making particularly high ratings for specific items, while competitors will make viciously low ratings for other competing items in the same class. Their goal is to influence the ranking of these items in the Recommended List and their chances of being recommended, i.e., the popularity of the items. "Shilling attacks" are a de facto fairness violation for the average user and greatly reduce the credibility of the recommendation system. This defeats the purpose of creating recommender systems to increase user engagement and retain a loyal customer base. Therefore, the most straightforward solution to the "shilling attacks problem" is to lock in these attackers and remove all associated false ratings and user information, presented by Roy and Dutta (2022).

Pusat Sumber FTSM

2.8 SUMMARY

This chapter focuses on a review of research on recommender systems and music recommender systems. In the module that introduces the types of recommender system practices in music, each of the four broad categories of systems and their main technical components are outlined includes: collaborative music recommender system, content-basis collaborative music recommender system, hybrid music recommender system and Chinese music recommender system. This section reviews the machine learning technology and the application of machine learning algorithm in recommendation system. Moreover, the technical support that has been proposed to implement the collaborative filtering system is summarized, including collaborative filtering (CF), alternating least squares algorithm (ALS), Spark, python, and Flask. In addition, the cold start problem, data sparsity problem, lack of personalization, gray sheep problem, lack of scalability and shilling attacks problem are raised in the current challenges section. The cold start problem, data sparsity problem and lack of personalization corresponds to the problem statement section.

CHAPTER III

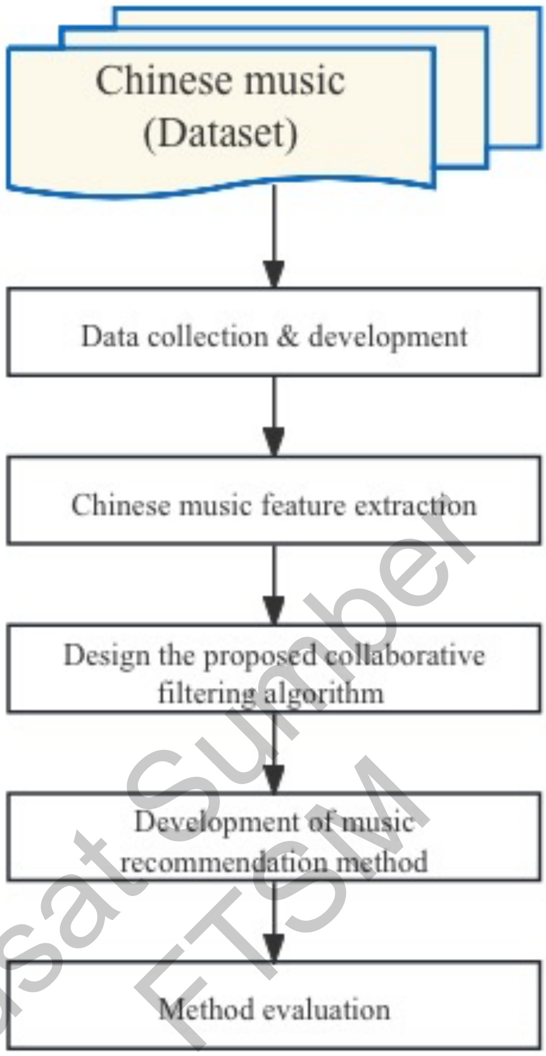
METHODOLOGY

3.1 INTRODUCTION

In this chapter, the author will explain the research methodology and design framework that will be used in this study. By definition, a methodology is a system of methods used in the study of concepts or theories in a particular field of study or activity by Rashin (2015). Therefore, the research methodology framework plays an important role in defining the objectives and selecting appropriate techniques to achieve them. The author applies the research path based on the methods described in this chapter. Furthermore, explained how the proposed application was designed, showed the path by which these researchers formulate questions and objectives and present results based on the data obtained during the research.

The methodology chapter mainly consists of 5 phases, including data collection and development, Chinese music feature extraction, design the proposed collaborative filtering algorithm, development of music recommendation method, method evaluation. As shown in Figure 3.1:

Pusat Sumber FTSM



Pusat Sumber FTSM

Figure 3.1 Process of Chinese music recommendation

3.2 PHASE 1: DATA COLLECTION AND DEVELOPMENT

3.2.1 Data

Gao (2021) argues that data is the foundation of machine learning applications and is the easiest first step to implement. The experimental dataset used in this project contains music item information, user information and user behavior records. The songs list data set is based on the existing 3773 playlists on NetEase Cloud Music Platform. The specific content is the relevant attributes and information of Chinese music, including music list name, music name, singer name, music style, music id. User information mainly consists of interest tags, which are collected from questionnaires of 10 users. User behavior records include all operations of these 10 users when using this music recommendation system. The use of those data set aims to help the collaborative algorithm better understand and identify users' music preferences based on items, thereby helping to improve the accuracy of music recommendations.

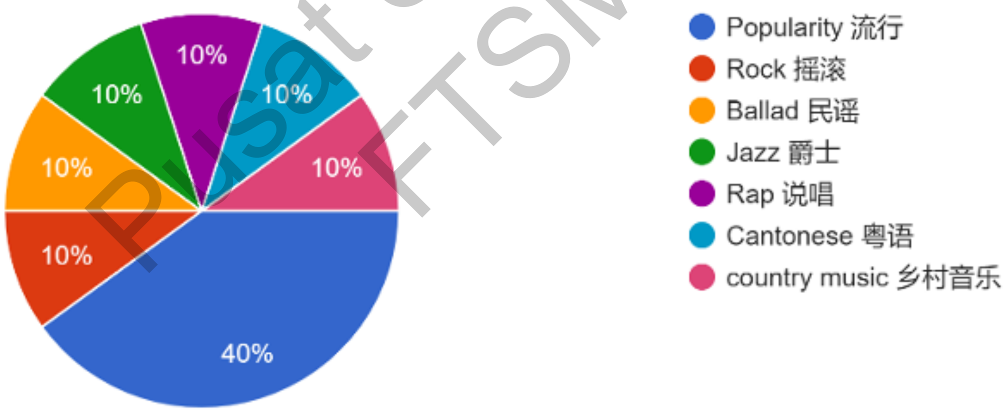
3.2.2 Data description

The music list name is created from a user-defined sharing post on the NetEase Cloud platform, and there are no definition rules for the name. However, the name of the appointment list can map the common characteristics of these songs to facilitate recommendation in the form of a playlist. For example, the music list name is defined by the theme of the singer's name, and the playlist name is defined by the application scenario such as running, learning etc. Parsing the 3773 song lists yields a total of 207,922 Chinese songs. In addition to the style definition consistent with the form, these independent song items are also attached with independent information that is only used to introduce specific songs. Music name is one of the most basic attributes of a song. It is also an important channel for users to directly query and obtain target music. The song names collected in this data set include WONIU, JUEJIANG, Little Love Song, Back View, etc. The Singer name marks the original singer or cover artist of each song, which is very important for identifying the user's singer preference. The singer names collected in this data set include well-known Chinese singers such as Jay Chou, Tian Fuzhen, and Li Ronghao. Music style is one of the indicators for effectively classifying music. It can be defined based on the speed of music rhythm, language, expressed emotions, application scenarios, etc. The music styles collected in this dataset include ancient style, pop, rock, ballad, jazz, rap, nostalgia, healing, sports, learning, Cantonese, countryside etc. Music ID is a numerical number that represents the unique identification of a song in the music library. Each song has its own unique song ID, which is very important for music platforms, players, etc. to accurately locate, play and manage songs.

In order to test the performance of the system, I collected 10 questionnaires from friends and classmates through Google forms, which were also regarded as the Chinese music preference information of 10 users. The survey subjects were all Chinese music

Pusat Sumber FTSM

listeners over 18 years old, with a male to female ratio of 7:3, and half of the users said that they had long used and benefited from the music recommendation system. As shown in Figure 3.2, based on the interest tag statistics of the 10 users' questionnaire, it is found that 4 users like pop-style Chinese music and the corresponding user numbers are user3144, user1277, user7500 and user5443. There are users who like rock, ballad, jazz, rap, Cantonese and countryside-style Chinese music correspond to one user, and the corresponding user numbers are user1282, user2697, user4751, user6915, user6679 and user5476. Therefore, we extracted Chinese music samples with these seven characteristics as research objects to complete the system test, including Pop, rock, ballad, jazz, rap, Cantonese and countryside. At the same time, I invited the above 10 users to personally experience the music system designed for this project and recorded their behaviors including liked, swipe across, listened and downloaded. It contributes greatly to the evaluation of user behavior scores, thus affecting the total music score and deriving differentiated recommendation lists.



Pusat Sumber FTSM

Figure 3.2 Survey of 10 users

I independently extracted and counted the Chinese music with these seven features from 207,922 songs. As shown in Table 3.1, the sample size of songs with popularity tag is 3759. The sample size of songs with rock tag is 3289. The sample size of songs with ballad tag is 751. The sample size of songs with ballad tag is 375. The sample size of songs with Cantonese tag is 550. The sample size of songs with rap tag is 712. The sample size of songs with Countryside tag is 631. According to the statistics, there are 10,067 Chinese music items with the above seven features after data preprocessing.

Table 3.1 The songs sample size for different interest tags

User ID	Interest tags	Sample size of songs
user3144, user1277, user7500, user5443	Popularity	3759
user1282	Rock	3289
user2697	Ballad	751
user4751	Jazz	375
user6679	Cantonese	550
user6915	Rap	712
user5476	Countryside	631
		10,067

Pusat Sumber FTSM

3.2.3 Preprocessing Data

Fan et al. (2021) In general, data preprocessing consists of five main tasks: data cleaning, reduction, scaling, transformation and partitioning. Data cleaning aims to improve data quality through missing value interpolation and outlier removal. In this project, data preprocessing mainly completes the following three tasks.

1. Processing of results returned by NetEase Cloud Music

   json.loads loads, parses out the playlist playlist_id, playlist name, tags, subscription count, song name songs, song id song_id, popularity in the json string, and eliminates playlists with subscriptions <100.

   After json.loads is loaded, parse out the playlist playlist_id, playlist name, tags, subscription count (subscribed_count), song name (songs), song id (song_id), and popularity in the json string. In order to make the dataset more consistent with popular preferences and to reduce memory space, a data reduction technique was adopted, which was achieved by eliminating song lists with <100 subscriptions.

2. Data cleansing

For most of the initial state datasets, the unstandardised nature of the data results in a large amount of data that is not of high utility value, and even some of the data may lead to errors in the operation of the system. Therefore, screening of useless information and comprehensive data cleansing becomes a necessary step for analysts in order to ensure the reliability and accuracy of the analysis results, presented by Gao (2021). The purpose of this step is to ensure that only the target items exist within the dataset: Chinese-language music and song lists. Specifically, use the regular rule '[a-zA-Z]' to exclude songs or song lists with English.

Moreover, in order to ensure the successful construction of feature engineering, it is necessary to delete data items without comment records. This is because items without comments mean that no one is interested in the song, which makes it meaningless to recommend it to the majority of users.

In addition, the conversion to unified units facilitates the reading of comment numbers. Specifically, the maximum unit of the number of comments in the count number is limited to *0,000. Even if the unit is *00,000, it needs to be converted to *0,000.

Finally, during the feature engineering construction process, the tag data of the playlist needs to be added to the attributes of the song. This is because the original data shows that the tags of the playlist contain keywords such as pop, nostalgia, and folk songs, but the tags of the songs themselves are missing. Therefore, associating playlist tags with songs is an important step in assigning song tags. When building the recommendation system, we calculate the tag score based on the user's interest tags and the song's tags.

3. Dataset Split

Pusat Sumber FTSM

Partitioning the dataset into training and testing sets is a necessary step in this project, which can help avoid overfitting, evaluate the performance of the model, and tune hyperparameters. I will set the ratio of training set to test set to 75:25.

Avoid overfitting: the training set contains actual data, while the test set contains unseen data. By comparing the prediction results of the training set and the test set, we can evaluate whether the model is overfitted, that is, whether the model performs well on the training set but performs poorly on unseen data. If overfitting is found, we can mitigate it by increasing the regularization term, reducing the number of features, or using a more complex model.

Evaluate model performance: By comparing the prediction results of the training set and the test set, we can evaluate the overall performance of the model. For example, we can calculate metrics such as precision, recall, F1 score, etc. to measure the model's performance in classification tasks.

Adjust hyperparameters: By comparing the prediction results of the training set and the test set, we can adjust the hyperparameters of the model to improve the performance of the model. For example, we can adjust hyperparameters such as learning rate, number of iterations, etc. to find the optimal model configuration.

Pusat Sumber FTSM

3.3 PHASE 2: CHINESE MUSIC FEATURE EXTRACTION TO ACHIEVE PERSONALIZED RECOMMENDATIONS

In fact, the most widely used music feature extraction methods are mainly divided into metadata and sentiment analysis of lyrics, usually using feature labels based on genre labels and audio samples, found by Basiński et al (2021). On the one hand, the established attributes of music can be conceptualized and measured to fit the user's music preferences, namely songs' own attributes. On the other hand, sentiment analysis is used to extract musical features in lyrics, which can capture cultural elements and style attributes, presented by Yılmaz and Scheffler (2023). The Chinese music features in this project are from songs' own attributes, displayed as list of music style in dataset. Therefore, the Chinese music feature extraction process is equivalent to the dataset extraction process, while extraction power is by data crawler.

As we all know, crawlers provide opportunities for automating large-scale data acquisition on the Internet. Claussen & Peukert (2019) provide guidelines for data crawling and state that Python is the most popular and widely used programming technical in the data crawling process. In this project, the process of crawling is summarized as boundaries for data crawling, identification of observations, obtaining the content. Since this process is often blocked, specific suggestions on how to circumvent Claussen & Peukert include changing the IP address, obtaining website offers by an application programming interface (API), obtaining and extracting readable directory files of multiple subpages, etc. In this project, Python is the programming language used in the data crawler. As shown in the figure 3.3, this process is divided into 3 steps. First, request the interface according to the parameters required by the interface. Secondly, use python to parse the JSON format to get the data content including playlist name, playlist type, playlist id, playlist introduction, list songs, song_name and song article. Finally, create a table according to the return fieldof json and insert the data into the table to complete the data entry.

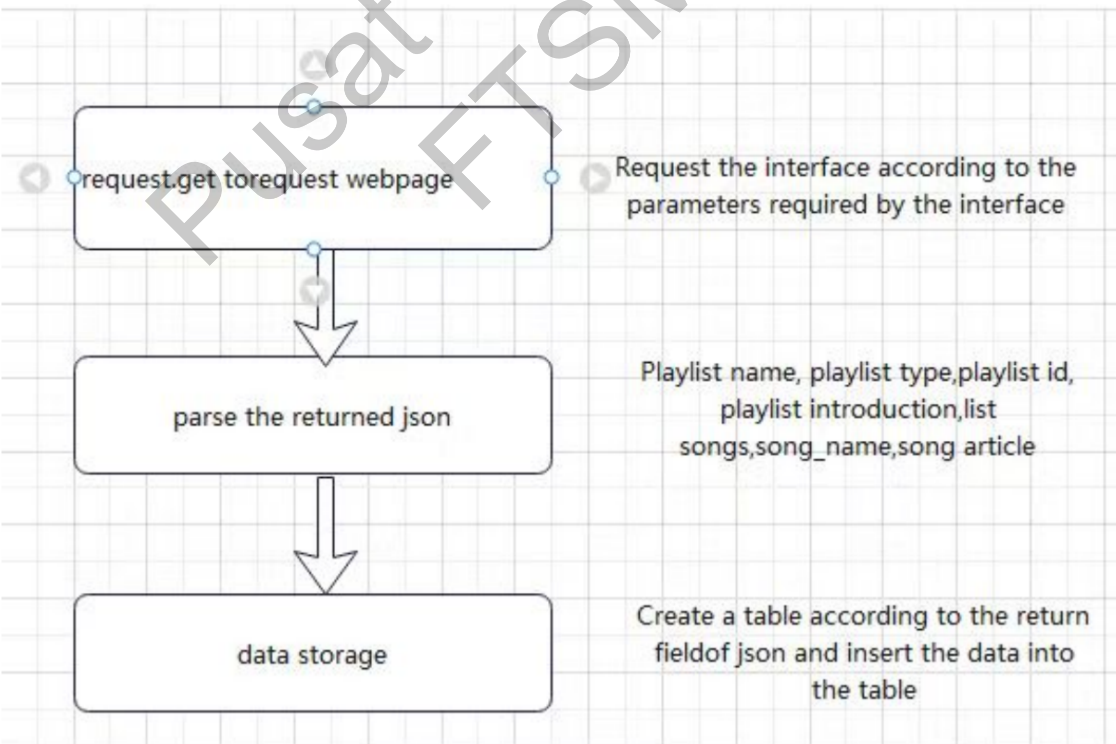


Figure 3.3 Chinese music feature extract process

Pusat Sumber FTSM

3.4 PHASE 3: DESIGN THE CONTENT-BASED FILTERING MODEL TO SOLVE THE COLD-START PROBLEM

3.4.1 Content-based filtering method

Nallamala et al (2020) showed that in content-based filtering, recommendations depend on the user's previous choices or desired elements. Thus, item descriptions and user-oriented profiles play an important role in implementing content-based filtering. Furthermore, the recommendation results don't rely on similarity calculation. Specifically, the relationship between the user preference and items' attributes is established in the user-item matrix. In fact, the low difficulty of implementing Content-based filtering is the most important reason for choosing it for this project. This is based on the fact that this model does not need to obtain any data about other users and can only focus on the introduction of music projects. This avoids the ethical issues caused by obtaining user private data. This corroborates the argument of Aziz & Fayyaz (2021): unlike user-based collaboration filtering approaches, content-based filtering approaches are agnostic to the user's profile or nothing to do with uer, as it does not affect its recommendation process in any way. Moreover, they pointed out that the advantages of the content filtering system also include that it can be used for recommendation immediately while ensuring the content availability of new items, which is beneficial to new users. Based on the huge amount of content, it is enough to recommend items that a single user has not found. Providing transparency to users, accuracy of recommendations does not require a large user base. The implementation of recommendation systems becomes easier, and the predictability is improved. In addition, the model can capture specific user interests based on recommendations tailored to specific users.

In this project, the content-based filtering mechanism focuses on checking the item genre (features) and user interest matching results. A positive result means that the English is consistent with the user's interests. On the contrary, a negative result means that the music is inconsistent with the user's interests. In theory, content-based filtering performs well in solving cold start problems and offline recommendations. On the one hand, the user based cold start problem has been solved by filling in at least one interest tag defined by the music genre at the time of registration, which is a common and

Pusat Sumber FTSM

realized way to actively collect user information. Thus, all the music that the user can match is found through filtering and the music of N music in this category is recommended under the random mechanism. On the other hand, the cold start problem based on the item is solved and the feature tags of the music itself help it find similar items in the database. The adaptation of the same type of music tags allows new items to be recommended immediately.

3.4.2 Data processing framework

Spark is defined as an improved distributed computing framework based on MapReduce, and its core concept is elastic distributed data set (RDD). In the study by Linzi et al (2023), Spark can implement task scheduling, invocation, operation and error recovery of applications, and provide APIs for upper-layer components. One of its features is that it is very fast in computation. Studies have shown that Spark is 100 times faster than MapReduce. And another feature of spark is locally based computation, which can effectively improve the efficiency of iterative computation in the case of sufficient memory space. Therefore, an important condition for running spark is to ensure sufficient memory space. In addition, when spark processes data, it can store intermediate processing result data in memory. Ahmed et al (2020) empirical analysis results on the performance of Apache Hadoop and Apache Spark show that Spark's performance is twice better than Hadoop under the WordCount workload and 14 times better under the Tera-Sort workload. Furthermore, throughput and speedup results show that Spark is more stable and faster than Hadoop because Spark data processing power is in memory instead of being stored on disk for map and reduce functions.

In this project, Spark is used for large-scale data processing and calculations. In addition to processing the user's music playback history data, the most critical thing is to calculate the similarity between music items. Specifically, I will be using Spark's machine learning library - MLlib, which is capable of building and training machine learning models. In this music recommendation system, the collaborative filtering algorithm in MLlib can implement the function of recommending similar music to users.

Pusat Sumber FTSM

Specifically, after completing the steps of data preparation including data cleaning, data transformation, etc., use Spark's read method to read the data, the file format supports text, CSV, JSON, etc. Then, use Spark's transform or map method to perform transformation operations on the read data, such as filtering, mapping, aggregation and so on. Finally, in order to optimise the recommendation results, the model parameters can be adjusted using grid search and the performance of the model can be evaluated by AUC score.

3.4.3 Web framework

Idris et al. (2020) introduced Flask as a framework for creating web applications in Python. Its most striking feature is its tiny size. In other words, Flask is a rule-following microframework that does one task at a time and does it correctly. Based on less design flask shows faster operational performance as compared to Django. It can provide support for hundreds of queries per second without slowing down operations. Additionally, Flask is able to freely integrate with NoSQL databases. Thakur and Jadon (2023) confirmed that another reason for the popularity of Django and Flask is that they can help reduce the duplication of tasks, thereby reducing the complexity of Web development code. Bonney et al. (2022) summarized the reasons why flask stands out in the python ecosystem as providing developers with creative flexibility, enabling the framework to target specific problems, accessibility options and its detailed documentation, and multi-method deployment.

Pusat Sumber FTSM

3.4.4 Database

Ohyver et al. (2019) explained in their study that MySQL is a relational database server that has become one of the most popular databases in the world due to its reliable open source database and strong compatibility. In addition, it also has the characteristics of economic efficiency and easy management. Rawat et al (2021) found that SQL has more outstanding reliability, speed, and ease-of-use features than other databases, making it the database of choice for many software and application developers on both online and desktop platforms. Moreover, MySQL has covered a wide range of customer groups, including individuals, small companies and industry giants. This project will use MySQL as data storage to store user information, song information, listening history and other data. According to the needs of the music recommendation system, an appropriate database structure will be designed. Including user table, song table, listening history table, etc. You can use ER diagram tools for database design.

3.5 PHASE 4: DEVELOPING A MUSIC RECOMMENDATION METHOD FOR MUSIC ITEM SIMILARITY CALCULATION TO OVERCOME THE DATA SPARSITY PROBLEM

3.5.1 Music score

Music score is an important decision-making basis for personalized recommendation, which comes from the different preference conformity and corresponding behavioral response of each user to different music items. Before we carry out the recommendation algorithm, we also need the support of users' music rating data. This project chooses to use the behavior of a certain music being liked, swipe across, listened to and downloaded by the current user to evaluate the user's liking for the music. The more times a certain piece of music is liked, listened and downloaded, the more the user likes the song. Moreover, the corresponding result is higher the score, the more times it is crossed, the more the user hates the song. Table 3.2 shows the music scoring rules and corresponding scoring values. Table 3.3 shows the equation for user behavior score, tag score, popularity score and total music score. It can be seen that the range of user scoring for individual music is [-10, 14].

Pusat Sumber FTSM

Table 3.2 Music scoring rules

Items	Score
Like once	+4
Swipe across once	-8
Same tags	+2
Tag exists but is not the same	-2
Score for a single review	+4

to be continued…

…continuation

Listened score	+1
Download score	+3

Table 3.3 Music score function

User behavior score	= Number of likes * Single score for likes + Number of swipes * Single score of swipes + Number of listens * Single score of listens + Number of downloads * Single score of downloads
Tag score	= When the song tag and user preference tag are the same, score 2 points or =If the song tag and the user preference tag are different,2points will be deducted.
Popularity score	= Number of comments on the song/Maximum number of comments on the song * Score for a single comment
Total music score	=User behavior score + Tag score + Popularity score

Pusat Sumber FTSM

The effect displays diagram of music score data is shown in table 3.4. The first column C1 is the user_id user code, the second column C2 is the item_id (music id), the third column C3 is the user_id's rating of item_id, and the fourth column C4 is the timestamp.

Table 3.4 Music score result

C1	C2	C3	C 4
2	59867	10	1556966041
2	62372	10	1556966041
2	63518	10	1556966041
2	63787	10	1556966041

to be continued…

…continuation

2	64269	10	1556966041
1	59867	9	1556966042
1	63518	11	1556966042
1	62372	-4.9784	1558152261.8956
1	62372	-4.9784	1558152544.7204
1	62372	-4.9784	1558152639.7094
1	62372	-8.9784	1558152768.1438
1	375180	2.0086	1558153742.7879
1	63787	2.1492	1558157591.3199
1	63787	-1.8508	1558157608.2450
1	178895	2.007	1558157614.1213
1	178895	3.007	1558157711.9966
1	64269	3.0117	1558383675.5071
1	64634	4.2654	1558418269.9617
1	64634	5.2654	1558418271.8304
1	115569	-0.8161	1558429752.6070
1	115569	2.1839	1558429772.8099
1	115569	2.1839	1558429793.5020
1	385973	1.0388	1558431283.5407
1	385973	1.0388	1558433988.9793
1	385973	1.0388	1558434210.5846
1	385973	1.0388	1558434233.7292
1	385973	1.0388	1558434356.2067
1	385973	1.0388	1558434667.6461
1	385973	1.0388	1558435598.9696

Pusat Sumber FTSM

3.5.2 Music recommendation algorithm

1. K-nearest neighbors algorithm (KNN)

K-nearest neighbors is a common mode-based method of content-based filtering (CBF). This is based on its ability to cope with the calculate challenges of project datasets with large-scale features and to be simple to implement and efficient. This method excels in solving problems of classification and regression and significantly improves their accuracy, present by Roy and Dutta (2022). The core recommendation algorithm of k-nearest neighbors algorithm (KNN) in this project, which uses user score data as a support for the derivation of predictive values, and similarity becomes the key. After the similarity calculation results are derived for sorting by taking the top K music items or users and using their similarity to the target object as weights, the scores are weighted and summed and finally the results are normalised by the sum of the similarity between these K music items or users and the target. As show in equation 3.1 according to the study by Schedl (2019), where r_{ui} is determined by the ratings of u for similar items, $N_u^k(i)$ Represents the set of the top k items most similar to the current item i in the item set evaluated by User u.

$$r_{ui} = \frac{\sum_{j \in N_u^k(i)} sim(i,j).r_{uj}}{\sum_{j \epsilon N_u^k(i)} sim(i,j)} \quad \cdots\cdots \; (3.1)$$

2. Singular Value Decomposition (SVD)

Presented by Jiang and Chaudhuri (2023) Singular Value Decomposition (SVD) is one of the most widely used linear algebra techniques in machine learning. In particular, its excellent performance in processing streaming data makes SVD a popular research topic in application fields such as IoT, prediction and recommendation systems. SVD becomes an essential part of Principal Component Analysis (PCA) by performing dimensionality reduction and extracting the most important information from the data. Specifically, SVD reduces the number of features of the dataset by reducing the spatial dimensions from N dimensions to K dimensions (where K < N). The SVD function is shown

Pusat Sumber FTSM

in equation 3.2, where U and V are orthogonal matrices and Σ is the pair A rectangular diagonal matrix with nonnegative values on the diagonals.

$$A = U\Sigma V^T \ldots\ldots (3.2)$$

A matrix structure is used in collaborative recommendation systems, where each row represents a user, and each column represents an item. The elements of this matrix are the user's ratings of the items. According to Equation 3.2, A is a utility matrix of m x n, and U is an orthogonal left singular matrix of m x r, representing the relationship between users and potential factors, Σ It is a diagonal matrix of r x r, describing the strength of each potential factor, and V is a diagonal right singular matrix of r x n, representing the similarity between the project and the potential factor. The underlying factors here are characteristics of the project, such as the genre of the music. SVD reduces the dimensionality of the utility matrix A by extracting its latent factors. It maps each user and each item to an r-dimensional latent space. This mapping helps to clearly represent the relationship between users and projects.

3. Latent Factor Model (LFM)

In study by Tegene et al (2023), LFM (Latent Factor Model) is an efficient feature mapping method that has become a popular research topic in the field of recommender systems. In fact, collaborative recommendation systems based on LFM have achieved great success in the field of personalized recommendation systems.

Yu (2020) explained that LFM can identify hidden topics or categories and establish relationships between features through hidden topics or categories. Specifically, interest preferences are obtained based on the user's current preference information and items corresponding to such interests are recommended to the current user. Therefore, the core idea of the LFM algorithm is to set the user-item rating matrix, solve two low-dimensional matrices, and multiply the two low-dimensional matrices to approximately represent the rating matrix. Also based on matrix factorization, LFM characterizes users and items

Pusat Sumber FTSM

by factor vectors inferred from user-item rating matrices and is used to find latent topics for these items. Therefore, the LFM algorithm formula is shown in Equation 3.3. The m×n rating matrix is R, the n×f user factor matrix P and the m×f item factor matrix Q. R [i][j] represents the rating of user u on item i, P [i][j] represents the user's level of interest in the item factor k, Q [i][k] represents the share of element k with item i, and T (Q) represents the transposition of matrix Q.

$$R_{UI} = P_U Q_I = \sum_{K=1}^{K} P_{U,K}\, Q_{K,I} \ldots\ldots (3.3)$$

4. Non-negative matrix factorization (NMF)

NMF (non-negative matrix factorization) aims to automatically extract hidden patterns from a series of high-dimensional vectors. Moreover, it has been successfully applied to dimensionality reduction, unsupervised learning and prediction. In Zhang (2012) it was concluded that NMF has good interpretability based on non-negative constraints, flexibility in the choice of objective function and solution algorithms, multi-channel usage and a solid theoretical foundation. In the field of mathematics, the expression of NMF is shown in Equation 3.4.

Pusat Sumber FTSM

Among them, for r-rank decomposition, W represents a matrix of size n × r and H represents a matrix of size r × m. included among this restriction is that all elements in W and H must be greater than 0. The rank matrix H contains r modes, which are the dominant spectral modes in the original data. In the weighting matrix W, each sample is assigned a weight corresponding to each spectral pattern, which represents the importance of the given pattern to the sample. In this project, A represents the matrix of user and music item ratings, W represents the matrix of user and implied feature preference values, and H represents the matrix of music item ratings and implied feature preference values. Where the implied feature preferences come from other labels for which interest labels have been defined. For example, the interest label used is pop, and the implicit features are rap and rock. the final matrix decomposition yields a prediction of the user's interest in unrated music. Chinese music items with more pronounced interest intensity are easily recommended to specific users.

$$A = WH \ldots\ldots (3.4)$$

3.5.3 Syntax code

For each unrated item ‘i’, traverse each item ‘j’ who has rated the current user (prepare to calculate the similarity) to Obtaining ratings for item ‘i’ and ‘j’ evaluated by current user and calculate the similarity s as two vectors and multiply the similarity s by a weight, that is, the rating of item ‘j’ on the current user for each unrated item ‘i’. After traversing all rated items, add the weighted similarities of each time and divide by the total similarity as the predicted value. After traversing all unrated items, recommend this item to these users based on the high of the predicted value. The following is a logical representation of the syntax code.

```
for Each unrated item i by the current user do
sim        = 0  ratSimTotal = 0
for Each rated item j by the current user do
rating = The current user's rating for item j
Get the ratings of users who rated both i and j, construct two column vectors
similarity = Similarity of two column vectors
ratSimTotal += similarity * rating
simTotal += similarity
end for
The rating prediction for unrated item i is ratSimTotal / simTotal
end for
Sort ratings and recommend to current user from high to low
```

3.5.4 Music item similarity measurement

The item-based collaborative filtering recommendation system is a commonly used recommendation algorithm, and its main parameters include similarity measure, similarity threshold and the number of recommended items.

Pusat Sumber FTSM

1. Similarity measure

Similarity measure is a measure used to calculate the similarity between items. Common ones include Pearson correlation coefficient and cosine similarity. The Pearson correlation coefficient is a number between -1 and 1, which measures the degree of linear correlation between two one-to-one corresponding series. That is, it represents the probability that corresponding numbers in two series will increase together or decrease together. It measures the tendency of numbers to change proportionally together, that is, there is a roughly linear relationship between the numbers in two series, presented by Liu and Zhang (2023). When the tendency is strong, the correlation value tends to 1. When the correlation is weak, the correlation value tends to 0. In the case of negative correlation, one series has high values and the other has low values -- the correlation tends to -1. The formal calculation of Pearson Correlation is shown in equation 3.5, present by Mana and Sasipraba (2020). In this project, the result of Pearson correlation can directly express the similarity between two music items 'a' and 'b'. When the result is close to 1, the two music items a and b are very similar. Thus, as a favourite of 'a music' will be pushed by 'b music'. On the contrary, when the result is equal to 0 or negative, it means that the two music items 'a', 'b' are completely unrelated or the listener of 'a' does not have the same preference as the listener of music 'b' at all. Therefore, the system will not push music item b to listeners of music a.

$$Pearson\ Correlation = \frac{n(\sum ab)-(\sum a)(\sum b)}{\sqrt{[n\sum a^2-(\sum a)^2][n\sum b^2-(\sum b)^2]}} \ldots\ldots (3.5)$$

Where, "a" and "b" are 2 different items.

While cosine similarity measures the cosine of the angle between the two vectors. That is, by measuring the cosine of the angle between the music items represented by vectors. The angle determines whether the direction of the vector is consistent or inconsistent in multidimensional space, present by Januzaj and Luma (2022). As shown in the figure 3.4, when the vectors point in the same direction, it means that the observation objects are similar and as the vectors between items get closer and closer, the similarity will be higher and

Pusat Sumber FTSM

higher. On the contrary, when the vector points in the opposite direction, it means that its similarity is low and the greater the distance between items, the smaller the similarity. When the value is used as the judgment index, the negative cosine similarity value indicates the dissimilarity between items. The cosine numerical results with similarity always fall in the range of 0 to 1, and the closer to 1, the greater the similarity between items. In addition, the measurement method to choose depends on the specific application scenario and data characteristics. In the survey of Januzaj and Luma (2022), cosine similarity is widely used and plays a very important role in analyzing and comparing the cases of document similarity comparison of text content. The formal calculation of cosine similarity is shown in equation 3.6. present by Mana and Sasipraba (2020). In this project, a cosine result of 1 means that vectors point in the same direction, music a and music b are completely similar. Therefore, the system will recommend music b to the fans of music a. On the contrary, a result closer to 0 means that vectors point in the indirection, this music item will be less recommended.

$$Cosine\ similarity\ (a,b) = \frac{(a,b)}{\|a\|.\|b\|} \ldots\ldots (3.6)$$

Where, "a" and "b" are 2 different items.

Pusat Sumber FTSM

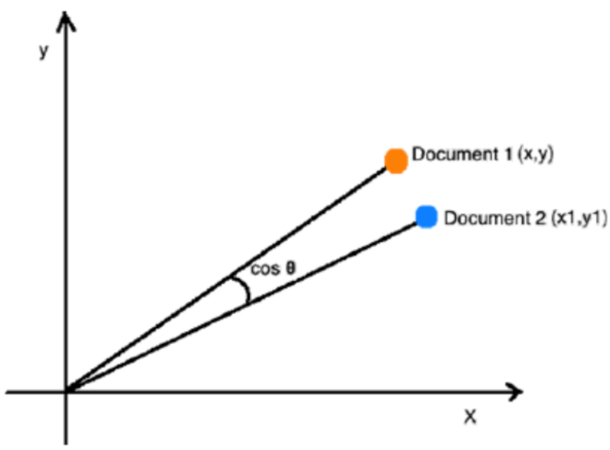


Figure 3.4 Cosine similarity space representation of similar items 1 and 2, example of document

Source: Januzaj and Luma (2022)

2. Similarity threshold

The similarity threshold is a threshold. When the similarity between two items exceeds this threshold, we consider them to be similar items. The choice of similarity threshold has a great impact on the accuracy and coverage of recommended results. If the threshold is set too low, the recommendation results may be too simple; if the threshold is set too high, the recommendation results may be too complex. Therefore, adjustments need to be made based on actual needs and data conditions.

3. Number of recommended items

The number of recommended items is the number of items recommended for each user based on the ratings of similar users. The number of recommended items can be adjusted according to actual conditions to balance the accuracy and coverage of recommendations. Generally speaking, the greater the number of recommended items, the higher the diversity of recommendations, but it will also increase the computational complexity and recommendation time.

Pusat Sumber FTSM

4. item-item based collaborative filtering mechanism

The focus of the collaborative filtering mechanism in this project is to sort the prediction scoring results from high to low, focusing only on the recommendation of items with Predict high-scoring and filtering out irrelevant items with lower scoring results. Specifically, a specific music item-based similarity matrix is created for each user to describe the similarity between two items, the similarity measure is performed using Pearson and cosine to find the k nearest neighbours that are similar to the target music item that has been selected as being of interest, and then the top N items are recommended to the current user based on the ratings sorted from the largest to the smallest results. In general, the steps of item-based collaborative filtering include generating an item-item score matrix, calculating item similarity, generating nearest neighbor items, predicting scores and sorting them from high to low, and deriving a recommendation list constructed from the Top N music items. Idea from He (2021).

3.6 PHASE 5: METHOD EVALUATION

In fact, different findings in response to different assessment classification methods will lead to different interpretation effects of some indicators. Therefore, it is a common work step to use a variety of evaluation methods and compare them in this paper. The purpose is to find the most interpretable evaluation method for the effectiveness of the algorithm or system. In this paper, performance evaluation has been measured by precision, recall and accuracy. Usually, the process of determining the results of this evaluation method may involve discussions about "TP", "FP", "FN" and "TN", as shown in the confusion matrix in figure 3.5. Therefore, the following explanation will help understand the calculation channels of different evaluation methods, present by Hicks et al (2022).

$$\mathbf{M} = \begin{pmatrix} \mathrm{TP} & \mathrm{FN} \\ \mathrm{FP} & \mathrm{TN} \end{pmatrix}$$

Figure 3.5 Confusion Metrics
Source: Hicks et al (2020)

Pusat Sumber FTSM

TP full name of true positive, indicates the number of correctly classified positive samples. For example, music samples containing rock features are correctly recommended to users with rock music preferences.

FP full name of false positive, indicates the number of samples that were incorrectly categorised as positive. For example, music samples that do not contain rock features are incorrectly recommended to users with rock music preferences.

FN full name of false negative, indicates the number of samples that were incorrectly categorised as negative. For example, music samples containing rock features were incorrectly recommended to users with no rock music preference.

TN full name of true negative, indicates the number of negative samples that were correctly categorised. For example, music samples that do not contain rock features are correctly pushed to users with no rock preference.

3.6.1 Precision

Gunawardana and Shani (2009), Visa and Salembier (2014) Precision measures the rate of true positives among all detections, as shown in equation 3.7. By other words, the proportion of positive samples that are predicted correctly among all the results that are predicted to be positive. Focus on the items to be recommended. Therefore, TP + FN is defined as the items to be recommended. The "precision" value is limited to the interval between 0 and 1, the closer to 1 the greater the proportion of correctly predicted samples in the category, the closer to 0 the greater the proportion of incorrectly predicted samples in the category.

$$Precision = \frac{TP}{TP + FP} \ldots\ldots (3.7)$$

Pusat Sumber FTSM

3.6.2 Recall

Visa and Salembier (2014) Recall measure the percentage of detected ground truth annotations, as show in equation 3.8. The proportion of correctly predicted positive samples among all positive samples represents the proportion of user-item interaction records included in the final prediction list. Focus on the items that the user is interested in. Therefore, TP + FN is defined as actually the items that the user is interested in. The numerical result of the recall always falls between 0 and 1, where the closer to 1 means that the result of recommending the positive category tends to be more or less perfect. On the contrary, a numerical result closer to 0 means that the error rate of the recommended positive category samples is higher. In this project, I used the recall_score methods in sklearn.metrics to calculate recall evaluation indicators.

$$Recall = \frac{TP}{TP + FN} \ldots\ldots (3.8)$$

3.6.3 Accuracy

Accuracy as shown in equation 3.9. The proportion of correctly predicted samples among all samples. It is widely used by researchers in various fields as one of the most common evaluation metrics in machine learning application research. In the case of unbalanced samples, it cannot be used as a good indicator to measure the results. The notable feature of "accuracy" is that its results may be misleading when faced with different category proportions. Therefore, the simplest way to improve its results is to adjust the sample allocation for each category. The accuracy value always falls within the range of 0-1. The closer it is to 1, the correct prediction of positive and negative samples, and the closer it is to 0, the incorrect prediction of positive and negative samples. In this project, I used the accuracy_score methods in sklearn.metrics to calculate accuracy evaluation indicators.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad \text{......(3.9)}$$

3.6.4 Coverage

Shinde & Potey (2016) state that coverage enables the assessment of the proportion of available information that can be predicted and is defined as two categories based on whether the recommended goal is an item or a user. On the one hand, catalogue coverage, also known as item-space coverage, is used to measure the performance of a system that uses items as recommendation material. The numerical result can indicate the proportion of available music items that the system recommends to users. On the other hand, prediction coverage, also known as user-space coverage, is used to measure the performance of a system that uses users as recommendation material. Its numerical results can indicate the proportion of users or user interactions for which the system is able to generate predictions. This project uses catalogue coverage as an assessment indicator, and the equation is shown in 3.10. Where S_r represents the set of items that are specifically recommended for that user, S_a represents the set of all available items. In layman's terms, the denominator represents the total number of all items in the database that can match the preference label and the numerator represents the number of items that can be recommended at one time.

Pusat Sumber FTSM

$$Catalogue\ coverage = \frac{S_r}{S_a} ......(3.10)$$

3.7 SUMMARY

The two most important parts of this chapter are data and algorithm. The sample data obtained through crawler technology contains music feature items of 3773 music lists. It explains that the steps of data preprocessing include data preprocessing, data conversion, data addition and deletion, data cleaning, and dividing the data into a training set and a prediction set. The ratio is 75:25. The algorithm development section lists the principles, technical support and reasons for selection of this recommendation system, including Content-based filtering, spark, flask, MySQL. In the process of building the algorithm, the rules of music score are defined, and the calculation results are obtained. Music item similarity measurement tools include cosine similarity and Pearson similarity. The k-nearest neighbors' algorithm (KNN), Singular Value Decomposition (SVD), Latent Factor Model (LFM)and non-negative matrix factorization (NMF) computes cosine similarity to ensure recommended implementation. Moreover, determine Precision, recall, accuracy and coverage as indicators of Performance Evaluation.

Pusat Sumber FTSM

CHAPTER IV

RESULTS AND EVALUATION

4.1 INTRODUCTION

The Results and Evaluation chapter mainly presents the experimental results and evaluates the effectiveness of the model or the proposed method. In this chapter, I will show the performance results of the Chinese music recommendation model by precision, recall and accuracy. Among them, comparative experiments are the method that can best highlight effectiveness and accuracy. Therefore, the comparative evaluation channels set up in this chapter include comparing performance metrics of different similarity measures of Cosine and Pearson, compare performance indicators for different recommended item quantities of k-value, compare performance metrics of the one user under different interest tags of music features, and compare the performance metrics of different recommendation algorithms of KNN, SVD, LFM and NMF.

Pusat Sumber FTSM

4.2 COMPARING PERFORMANCE METRICS OF DIFFERENT SIMILARITY MEASURES

Cosine similarity and Pearson correlation similarity are used respectively as similarity measures to calculate the similarity between items. As shown in Figure 4.1, the KNN performance is demonstrated based on Cosine similarity and Pearson correlation similarity in the initial model, that is, the K value is not defined. Under the Pearson correlation similarity. Precision is 0.69, which can be interpreted as 69% of the Chinese music recommended by the system are actually of interest to users. This shows that the Chinese music recommended by this system is highly reliable. The recall value of 0.65 can be interpreted as the system's recommendation results in 65% of all music items of interest to the user. This means that this system is able to effectively identify and recommend music in relevant categories based on the user's tastes. Furthermore, accuracy value is 0.76, which can be interpreted as 76% of all recommendation results provided by the system that can cover user tastes. Under the Cosine similarity, Precision is 0.71, which can be interpreted to mean that 71% of the Chinese music recommended by the system is of real interest to users. The recall value is 0.55, which can be interpreted as the recommendation result of the system accounting for 55% of all Chinese music items that users are interested in. Moreover, the accuracy value is 0.78, which can be interpreted as the percentage of all recommendations given by the system that can cover the user's taste is 78%.

When precision is selected as the evaluation index, the value of cosine similarity is higher than 70% and its performance is better than Pearson correlation similarity. When recall is selected as the evaluation index, the value of Pearson correlation similarity is higher than 60% and its performance is better than cosine similarity. When accuracy is selected as the evaluation index, the accuracy of cosine similarity is higher than 75% and its performance is better than Pearson correlation similarity. This means that this system has a fairly high accuracy in selecting and recommending Chinese music based on user tags. All in all, when our business goal is to improve the recall rate, we choose Pearson correlation similarity as the similarity measure. When our business goal is to improve the precision or accuracy rate, we choose Cosine similarity as the similarity measure.

Pusat Sumber FTSM

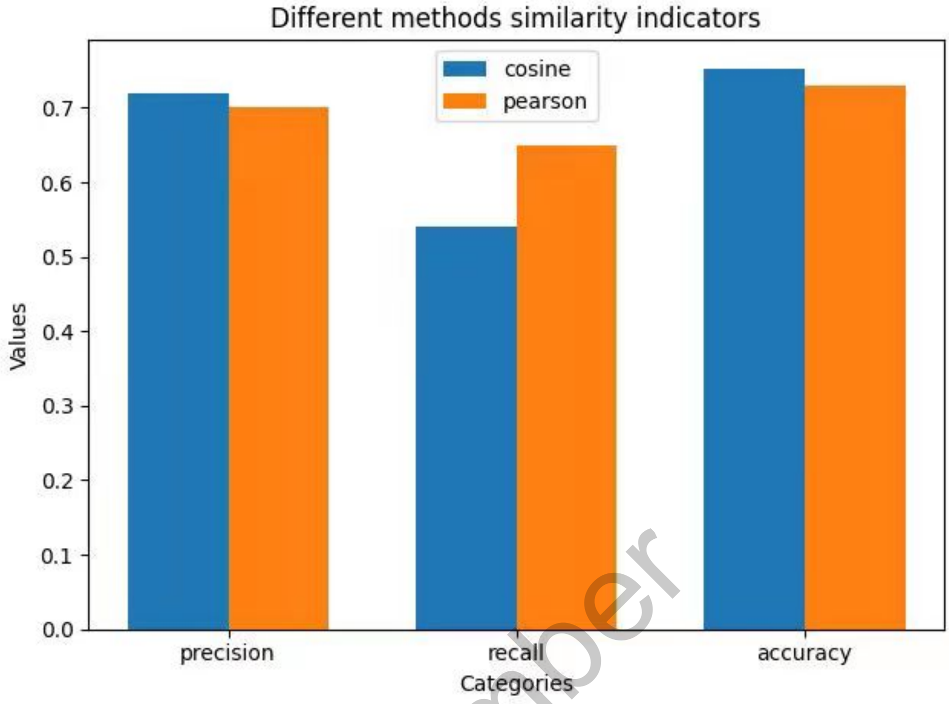
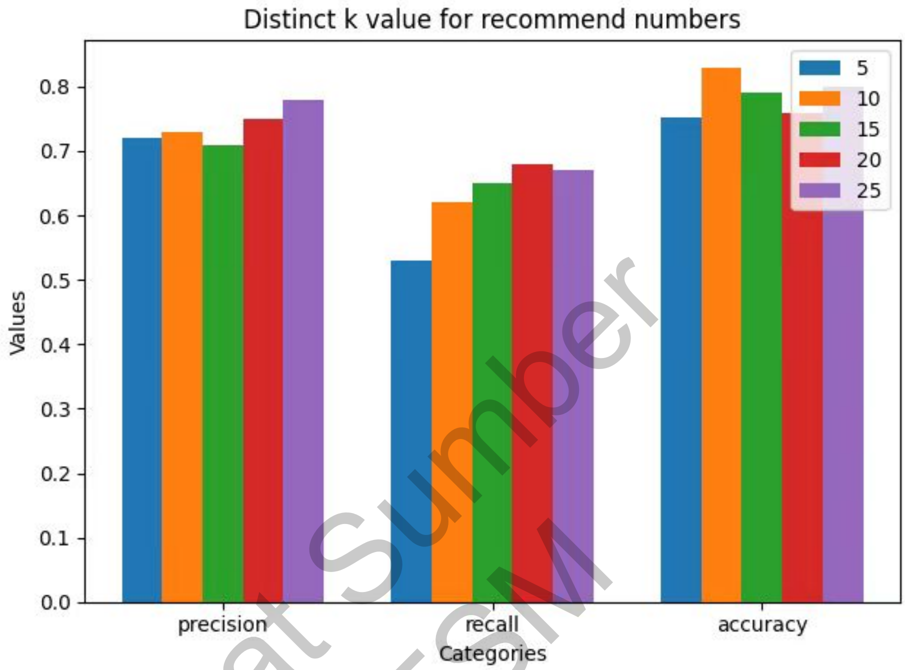


Figure 4.1 Similarity result for Cosine and Pearson

Pusat Sumber FTSM

4.3 COMPARE PERFORMANCE INDICATORS FOR DIFFERENT RECOMMENDED ITEM QUANTITIES

There are differences in the performance of evaluation results in different cases of recommended items, power by KNN. We can see that, the k value corresponding to the peak result of each evaluation indicator result is the optimal recommendation number. When precision is the evaluation indicator, the number of recommendations with the best effect is up to 25. When recall is used as an evaluation indicator, the best number of recommendations is 20. When accuracy is used as an evaluation indicator, the best recommended number is 10. Comparing the performance metrics for the number of recommended items at 5, 10, 15, 20, 25 respectively, when more recommended items need to be satisfied, the performance of precision and recall is better than accuracy. When less recommended items need to be satisfied, accuracy performs better than precision and recall. Therefore, the results of this experiment show that the number of recommended items can be increased (better to go up to k=20 in this case) appropriately when we have recall as a business goal. The number of recommended items can be increased (better to go up to k=25 in this case) appropriately when we have precision as a business goal. And the number of recommended items can be decreased (better to go up to k=10 in this case) appropriately when we have accuracy as a business goal. This is consistent with the experimental results of Ezeh (2023): smaller K may lead to higher accuracy, while larger K may lead to higher recall.



Pusat Sumber FTSM

Figure 4.2 Performance evaluate result in different k-items case

In fact, the algorithm's accuracy ranges from moderate to high, which means it provides users with accurate recommendations. The accuracy result is from 0.75 to 0.85 under the different k values. This shows that the KNN algorithm can relatively effectively predict Chinese music that suits the user's taste. The recall result is from 0.55 to 0.78 under the different k values. This shows that the KNN algorithm can relatively diversely and completely recommend results related to users. That is, the system can achieve diversified recommendations in different k value. The precision and credibility of recommended items related to user tastes can be identified through the Precision value, which is from 0.7 to 0.78 under different k value. Higher results indicate that the KNN algorithm can accurately provide users with taste-related Chinese music items.

Table 4.1 summarizes the optimal evaluation index results under the same K-value, and is represented by the symbol '✓'. We can see that, Accuracy is better than Precision, Recall under the k value of 5, 10, 15, 25. Moreover, Precision and accuracy perform similarly and are better than recall under k value of 20. Therefore, accuracy has the best overall performance, so that as an evaluation index, it can adapt and perform well in projects with different k values, and it shows that the KNN algorithm can effectively and significantly provide accurate and diverse recommendations to users.

Table 4.1 Evaluation's results under different K-value

	5	10	15	20	25
Precision				✓	
Recall					
Accuracy	✓	✓	✓	✓	✓

Pusat Sumber FTSM

4.4 COMPARE PERFORMANCE METRICS OF THE ONE USER UNDER DIFFERENT INTEREST TAGES

Select only one user, user3144, as the experimental object to evaluate the performance of different tags in a music recommendation system based on KNN algorithm, confirmed based on the Accuracy, recall and coverage. As shown in table 4, the model performs relatively well on the popularity tag in accuracy rate results and reaches 83%. The model has the best result for the popularity tag in the recall results and reaches 93.57%. The model performs relatively well on the popularity and Jazz tags with 0.67% in coverage rate results. Therefore, it is optimal to choose Recall as a measurement tool for the popularity tag.

Table 4.2 Accuracy, recall, coverage result for 1 user

User	Interest tags	Accuracy	Recall	Coverage
user3144	Popularity	0.83	0.9357	0.0067

to be continued…

…continuation

Rock	0.65	0.2866	0.01
Ballad	0.78	0.7131	0.0333
Jazz	0.69	0.2631	0.0667
Cantonese	0.74	0.2994	0.05
Rap	0.75	0.7623	0.04
Countryside	0.69	0.5749	0.04

Table 4.2 and table 4.3 summarize the optimal evaluation index results under different tags and is represented by the symbol "√". In the 'pop' tag, the performance index results are as follows: Accuracy of 0.83, recall of 0.9357, and coverage of 0.0067. Recall shows significant evaluation advantages, which can be interpreted as the system's recommendation results satisfy 93.57% of users with 'pop' preferences. In the 'rock' tag, the performance indicator results are as follows: Accuracy of 0.65, recall of 0.2866, and coverage of 0.01. Accuracy shows a significant evaluation advantage, which can be explained by the fact that 65% of all recommendation results given by the system can cover the user's 'rock' preferences. In the 'ballad' tag, the performance index results are as follows: Accuracy of 0.78, recall of 0.7131 and coverage of 0.0333. Accuracy shows a significant evaluation advantage, which can be explained by the fact that 78% of all recommended results given by the system can cover the user's 'ballad' preferences. In the 'Jazz' tag, the performance index results are as follows: Accuracy of 0.69, recall of 0.2631 and coverage of 0.0667. Accuracy shows a significant evaluation advantage, which can be explained by the fact that 69% of all recommendation results given by the system can cover the user's 'Jazz' preferences. In the 'Cantonese' tag, the performance index results are as follows: Accuracy of 0.74, recall of 0.2994, and coverage of 0.05. Accuracy shows significant evaluation advantages, which can be explained as all recommendation results given by the system can cover the user 'Cantonese' The preference ratio is 74%. In the 'rap' tag, the performance index results are as follows: Accuracy of 0.75, recall of 0.7623 and coverage of 0.04. Recall shows significant evaluation advantages, which can be explained as the system's

Pusat Sumber FTSM

recommendation results satisfy 76.23% of the users with 'rap' preference User. In the 'countryside' tag, the performance index results are as follows: Accuracy of 0.69, recall of 0.5749, and coverage of 0.04. Accuracy shows significant evaluation advantages, which can be explained as all the recommendation results given by the system can cover the user 'countryside' The preference ratio is 69%. All in all, Accuracy is the best tool for evaluating tag performance in KNN-based Chinese music recommendation system.

Indeed, the principle of variation of Accuracy, Recall and Coverage across interest tags can be hinted from equations on 6, 7. In addition, the number of classes predicted to be positive varies with the classification threshold. Firstly, recall will increase with increasing TP and decreasing FN. Secondly, Accuracy will increase with the increase of TP or TN. Thirdly, coverage will increase as the number of samples that meet user preferences increases. In the study of Foody (2023), it was pointed out that the reason for the large imbalance in coverage is that the greater the deviation of Interest tags from 0.5 in the total sample, the greater the degree of imbalance. Since the number of songs corresponding to each independent Interest tags accounts for a different and smaller proportion in the overall sample (total of 7 tags), and each coverage value only assesses 1 user preference tag, Coverage is presented in this project A huge imbalance.

In general, there is a trade-off between Accuracy, Recall and Coverage. Higher accuracy may mean lower recall. This is because the system will focus more on Chinese music items that closely match the user's preference and ignore the more marginal related items. Moreover, higher recall may mean lower coverage. This is because the system may focus more on Chinese music items that are relevant to the user's preferences to ensure coverage of the user's interest labels, while ignoring items that are not relevant to the interests in the whole sample. It follows that increasing coverage is unlikely to achieve a parallel increase in accuracy.

Table 4.3 Best performance tool for evaluating different tag

User	Interest tags	Accuracy	Recall	Coverage
user3144	Popularity		✓	

to be continued…

Pusat Sumber FTSM

…continuation

Rock	✓	
Ballad	✓	
Jazz	✓	
Cantonese	✓	
Rap		✓
Countryside	✓	

4.5 COMPARE THE PERFORMANCE METRICS OF DIFFERENT RECOMMENDATION ALGORITHMS

We invited a total of 10 classmates/friends to register 10 new users in the system, and filled in the music interest tags of each user. After users enter the system, according to their interest tags, 20 songs are recommended to them as initial recommendations. We use the four algorithms of KNN, SVD, LFM, and NMF to generate recommendation sequences and insert recommended records into the recommendation record table. After receiving feedback from 10 users on the recommended sequence, liking, listening, downloading and commenting on songs by users are considered as correct predictions, while swpie across and not liking songs are considered as prediction errors. Then, calculate their average recommendation accuracy.

As shown in Table 4.4, the average accuracy of KNN for recommending songs from different users is 0.753, SVD for recommending songs from different users is 0.742, LFM for recommending songs from different users is 0.587, and NMF for recommending songs from different users is 0.591. It can be seen that KNN becomes the optimal algorithm in this music recommender system, thus it is more adaptable to different users. Moreover, the four users with the same Interest tags- popularity perform differently in the four algorithmic models, which indicates that the individual differences among users affect the performance of the models.

Pusat Sumber FTSM

Table 4.4 Average accuracy results for 10 users

User	Interest tags	KNN	SVD	LFM	NMF
user3144	Popularity	0.83	0.68	0.66	0.53
user1282	Rock	0.81	0.67	0.62	0.54
user2697	Ballad	0.73	0.83	0.62	0.69
user4751	Jazz	0.68	0.76	0.53	0.54
user1277	Popularity	0.81	0.66	0.66	0.64
user6679	Cantonese	0.78	0.76	0.53	0.68
user6915	Rap	0.74	0.78	0.54	0.62
user5476	Countryside	0.73	0.8	0.54	0.51
user7500	Popularity	0.73	0.68	0.53	0.57
user5443	Popularity	0.69	0.8	0.64	0.59
Average accuracy		0.753	0.742	0.587	0.591

Pusat Sumber FTSM

As shown in Table 4.5, the average recall of KNN for different user song recommendations is 0.617, SVD for different user song recommendations is 0.559, LFM for different user song recommendations is 0.404 and NMF for different user song recommendations is 0.37. It can be seen that KNN becomes the optimal algorithm in this music recommendation system.

Table 4.5 Average recall results for 10 users

User	Interest tags	KNN	SVD	LFM	NMF
user3144	Popularity	0.9357	0.2669	0.6460	0.3660
user1282	Rock	0.6041	0.2196	0.4113	0.3017
user2697	Ballad	0.3319	0.8955	0.3104	0.2344

to be continued…

…continuation

user4751	Jazz	0.5150	0.6423	0.4595	0.4423
user1277	Popularity	0.6857	0.5967	0.5152	0.2103
user6679	Cantonese	0.5070	0.5678	0.3519	0.3307
user6915	Rap	0.8085	0.7232	0.4729	0.6132
user5476	Countryside	0.6746	0.2645	0.2128	0.4612
user7500	Popularity	0.5199	0.5831	0.3302	0.2142
user5443	Popularity	0.5858	0.8302	0.3329	0.5293
Average recall		0.6168	0.5590	0.4043	0.3703

As shown in table 4.6, the average coverage of KNN, SVD, LFM and NMF is 0.027 for different user music recommendations. In fact, the results of the same average coverage value for users who selected the same tag items are due to the influence of the scoring system based on tag selection, not due to differences in algorithms. Therefore, the coverage values of the four users who all selected popularity as the interest tag are the same based on different algorithms. Moreover, based on the difference in the core idea of matrix decomposition of each algorithm and the influence of the fixed proportion of different tags in the total music library (data set) in the data set, the coverage results of different tags are different. For example, for the popularity tag, the coverage is 0.0067. For Rock tag, the coverage is 0.01. For Cantonese tag, the coverage is 0.05. This is consistent with the results found by Longo (2018) that recommender systems using collaborative algorithms can only recommend a very small fraction of the training set items, which manifests itself as a much lower coverage of collaborative recommenders than random recommenders.

Table 4.6 Average coverage results for 10 users

User	Interest tags	KNN	SVD	LFM	NMF
user3144	Popularity	0.0067	0.0067	0.0067	0.0067

to be continued…

Pusat Sumber FTSM

…continuation

user1282	Rock	0.01	0.01	0.01	0.01
user2697	Ballad	0.0333	0.0333	0.0333	0.0333
user4751	Jazz	0.0667	0.0667	0.0667	0.0667
user1277	Popularity	0.0067	0.0067	0.0067	0.0067
user6679	Cantonese	0.05	0.05	0.05	0.05
user6915	Rap	0.04	0.04	0.04	0.04
user5476	Countryside	0.04	0.04	0.04	0.04
user7500	Popularity	0.0067	0.0067	0.0067	0.0067
user5443	Popularity	0.0067	0.0067	0.0067	0.0067
Average coverage		0.0267	0.0267	0.0267	0.0267

Pusat Sumber FTSM

4.6 SUMMARY

In this chapter, the evaluation results of the music recommendation system developed based on KNN are shown and compared with the recommendation systems based on SVD, LFM and NMF algorithms. First of all, when there are many recommended items, it is best to choose Recall as the evaluation tool. It shows that the system can provide users with diversified Chinese music recommendations by effectively identifying relevant items. When there are few recommended items, accuracy is chosen as the best evaluation tool. Secondly, Cosine similarity to achieve similarity measure, and it performs best when using precision and accuracy as indicators. Third, Accuracy is the best tool when measuring the recommendation performance of different tags. It shows that the system can provide users with highly accurate Chinese music recommendations. However, the best measurement tool for the popularity tag alone is to select recall. Finally, in the comparison based on different users and different algorithms, it was found that KNN is a relatively excellent algorithm.